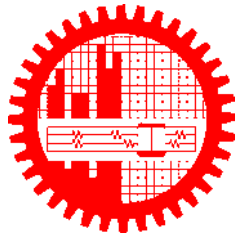


M.Sc. Engg. (CSE) Thesis

# **Risk Prediction of Loan Default Using Knowledge Graph**

Submitted by  
Md. Nurul Alam  
0417052082

Supervised by  
Dr. Muhammad Masroor Ali



Submitted to  
**Department of Computer Science and Engineering**  
**Bangladesh University of Engineering and Technology**  
Dhaka, Bangladesh

in partial fulfillment of the requirements for the degree of  
Master of Science in Computer Science and Engineering

March 2022

## Candidate's Declaration

I, do, hereby, certify that the work presented in this thesis, titled, "Risk Prediction of Loan Default Using Knowledge Graph", is the outcome of the investigation and research carried out by me under the supervision of Dr. Muhammad Masroor Ali, Professor, Department of CSE, BUET.

I also declare that neither this thesis nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

*Md. Nurul Alam*

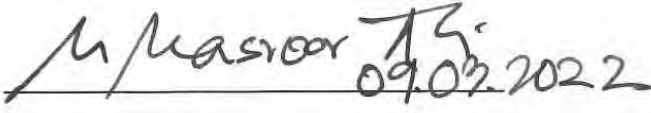
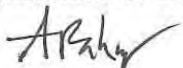
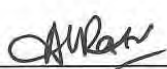
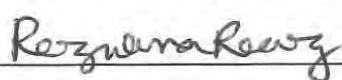

---

Md. Nurul Alam

0417052082

The thesis titled “**Risk Prediction of Loan Default Using Knowledge Graph**”, submitted by Md. Nurul Alam, Student ID 0417052082, Session April 2017, to the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, has been accepted as satisfactory in partial fulfilment of the requirements for the degree of Master of Science in Computer Science and Engineering and approved as to its style and contents on March 9, 2022.

### Board of Examiners

1.  09.03.2022  
Dr. Muhammad Masroor Ali  
Professor  
Department of CSE, BUET, Dhaka  
Chairman  
(Supervisor)
2.   
Dr. A.K.M. Ashikur Rahman  
Professor and Head  
Department of CSE, BUET, Dhaka  
Member  
(Ex-Officio)
3.   
Dr. A.B.M. Alim Al Islam  
Professor  
Department of CSE, BUET, Dhaka  
Member
4.   
Dr. Rezwana Reaz  
Assistant Professor  
Department of CSE, BUET, Dhaka  
Member
5.   
Dr. Md. Monzur Morshed  
Professor  
Department of Accounting & Information Systems  
University of Dhaka, Dhaka  
Member  
(External)

# Acknowledgement

*All praises due to Allah, the most benevolent and merciful.*

First and foremost, I would like to express my gratitude to Almighty Allah for the unwavering support with abundant blessings to help me complete my research work and submit the master's thesis on time and without any difficulties.

I would like to convey my heartfelt gratitude to my honorable supervisor, Professor Dr. Muhammad Masroor Ali, for believing in me and allowing me to work under his invaluable guidance. He introduced me to the interesting and exciting discipline of the semantic web and knowledge graphs. I have learned how to conduct successful research from him, as well as how to write an academic paper, speak, and present well. I am grateful for his patience in carefully listening to my ideas, critical analysis of my observations, reading through my manuscripts numerous times, detecting flaws (and amending thereby) in my thinking and writing, pointing me in the right direction, instilling hope and inspiration in me, and motivating me to continue my research. I want to offer my gratitude and deep respect to him once more for his constant guidance, recommendations, and unwavering supervision throughout this thesis work.

I want to convey my sincere gratitude to all of the members of my defense board, Dr. A.B.M. Alim Al Islam, Dr. Rezwana Reaz, Dr. Md. Monzur Morshed, and Dr. A.K.M. Ashikur Rahman, for their insightful remarks and constructive feedback for improving my thesis work further.

Finally, I sincerely thank my parents for their love, support, and prayer for me. I also thank my wife and children for their patience, sacrifices, and understanding throughout my M.Sc. program.

Dhaka  
March 9, 2022

Md. Nurul Alam  
0417052082

# Contents

<b>Candidate’s Declaration</b>	<b>i</b>
<b>Board of Examiners</b>	<b>ii</b>
<b>Acknowledgement</b>	<b>iii</b>
<b>List of Figures</b>	<b>vi</b>
<b>List of Tables</b>	<b>vii</b>
<b>List of Algorithms</b>	<b>viii</b>
<b>Abstract</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Objectives of the Thesis . . . . .	2
1.3 Outline of Methodology . . . . .	3
1.4 Research Contribution . . . . .	4
1.5 Organization of the Thesis . . . . .	4
<b>2 Preliminaries</b>	<b>5</b>
2.1 Financial Risk . . . . .	5
2.1.1 Market Risk . . . . .	5
2.1.2 Credit Risk . . . . .	6
2.1.3 Operational Risk . . . . .	7
2.2 Semantic Web . . . . .	8
2.2.1 Resource Description Framework . . . . .	9
2.2.2 Ontology . . . . .	10
2.2.3 Ontology Mapping . . . . .	11
2.3 Knowledge Graph . . . . .	11
2.3.1 How to Build Knowledge Graphs? . . . . .	14
2.4 Knowledge Graph Embedding . . . . .	15

2.4.1	Translational Models . . . . .	16
2.4.2	Semantic Matching Models . . . . .	18
2.5	Applications of Knowledge Graphs . . . . .	20
2.5.1	Organizational Data Governance and Integration Platform . . . . .	20
2.5.2	KGE to Advance Precision for Machine Learning . . . . .	22
2.6	Contextual AI . . . . .	23
<b>3</b>	<b>Related Work</b>	<b>24</b>
3.1	Statistical Methods in Consumer Credit Scoring . . . . .	24
3.2	Machine Learning Methods for Managing Credit Risk . . . . .	26
3.2.1	Supervised Learning Methods . . . . .	26
3.2.2	Unsupervised Learning Methods . . . . .	27
3.3	Deep Learning Techniques for Default Prediction . . . . .	28
3.4	Research Queries . . . . .	28
<b>4</b>	<b>Methodology</b>	<b>30</b>
4.1	Knowledge Extraction . . . . .	31
4.2	Semantic Data Modeling . . . . .	33
4.3	Knowledge Graph Construction . . . . .	34
4.4	Knowledge Graph Embedding Model Selection . . . . .	35
4.5	Loan Classification Using Knowledge Graph Embedding . . . . .	36
4.6	Explanations for Loan Default Prediction . . . . .	37
<b>5</b>	<b>Experimental Result and Analysis</b>	<b>39</b>
5.1	Experimental Data . . . . .	39
5.1.1	Exploratory Data Analysis . . . . .	41
5.1.2	Data Preprocessing . . . . .	44
5.2	Experimental Settings . . . . .	46
5.3	Training Knowledge Graph Embedding . . . . .	46
5.4	Evaluation Metrics . . . . .	48
5.5	Experimental Result . . . . .	51
5.6	Analytical Discussion . . . . .	53
<b>6</b>	<b>Conclusion and Future Work</b>	<b>54</b>
6.1	Summary and Conclusion . . . . .	54
6.2	Future Research Direction . . . . .	55
	<b>Publications</b>	<b>56</b>
	<b>References</b>	<b>57</b>

# List of Figures

2.1	Taxonomy of financial risks . . . . .	6
2.2	Semantic Web technology stack [1] . . . . .	8
2.3	Structure of a RDF statement . . . . .	9
2.4	Sample RDF triples in a graph . . . . .	10
2.5	An example of knowledge graph built from the knowledge base of Table 2.1 [2]	13
2.6	A top-down approach for the construction and implementation of knowledge graphs . . . . .	14
2.7	Simple illustrations of TransE, TransH, and TransR [3–5] . . . . .	17
2.8	Simple illustrations of RESCAL, DistMulti [6] . . . . .	19
2.9	Knowledge graph provides a platform to integrate disparate data . . . . .	21
2.10	The embedded representation of a knowledge graph can be used for different machine learning applications [7] . . . . .	22
4.1	Technology architecture for loan default prediction model . . . . .	31
4.2	Semantic data model . . . . .	33
4.3	Simple illustrations of the knowledge graph construction process from disparate data sources . . . . .	34
4.4	Part of a knowledge graph for loan default prediction . . . . .	35
4.5	A knowledge graph enables better feature engineering for machine learning . . . . .	37
5.1	Relationship diagram of the Home Credit Dataset [8] . . . . .	40
5.2	The distribution of loan repayment status . . . . .	41
5.3	The distribution of income sources of applicants . . . . .	42
5.4	Applicants’ income sources in terms of repaid or not (in percent) . . . . .	42
5.5	Default in repayment by age group . . . . .	43
5.6	The contract status in previous applications . . . . .	44
5.7	Top reasons for previous applications’ rejection . . . . .	44
5.8	Knowledge graph query results for two sample loan applicants . . . . .	47
5.9	A simple illustration of the ROC curve . . . . .	50
5.10	Comparison of the ROC curves of different ML classification models with and without KGE features . . . . .	52
5.11	Confusion matrix for ‘XGBoost + KGE’ model . . . . .	52

# List of Tables

2.1	A sample set of factual triples in a knowledge base [2]	13
3.1	Loan applicant's characteristics used in building typical credit scoring	25
4.1	Dataset description and key attributes	32
4.2	Explanations and supports for prediction made by the model	38
5.1	Hyper-parameters used for the training of KGE	47
5.2	Confusion matrix	48
5.3	A summary of evaluation metrics based on the confusion matrix	50
5.4	The performance comparison of machine learning model using knowledge graph embeddings	51



# List of Algorithms

1	Compute entity and relation embeddings . . . . .	16
---	--	----

## Abstract

Loan default risk, also known as credit risk, is one of the significant financial challenges in banking and financial institutions since it involves the uncertainty of the borrowers' ability to perform their contractual obligations. Banks and financial institutions rely on statistical and machine learning methods for loan default prediction to reduce the potential losses of issued loans. These machine learning applications may never achieve their full potential without the semantic context of the data.

A knowledge graph is a collection of linked entities and objects that include semantic information to contextualize them. Knowledge graphs allow machines to incorporate human expertise into their decision-making and provide context for machine learning applications. A Knowledge Graph can semantically incorporate various data and link knowledge from many areas without altering its original form, enabling organizations to leverage the power of collective intelligence. Furthermore, knowledge graph embedding is now a widely adopted technique for representing knowledge. This graph embedding preserves the original graph's semantic information and structure. It can be a beneficial source of features for a subsequent machine learning classification task. So, a knowledge graph-based approach will improve the prediction model's performance and interpretability.

In this thesis, we present a hybrid approach combining a knowledge graph and machine learning to enhance the performance and rationality of the loan default prediction model. For this purpose, we developed an ontology for the semantic data model. Then, we mapped our semantic data model with a publicly available credit dataset to construct the knowledge graph. Next, we used knowledge graph embedding methods to discover the knowledge graph's semantic and structural content. Finally, we inputted the vectors extracted from the graph embedding as features to the machine learning classifier to forecast loan default. The experimental results demonstrate that incorporating knowledge graph embedding as features can boost the performance of conventional machine learning classifiers in predicting loan default risk. To evaluate the performance of several machine learning classifiers that exhibited strong performance in the credit default prediction task, we employed accuracy, precision, recall, F1 score, MCC, and ROC AUC as evaluation metrics. The "XGBoost + KGE" model performed best in all evaluation measures, with a ROC AUC of 0.836 (an increase of around 10.14% over the conventional technique).

# Chapter 1

## Introduction

Over the decade, emerging and developing economies experience various financial system innovations, particularly in the banking sector. These include mobile banking, Internet banking, agent banking, and other digital financial services. But, a large part of the population is outside the financial inclusion [9]. Most banks and financial institutions still use financial repayment history-based credit decision systems where most people in the market do not even have a credit score. So, populations with low exposure to banking services face the risk of being completely excluded from financial ecosystems, mainly from access to credit.

Credit risk or loan default risk is one of the most significant financial challenges in banking and financial institutions. Credit risk refers to the uncertainty surrounding their borrowers' ability to repay a loan or fulfill the stipulated contractual obligations, resulting in loan default or bankruptcy. Retail lenders (providing loans to persons or retail consumers) and business lenders (offering loans to organizations or enterprises) are both exposed to credit risk [10]. The loan default risk mostly depends on the borrower's profile and credit parameters, including the solvency, credit type, maturity, loan amount, and other factors intrinsic in financial operations. This thesis presents a hybrid approach using a knowledge graph and machine learning to enhance the performance and rationality of the risk prediction of loan default.

### 1.1 Motivation

Nowadays, banks and financial organizations increasingly use machine learning applications in their day-to-day decision-making [11]. Without the semantic context in the data, these machine learning applications may never achieve their full potential [11]. A knowledge graph is a collection of linked entities and objects that include semantic information to contextualize them [12]. Knowledge graphs allow computers to integrate human knowledge into decision-making and provide context to the machine learning applications. Knowledge graph may semantically incorporate various data and link knowledge from many areas without altering its

original form [13]. As a result, it will enable organizations to leverage the power of collective intelligence. With a knowledge graph, the bank or lending organization can view and analyze all customer data, transactions, risk dimensions, and laws/regulations in one location, all correlated based on their significance for detailed analysis.

The feature engineering process is a necessary but time-consuming element of machine learning applications. The performance of most machine learning algorithms is highly dependent on the feature vectors' representation [14]. When there are numerous disparate data sources, one of the challenges for data scientists is wrangling the data. As a result, data scientists spend a significant amount of time collecting the data and making sense of it. A knowledge graph is also a key enabler for machine learning applications. It is a way to have far better improvements in feature engineering for machine learning.

Financial risk management researchers have increasingly used machine learning, support vector machines, and neural networks in recent years [2, 15–17]. Most of the models in the current researches only concentrate on prediction tasks, but relatively a few try to establish the causal relationship between the attributes and model prediction. Credit approval is a critical decision for the organizations, and both model creators and regulators want a causal explanation of the prediction model.

Knowledge graph embedding (KGE) [3, 18–20] is now a widely adopted technique for representing knowledge. KGE preserves the original graph's semantic information and structure. It can be a beneficial source of features for a subsequent machine learning classification task. So, a knowledge graph-based approach will improve the prediction model's performance and interpretability.

## 1.2 Objectives of the Thesis

The main objectives of our research work are:

- To design and develop a semantic data model for knowledge graph construction utilizing the principles and standards of the Financial Industry Business Ontology (FIBO). FIBO is an illustration of a conceptual business model for the financial industry.
- To construct a knowledge graph from a publicly available credit dataset. With this knowledge graph, the bank and financial institutions can view and analyze all customer data, transactions, risk dimensions, and laws/regulations in one location, all correlated based on their significance for comprehensive analysis.
- To utilize appropriate Knowledge graph embedding (KGE) techniques to embed the entities and relationships in low-dimensional vector spaces preserving the semantics of the original

graph. These vectors will directly be used as features to the downstream prediction task, i.e., Link Prediction/Triple Classification.

- To propose a loan default prediction model and demonstrate the effectiveness of using knowledge graphs for boosting the traditional machine learning model's performance.

### 1.3 Outline of Methodology

The following steps outline our proposed methodology and experimentation:

- We created an ontology-based semantic data model. A semantic data model has the advantage of connecting and integrating disparate data sources, harmonizing them, and then exporting them to a variety of target sources. With a semantic data model, we can represent data in a single interchangeable format, such as RDF, so that both machines and humans can understand it.
- We constructed the knowledge graph using the credit dataset (publicly available) mapped with our semantic data model to obtain entities, entity features, and relationships within entities. Nodes represent entities in the graph, and the edges connecting nodes indicate their relationships. After knowledge extraction and validation, we stored the knowledge graph in the graph database.
- Machines cannot directly access the knowledge graph represented by symbols to perform computations. So, we adopted Knowledge Graph Embedding (KGE) techniques for representing knowledge that embeds entities and relationships in low-dimensional vector spaces. The vectors preserve the original graph's semantic and structural content. Next, we formulate the loan default risk prediction as a classification problem within the KGE space by computing similarities (link prediction/triple classification) between loan applicants. Lastly, we will input the vectors extracted from the graph embedding as features to the machine learning classifier to forecast loan default.
- Finally, we conduct extensive experimentation on our proposed approach to compare it with the different machine learning methods. We use the receiver operator characteristic (ROC) curve, accuracy, precision, recall, F1 score, and MCC as evaluation metrics. For performance evaluation, we consider popular ML classification models that have exhibited strong performance in credit default prediction tasks, such as logistic regression (as a baseline), random forest (RF), Light Gradient Boosting Machine (LightGBM), and extreme gradient boosting algorithm (XGBoost).

The specific steps of the proposed methodology are elaborated in Chapter 4

## 1.4 Research Contribution

The major research contributions of this thesis are as follows:

- A semantic data model to show that consumer attributes and financial states can be mapped and put into a knowledge graph embedding space for the consumption of machine learning applications to make predictions.
- A knowledge-graph-based loan default prediction model to demonstrate that it can boost the traditional machine learning model's performance in terms of accuracy, precision, recall, F1 score, MCC, and ROC AUC. The knowledge graph makes this model's predicted outcome interpretable by the model creators, regulators, and loan applicants.
- A comprehensive analysis and comparison of our proposed approach with other state-of-the-art machine learning classifiers in the risk prediction of loan default.

## 1.5 Organization of the Thesis

The remaining chapters of this thesis are organized as follows:

Chapter 2 describes the relevant concepts and necessary definitions of terminologies used in this thesis to understand our research work.

Chapter 3 presents the existing research related to the risk prediction of loan default. It also highlights the scope of works based on the limitations found in the current literature.

Chapter 4 provides a detailed discussion of the overall methodology of the research. The technique for constructing the solution to the problem and the components of the solution are also discussed here.

Chapter 5 covers our experimental work, including data preprocessing, experimental settings, training of knowledge graph embedding, and results of the experiment with comparative analysis.

Chapter 6 concludes the thesis, highlighting our contributions and possible directions for future work.

# Chapter 2

## Preliminaries

In this chapter, we define some basic terms and concepts related to financial risk, semantic web, and knowledge graph technology. Concepts or definitions not covered in this chapter will be introduced as they are needed.

### 2.1 Financial Risk

The financial industry is vulnerable to risks of diverse dimensions due to its direct exposure to many sectors of the economy and the cross-border implications inherent in its activities. The financial performance of an organization can be impacted by a variety of events such as financial market downturns, loan defaults, unanticipated insurance claims, fraud, and customer churn [21]. Depending on the source of risk factors, the risk areas in the financial industry are broadly categorized as *market risk*, *credit risk*, and *operational risk*. The core risks in the financial industry can be organized into taxonomies as shown in the Figure 2.1.

#### 2.1.1 Market Risk

Market risk refers to the uncertainties in the value of the company's underlying assets, liabilities, or income due to exposure to a highly dynamic financial market [21, 22]. From the standpoint of an investor, it is the likelihood of a loss due to the factors affecting the overall performance of the financial markets in which the investor has made investments. For banking and financial institutions, market risk mostly occurs from their activities in capital markets. Market risk includes *price risk* that arises from changes in the value of trading positions in the interest rate, foreign exchange, equity, and commodities markets.

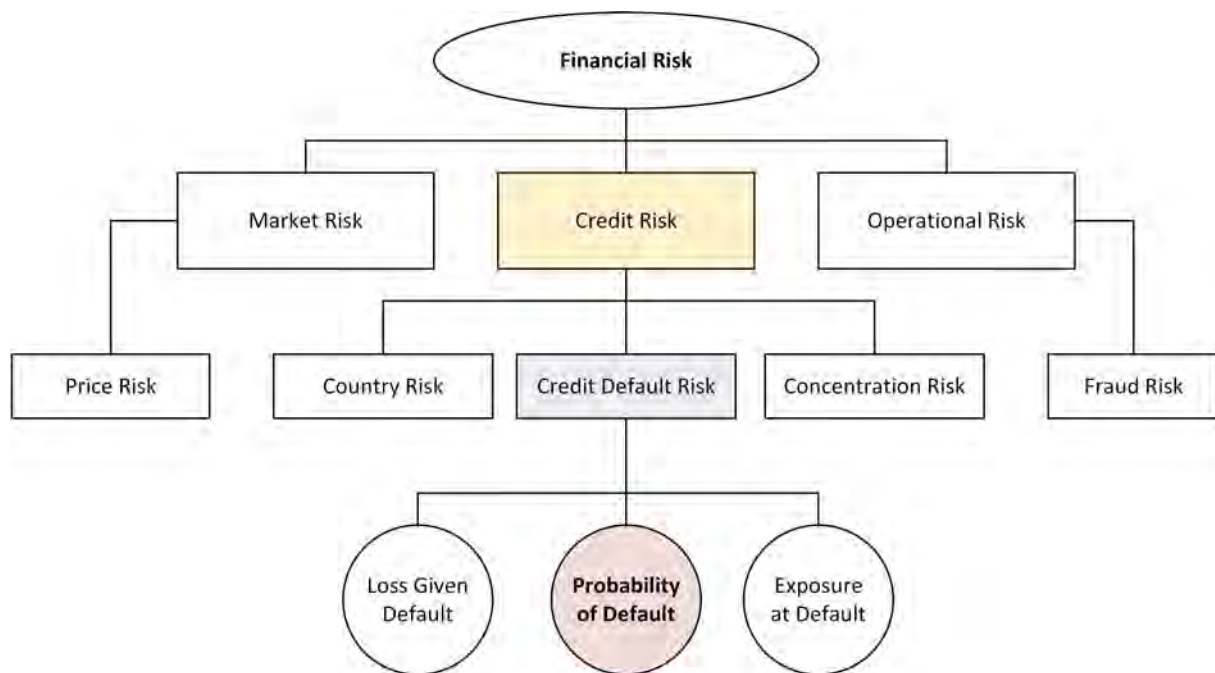


Figure 2.1: Taxonomy of financial risks

### 2.1.2 Credit Risk

Credit risk is the possibility of a loss resulting from a borrower failing to repay a loan or meet contractual obligations [23]. It is the biggest risk for banking and financial institutions. The following are the main types of credit risk:

- *Credit default risk* occurs when there is a chance that a borrower may stop making payments on a loan as outlined in the lending agreement, i.e., the borrower is unable to pay the loan obligation in full. Credit default risk may affect all credit-sensitive financial transactions such as loans, bonds, securities, and derivatives. The level of default risk can change due to a broader economic change. It can also be due to a change in a borrower's economic situation, such as increased competition or a recession, which can affect the company's ability to set aside repayments for the loan.
- *Concentration risk* arises from exposure to a single counterparty or sector, and it offers the potential to produce large amounts of losses that may threaten the lender's core operations. The risk results from the observation that more concentrated portfolios lack diversification; therefore, the returns on the underlying assets are more correlated. For example, a corporate borrower who relies on one major buyer for its main products has a high level of concentration risk and has the potential to incur a large amount of losses if the main buyer stops buying their products.
- *Country risk* occurs when a country freezes foreign currency payment obligations, resulting in a default on its commitments. The risk is associated with the country's political



instability and macroeconomic performance, which may adversely affect the value of its assets or operating profits. The changes in the business environment will affect all companies operating within a particular country.

To minimize the level of credit risk, lenders should forecast credit risk with greater accuracy. Listed below are some of the factors that lenders usually consider when estimating the level of credit risk:

- *Probability of default (POD)* is the likelihood that a borrower will default on their loan obligations. For individual borrowers, POD is based on a combination of two factors, i.e., credit score and debt-to-income ratio. On the other hand, the POD for corporate borrowers is obtained from the credit rating agencies [24].
- *Loss given default (LGD)* refers to the amount of loss that a lender will suffer in the event that a borrower defaults on the loan [24]. For example, assume that two borrowers, A and B, have the same debt-to-income ratio and an identical credit score. Borrower A takes a loan of BDT 10,000 while B takes a loan of BDT 200,000. The two borrowers present with different credit profiles, and the lender stands to suffer a greater loss when Borrower B defaults since the latter owes a larger amount. Although there is no standard practice for calculating LGD, lenders consider an entire portfolio of loans to determine the total exposure to loss.
- *Exposure at default (EAD)* evaluates the amount of loss exposure that a lender is exposed to at any particular time, and it is an indicator of the risk appetite of the lender [24]. EAD is an important concept that refers to both individual and corporate borrowers. It is calculated by multiplying each loan obligation by a specific percentage that is adjusted based on the particulars of the loan.

### 2.1.3 Operational Risk

Operational risk is the risk of indirect or direct losses caused by flawed or failed processes, policies, systems, or events that disrupt business operations [23]. Employee errors, criminal activity such as fraud, and physical events are among the factors that can trigger operational risk. It is considered inherent in all banking products, activities, processes and systems [23]. *Fraudulent activities* are major sources of operational risk for the finance sector. It can take many forms, for example, fraudulent credit card transactions [25], money laundering activities [26], and fraudulent insurance claims [27]. Financial fraud can be economically devastating for a business. Therefore, financial fraud detection systems are becoming increasingly important for effective and timely fraud prevention [28].

## 2.2 Semantic Web

The Semantic Web [29–31] is a vision of an extension of the existing World Wide Web that provides software programs with machine-interpretable metadata for all published content and data. In simple terms, structuring the entire knowledge base on the traditional web in a machine-readable format. It enables data to be linked from a source to any other source and to be understood by computers so that they can perform increasingly sophisticated tasks on our behalf. The word “semantic” implies meaning or understanding. As such, the basic distinction between Semantic Web technologies and other data-related technologies (such as relational databases or the World Wide Web itself) is that the Semantic Web is concerned with the meaning of data rather than the structure of data. According to the World Wide Web Consortium (W3C) [32], “The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries.” The term was coined by Tim Berners-Lee (the inventor of World Wide Web) for a web of data that can be processed by machines.

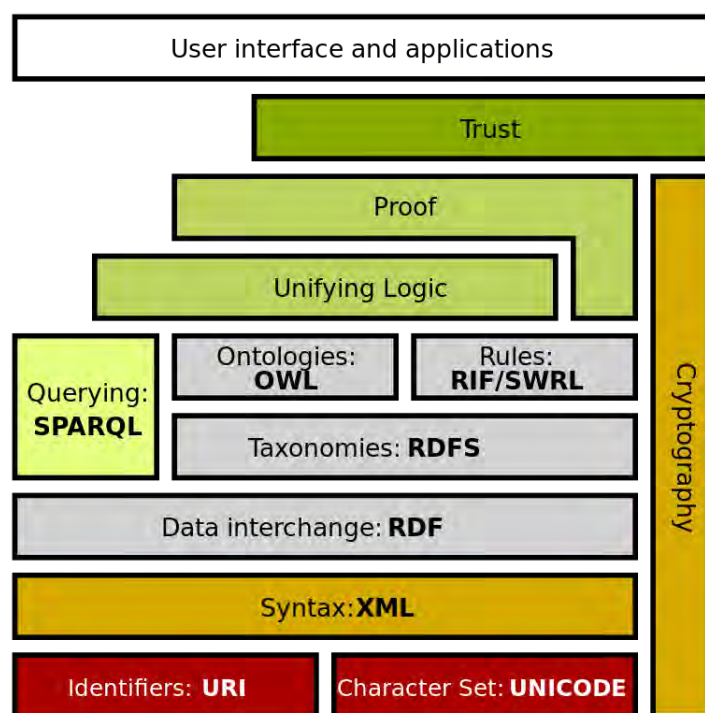


Figure 2.2: Semantic Web technology stack [1]

Figure 2.2 illustrates the layered implementation of semantic web. The Extensible Markup Language (XML) is used at the bottom layer for information exchange between applications and computers. The next top layer is the Resource Description Framework (RDF) [30, 33] which describes information in simple triple statements (subject, object, and their relationship). Any object can be described by the triples, provided it has a Uniform Resource Identifier (URI) i.e. resource name. RDF Schema layer helps to build ontologies [30, 33]. The ontology layer is used for deriving meaning from the RDF statements, which are subsequently mapped by a logic

layer to assert the relevant knowledge. The Proof layer and Trust layer validate the knowledge and generate digital signatures, respectively.

### 2.2.1 Resource Description Framework

The Resource Description Framework (RDF) is a framework for expressing information about resources [34]. Resources can be anything, including documents, people, physical objects, and abstract concepts. RDF provides a common framework for expressing interconnected data so that it can be exchanged between applications without any loss of meaning. An RDF statement expresses a relationship between two resources. The structure of these statements is a simple triple (three elements), as shown in Figure 2.3. The *subject* and the *object* represent the two resources being related; the *predicate* represents the nature of their relationship. The relationship is stated in a directed manner (from subject to object) and is referred to as a *property* in RDF [34].



Figure 2.3: Structure of a RDF statement

The following are a few examples of RDF triples (informally expressed for better understanding):

```
<Bob>          <is a>          <person>.
<Bob>          <is a friend of> <Alice>.
<Bob>          <is born on>    <14 July 1990>.
<Bob>          <is a customer of> <Bank>.
<Bank>         <located in>    <Bangladesh>.
<Financial Services> <provided by> <Bank>.
```

In RDF statements, the same resource can be referenced in multiple triples. In the example shown above, *Bob* is the subject of four triples, and *Bank* is the subject of one and the object of two triples. This ability to have the same resource be in the subject position of one triple and the object position of another makes it possible to find connections between triples, which is a vital part of RDF's power. RDF triples represent facts, relationships, and data by connecting resources of different kinds. So, it enables organizations to connect and interlink various datasets to perform cross-dataset queries using SPARQL, a query language to retrieve and manipulate data stored in RDF format.

We can visualize triples as a connected graph. Graphs consist of nodes and arcs. The subjects and objects of the triples make up the nodes in the graph; the predicates form the arcs. Figure 2.4 shows the graph resulting from the sample triples used in the above example. We will discuss this in details in Section 2.3.

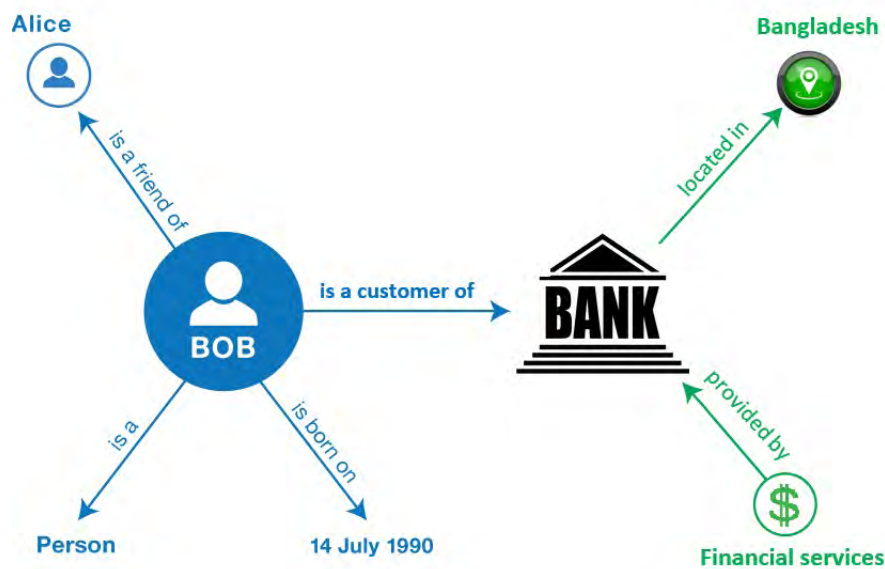


Figure 2.4: Sample RDF triples in a graph

### 2.2.2 Ontology

An ontology is a formal, explicit specification of knowledge as a set of concepts within a domain and the relationships that hold between them [35]. Ontology plays an essential role in solving the problem of interoperability between applications across different organizations by providing a shared understanding of common domains, enabling semantic interoperability among heterogeneous and widely spread application systems. Artificial intelligence (AI) research communities have been using ontologies for a long time to facilitate knowledge sharing and reuse. One of the important features of ontologies is that they enable automated data reasoning by incorporating the critical relationships between concepts. As a result, ontologies do not only introduce a sharable and reusable knowledge representation but can also add new knowledge about the domain. Furthermore, ontologies work in the same way as a “brain”. They use concepts and relationships to “work and reason” in ways that are similar to how humans see interconnected things. Another notable feature is that ontologies can be easily extended by adding relationships and concepts to the existing ontologies. An ontology consists of four fundamental components: concepts, instances, relations, and axioms. The following are the definitions of these ontology components:

- **A concept** (also known as a class) is an abstract group, set, or collection of things. It is the core building block for an ontology and usually represents a group or class whose members share common characteristics. Any entity that has the same characteristics as other entities is part of the same concept or class. Classes can be represented in hierarchical graphs, giving them an object-oriented aspect. A “super-class” represents the higher or “parent class”, while a “subclass” represents the lower or “child class”. For example, a financial

institution could be represented as a class with many subclasses, such as banks, financial products, and services.

- **An instance** (alternatively referred to as an individual) is the “ground-level” component of an ontology that represents a particular object or a member of a class or concept. For instance, “Bangladesh” could be an instance of the class “SAARC country” or simply “country”.
- **A relation** is a way to explain the relationship between two concepts in a specific domain. For instance, “provide” is the relationship that establishes the link between the “bank” and “financial service” concepts. This relation, in terms of the Semantic Web, is defined as a property.
- **An axiom** is used to impose constraints on the values of classes or instances. Hence, axioms are usually represented in logic-based languages like first-order logic, and they are used to validate the ontology’s consistency.

The Web Ontology Language (OWL) [36] is a W3C recommendation for expressing ontologies. It contains a well-defined syntax and semantics that are intended to represent knowledge about things or objects and the relations between them. OWL also provides detailed, consistent, and meaningful distinctions between classes, properties, and relationships.

### 2.2.3 Ontology Mapping

Ontology mapping [37] is a process that defines the semantic relationships between entities from different ontologies. It is an integral part of many ontology application domains, such as the Semantic Web and e-commerce, which are used to interlink heterogeneous data sources through some common concepts. In other words, it is the process of discovering similarities between two ontologies. Mapping the two ontologies,  $O1$  onto  $O2$ , means that each entity in ontology  $O1$  is trying to find a corresponding entity that has the same intended meaning in ontology  $O2$ . For example, the entity names “client” and “customer” hold the same meaning in different ontologies. As more and more ontologies are being developed, using existing ontologies becomes increasingly essential. Ontology mapping facilitates the exchange of information between different domains by extending and combining existing ontologies.

## 2.3 Knowledge Graph

A knowledge graph is a specific type of graph with an emphasis on contextual understanding. Knowledge graphs are structured representations of facts, consisting of entities, relationships, and semantic descriptions in a human- and machine-understandable format [2]. A knowledge graph

can be represented as a directed graph that consists of nodes, edges, and labels. Any real-world objects, events, or things can act as a node, such as people, places, companies, computers, etc. An edge connects a pair of nodes and captures the relationship of interest between them, for example, a customer relationship between a company and a person, or a network connection between two computers. The labels capture the well-defined meaning of the relationship; for example, the friendship relationship between two people.

More formally, given a set of nodes or entities  $E$ , and a set of labels or relations  $R$ , a knowledge graph  $G$  is a subset of the cross product  $E \times R \times E$ . Each member of this set is referred to as a triple in the form of “*Entity – Relation – Entity*”. This triple is expressed as (*head, relation, tail*) or (*subject, predicate, object*) as per the RDF data model shown in the Figure 2.3.  $F$  is a set of edges representing facts connecting pairs of entities. Each fact is a triple  $\langle h, r, t \rangle$ , where  $h$  is the head,  $r$  is the relation, and  $t$  is the tail of the fact. Hence, we can define a knowledge graph with the below notations.

$$\begin{aligned}
 G &= (E, R, F) \\
 E &= \{e_1, e_2, e_3, \dots, e_{|E|}\} \\
 R &= \{r_1, r_2, r_3, \dots, r_{|R|}\} \\
 F &\subseteq E \times R \times E \\
 (h, r, t) &\in F
 \end{aligned}$$

Knowledge graphs have been around for a long time, but were made popular in 2012 when Google announced that they were using knowledge graph technology to enhance the search engine’s capability. The basic motto behind the Google Knowledge Graph was to make search about *things not strings* [38].

Ontologies provide the terminology definitions for knowledge graphs and act as the data schema for the graph. They serve as a formal contract between the knowledge graph’s creators and its consumers (i.e., a user or a computer system) so that they can understand the meaning of the underlying data that adds connected context to support reasoning and knowledge discovery. When formal semantics are included in the data model, they can be used as a knowledge base for interpreting and drawing inferences about facts and events [39]. Figure 2.5 illustrates an example of a knowledge graph built from a sample set of factual triples in the knowledge base shown in Table 2.1.

Knowledge graphs can be of two types: generic (open world, or domain-independent) and domain-specific. *Generic knowledge graphs* have been continuously constructed even before coining the term “knowledge graph”. In fact, since the Semantic Web’s inception, generic knowledge graphs have been associated with Linked Open Data (LOD) as a natural representation for interconnected entities [12]. Examples of notable generic or open-source knowledge graphs are FreeBase [40], WikiData [41], DBpedia [42], and Yago [43]. These massive knowledge graphs can contain

Table 2.1: A sample set of factual triples in a knowledge base [2]

Subject (head)	Predicate (relation)	Object (tail)
Albert Einstein	BornIn	German Empire
Albert Einstein	SonOf	Hermann Einstein
Albert Einstein	GraduateFrom	University of Zurich
Albert Einstein	WinnerOf	Nobel Prize in Physics
Albert Einstein	ExpertIn	Physics
Nobel Prize in Physics	AwardIn	Physics
The theory of relativity	TheoryOf	Physics
Albert Einstein	SupervisedBy	Alfred Kleiner
Alfred Kleiner	ProfessorOf	University of Zurich
The theory of relativity	ProposedBy	Albert Einstein
Hans Albert Einstein	SonOf	Albert Einstein

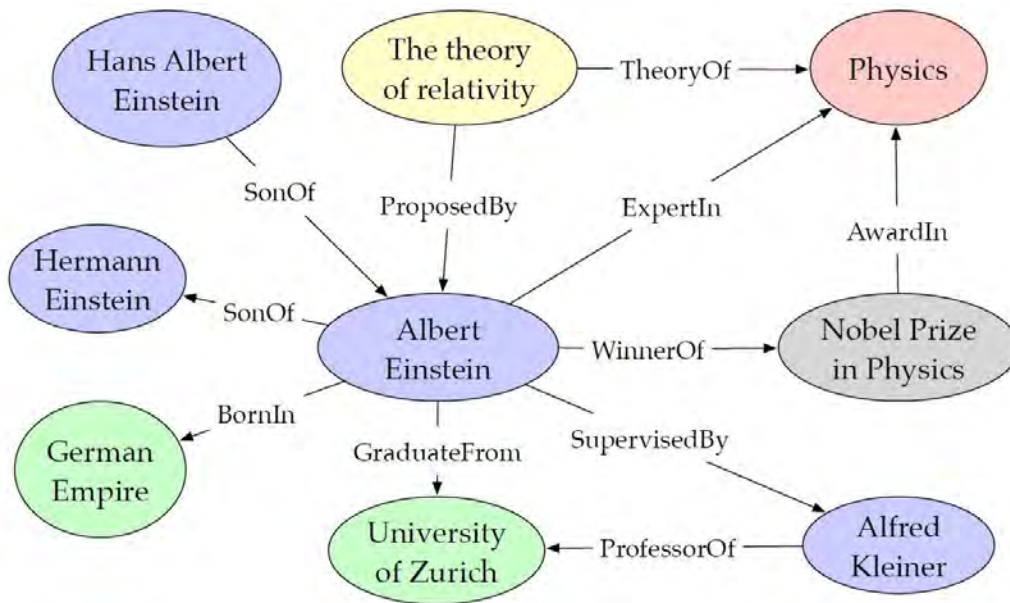


Figure 2.5: An example of knowledge graph built from the knowledge base of Table 2.1 [2]

millions of entities and billions of facts. On the other hand, *domain-specific knowledge graphs* are built from the underlying ontology to represent semantically interrelated entities and relations of a particular subject area [44]. Knowledge graphs are recognized by academia and industry as an efficient approach to data governance, metadata management, and data enrichment. They are widely employed in various domains, ranging from question answering to information retrieval, content-based recommendation, and financial fraud detection systems [45].

### 2.3.1 How to Build Knowledge Graphs?

Knowledge graph construction is an iterative engineering process where many methods and tools can be applied. From the perspective of knowledge graphs based on ontology, there are two main approaches to creating knowledge graphs [46]. One is top-down, and the other is bottom-up. *The top-down approach* means that the ontology and schema should be defined first, and then knowledge instances should be added to the knowledge base. This approach emphasizes the use of well-defined domain ontologies to represent the actual instances of knowledge graphs. *The bottom-up approach* extracts knowledge instances from the Linked Open Data (LOD) or other knowledge resources such as social media and crowd-sourced data. Although the bottom-up approach can process a large number of datasets and quickly build a big knowledge graph, the precise logical representation and assertions for the entities and relationships contained in the resulting graph remain a challenge [47].

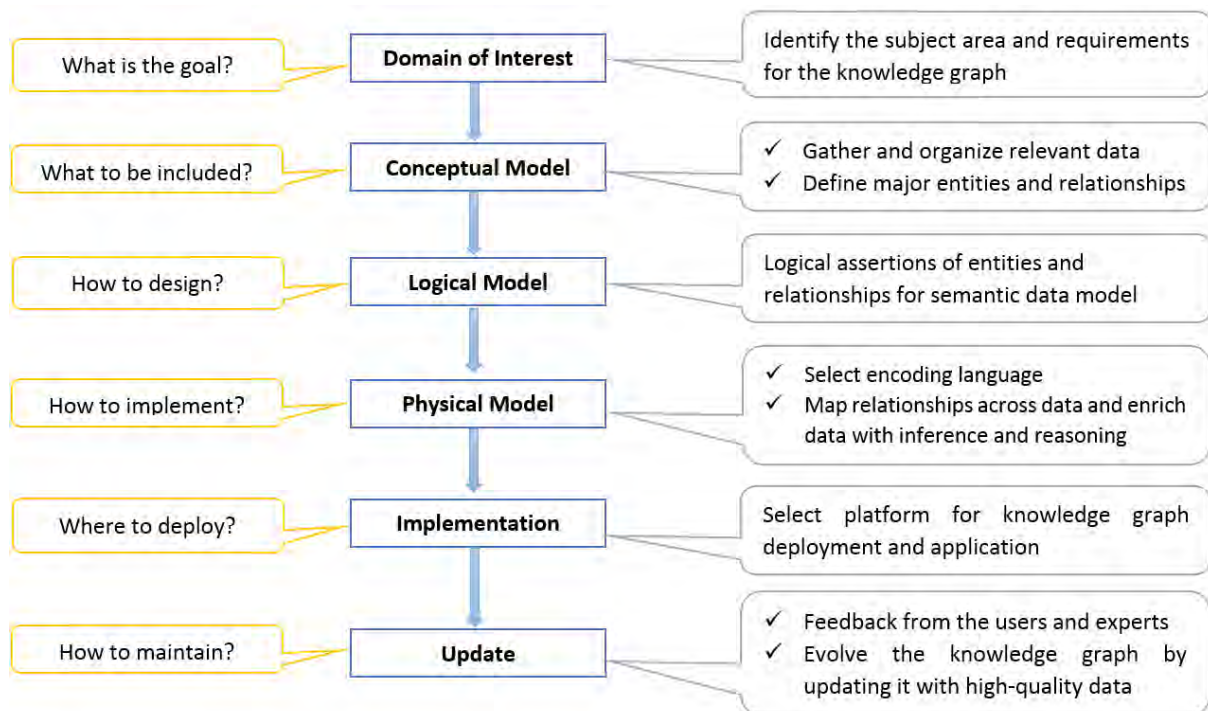


Figure 2.6: A top-down approach for the construction and implementation of knowledge graphs

In this thesis, we used the top-down approach for knowledge graph construction. Figure 2.6 depicts the steps of the top-down approach. First, a subject domain and a list of research requirements are identified. Second, a conceptual model will be designed to collect the entities of interest, their inter-relationships, and the categories. Third, the logical and physical models will add logical representation and assertions to the collected entities and relationships to develop a semantic data model. Fourth, the technical development and implementation need to consider the coding language to use (e.g., RDF and OWL), the serialization formats (e.g., RDF/XML, Turtle, and JSON-LD), and ontology development tools such as Protégé [48]. The last step is to



deploy the knowledge graph as a service to allow the stakeholders (i.e., related users and experts) to use it and provide feedback. In general, this is a process to transform the knowledge in the domain expert's brain to a machine-readable representation [47].

## 2.4 Knowledge Graph Embedding

The knowledge graph is a multi-relational graph (where nodes are entities and edges are relations) represented by a set of triples. For example,  $(Dhaka, IsCapitalOf, Bangladesh)$  is a triple where *Dhaka* and *Bangladesh* are entities, and *IsCapitalOf* is the relation between them. Although the representation appears to be scientific, utilizing it in real-world applications is challenging due to the heavy reliance on machine learning and deep learning technologies in these applications. These machine learning applications don't operate well with strings and symbols, they need numbers or numeric representations to produce the optimum outcomes. *Knowledge graph embedding* is a solution to incorporate the knowledge from the knowledge graph into a real-world application.

The motivation behind knowledge graph embedding (KGE) is to preserve the graph's structural information and semantic meaning, i.e., the relations between entities, and represent it in a low-dimensional vector space [18]. KGE, also referred to as knowledge representation learning (KRL), is characterized by four different aspects [2].

1. *Representation space*: the low-dimensional space in which entities and relations are represented.
2. *Scoring function*: a measure for assessing the quality of a triple embedded representation.
3. *Encoding models*: the modality in which the embedded representation of the entities and relations interact with each other.
4. *Auxiliary information*: any additional information coming from the knowledge graph that can enrich the embedded representation. Usually, an ad hoc scoring function is integrated into the general scoring function for each additional piece of information [49].

To learn the semantic meaning of the facts in knowledge graph, all of the different KGE models follow a similar kind of procedure [50]. First of all, for a knowledge graph with a set of triples in the form of  $(h, r, t)$  representing *(head, relation, tail)*, the embedding vectors of the entities and relations are initialized to random values. Then, beginning from a training set until a stop condition is reached, the algorithm continuously optimizes the embeddings. Usually, the stop condition is given by overfitting of the training set [50]. For each iteration, a batch of size  $b$  from the training set is sampled, and for each triple in the batch, a random corrupted fact (a triple that does not represent a true fact in the knowledge graph) is sampled. The corruption of a triple

involves substituting the *head* or the *tail* (or both) of the triple with another entity that makes the fact false. The original triple and the corrupted triple are added in the training batch, and then the embeddings are updated, optimizing the scoring function [49]. At the end of the algorithm, the learned embeddings should have extracted semantic meaning from the triples and correctly identified unseen true facts in the knowledge graph. The pseudocode for the general embedding procedure [18, 50] is shown in Algorithm 1.

---

**Algorithm 1** Compute entity and relation embeddings
 

---

**Input:** The training set  $S = \{(h, r, t)\}$ ,  
 entity set  $E$ ,  
 relation set  $R$ ,  
 embedding dimension  $k$

**Output:** Entity and relation embeddings

*initialization:* the entities  $e$  and relations  $r$  embeddings (vectors) are randomly initialized

**while** stop condition **do**

$S_{batch} \leftarrow \text{sample}(S, b)$  // from the training set randomly sample a batch of size  $b$

**for each**  $(h, r, t)$  in  $S_{batch}$  **do**

$(h', r, t') \leftarrow \text{sample}(S')$  // sample a corrupted fact or triple

$T_{batch} \leftarrow T_{batch} \cup \{((h, r, t), (h', r, t'))\}$

**end for**

Update embeddings by minimizing the loss function

**end while**

---

The KGE model encodes the knowledge graphs' topology (nodes, edges, and their relationships) into a low-dimensional continuous vector space for the consumption of real-world applications. KGE models are broadly classified into two groups: *translational models* and *semantic matching models*. We will describe KGE models that are relevant to this work in the following sections. However, many more models are available in the literature with their pros and cons.

### 2.4.1 Translational Models

The translational models use distance-based measures (e.g., Euclidean distance) to generate the similarity score for a pair of entities and their relationships [5].

**TransE** [18] is the pioneering KGE model that is still widely used due to its competitive performance. It is an *energy-based* model that represents entities and relations in the same space, say  $d$ -dimension vector space  $\mathbb{R}^d$ . If  $(h, r, t)$  holds, then the embedding of the tail entity  $t$  should be close to the embedding of the head entity  $h$  plus some vector that depends on the relationship  $r$ . Mathematically, TransE attempts to generate an embedding for each  $h, r$ , and  $t$  such that for triples observed in the knowledge graph, the following translation relationship  $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$  should hold. It can be graphically represented as Figure 2.7(a). The following scoring function

is defined to measure the plausibility of a triple:

$$f_r(h, t) = - \|h + r - t\|_{1/2} \quad (2.1)$$

The loss function as shown in Equation (2.2) is optimized so that the valid triples are ranked above the corrupt triples.

$$\min_{\Theta} \sum_{(h,r,t) \in F} \sum_{(h',r,t') \in F'} \max(0, f_r(h, t) + \gamma - f_r(h', t')) \quad (2.2)$$

Here  $\gamma$  is a margin. The corrupt triple  $(h', r, t')$  is constructed by randomly changing a head or tail entity or both entities in the knowledge graph. Most translation-based embedding methods use this margin-based loss function [51]. A key strength of TransE is its reliance on a reduced set of parameters since it learns only one low-dimensional vector for each entity and each relationship. The energy-based optimization function (based on translation) is also simple and intuitive to understand. A range of alternatives (commonly referred to as Trans\*) have been built using the same fundamental principles of TransE but with richer optimizations and information sets [44]. Despite its simplicity and efficiency, TransE has flaws in dealing with 1-to-N, N-to-1, and N-to-N relations [3, 4].

Assuming the ideal case of no-error embedding where  $\mathbf{h} + \mathbf{r} - \mathbf{t} = \mathbf{0}$  if  $(h, r, t) \in F$ , we can get the following outcomes directly from the TransE model.

- If relation  $r$  is a 1-to-N relation, i.e.,  $\forall i \in \{0, \dots, m\}$  such that  $(h, r, t_i) \in F$ , then  $\mathbf{t}_0 = \dots = \mathbf{t}_m$ .
- If relation  $r$  is a N-to-1 relation, i.e.,  $\forall i \in \{0, \dots, m\}$  such that  $(h_i, r, t) \in F$ , then  $\mathbf{h}_0 = \dots = \mathbf{h}_m$ .
- If  $(h, r, t) \in F$  and  $(t, r, h) \in F$ , i.e.,  $r$  is a reflexive relation, then  $\mathbf{r} = \mathbf{0}$  and  $\mathbf{h} = \mathbf{t}$ .

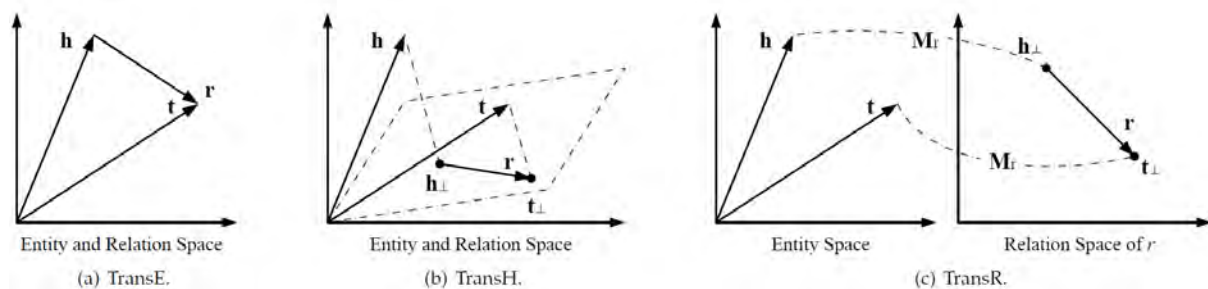


Figure 2.7: Simple illustrations of TransE, TransH, and TransR [3–5]

**TransH** [3] is an extension of TransE to address the limitations of TransE in modeling N-to-N relations. It proposes that an entity may have multiple roles in different relations. This model

uses an additional hyperplane to represent relations. Then, the translation from the head to the tail entity is performed in that relation-specific hyperplane as shown in Figure 2.7(b). Given a fact  $(h, r, t)$ , the entity representations  $h$  and  $t$  are first projected onto the hyperplane as per the below formulation:

$$h_{\perp} = h - w_r^T h w_r, \quad t_{\perp} = t - w_r^T t w_r \quad (2.3)$$

The projections are then assumed to be connected by  $r$  on the hyperplane with low error if  $(h, r, t)$  holds, i.e.,  $\mathbf{h}_{\perp} + \mathbf{r} \approx \mathbf{t}_{\perp}$ . The scoring function is accordingly defined as follows similar to the one used in TransE.

$$f_r(h, t) = - \|h_{\perp} + r - t_{\perp}\|_2^2 \quad (2.4)$$

**TransR** [19] also followed the basics of TransE as an extension of TransH with the difference that it introduces relation-specific spaces rather than hyperplanes. Figure 2.7(c) presents the intuition behind the TransR model. In this model, relations are in the matrix representation of  $M_r$  which takes entities projected into the relational specific space:

$$h_{\perp} = M_r h, \quad t_{\perp} = M_r t \quad (2.5)$$

Based on this representation, the score function is designed as follows:

$$f_r(h, t) = - \|h_{\perp} + r - t_{\perp}\|_2^2 \quad (2.6)$$

This model is capable of handling complex relations as it uses different spaces. However, its computation is extremely expensive due to the large number of parameters required to maintain the projection matrix for each relation [5]. As a result, it lacks the simplicity and efficiency of TransE/TransH (which model relations as vectors and require only  $\mathcal{O}(d)$  parameters per relation).

## 2.4.2 Semantic Matching Models

Semantic matching models exploit similarity-based scoring functions. They measure the plausibility of facts or triples by matching the latent semantics of entities and relations embodied in their vector space representations. Several KGE models fall into this category; we will discuss a few of the best-performing ones related to our work.

**RESCAL** [52] is a bilinear model [53] associates each entity with a vector to capture its latent semantics. Each relation is represented as a matrix that models pairwise interactions

between latent factors. The score of a fact  $(h, r, t)$  is defined by the following bilinear function

$$f_r(h, t) = h^T M_r t = \sum_{i=0}^{d-1} \sum_{j=0}^{d-1} [M_r]_{ij} \cdot [h]_i \cdot [t]_j \quad (2.7)$$

where  $h, t \in \mathbb{R}^d$  are vector representation of entities, and  $M_r \in \mathbb{R}^{d \times d}$  is a matrix representation of  $r^{th}$  relation. Thus, from this equation, we can calculate the score of the triple using the weighted sum of all the pairwise interactions between the latent features of the entities  $h$  and  $t$  as shown in Figure 2.8(a).

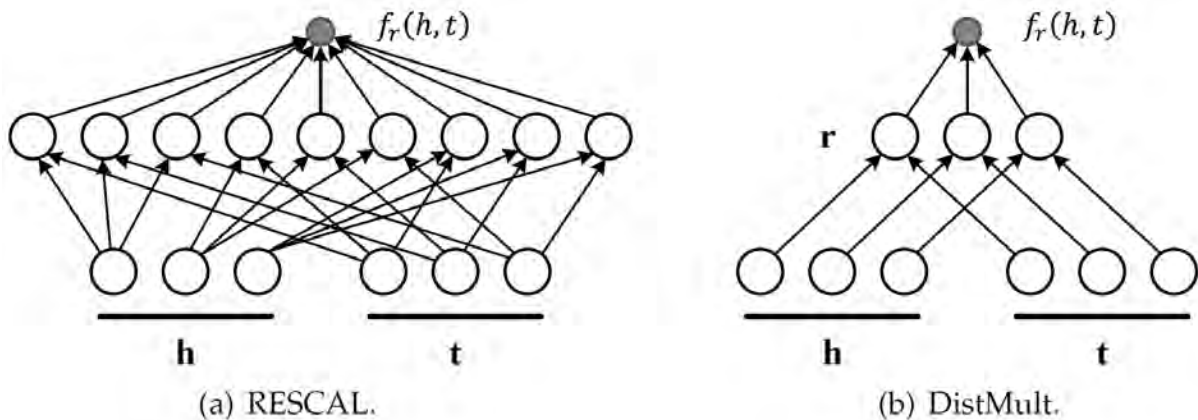


Figure 2.8: Simple illustrations of RESCAL, DistMulti [6]

**DistMult** [54] is a model that focuses on capturing the relational semantics and the composition of relations as characterized by matrix multiplication. This model considers learning representations of entities and relations within the underlying knowledge graph. It simplifies RESCAL by limiting  $M_r$  to diagonal matrices. For each relation  $r$ , it introduces a vector embedding  $r \in \mathbb{R}^d$  and requires  $M_r = \text{diag}(r)$ . The scoring function is accordingly defined as:

$$f_r(h, t) = h^T \text{diag}(r) t = \sum_{i=0}^{d-1} [r]_i \cdot [h]_i \cdot [t]_i \quad (2.8)$$

As shown in the Figure 2.8(a), this score captures pairwise interactions between only the components of  $h$  and  $t$  along the same dimension and reduces the number of parameters to  $\mathcal{O}(d)$  per relation. The restriction to diagonal matrices makes DistMult more computationally efficient than RESCAL but less expressive. However, this model can only deal with symmetric relations, which is not powerful enough for general knowledge graphs.

**ComplEx** [20] is an extension of DistMult to better model asymmetric relations. It can be observed from the scoring function of DistMult that it has a limitation in representing anti-symmetric relations since  $h^T \text{diag}(r)t$  is equivalent to  $t^T \text{diag}(r)h$ . Symmetric, reflexive, anti-reflexive, and transitive relations can all be represented using the dot product of vector embedding of KG triplets [55]. However, it can't be used for anti-symmetric relations. For example, consider a triple  $(Dhaka, IsCapitalOf, Bangladesh)$ . Here, the relation  $IsCapitalOf(Dhaka, Bangladesh)$  is not symmetric since we cannot interchange subject and object entities in this relation. Therefore, we need to have different embedding for a subject entity and an object entity, which increases the number of parameters. ComplEx embedding facilitates joint learning of subject and object entities while preserving the asymmetry of the relation. It uses Hermitian dot products for embedding subject entities and object entities. In ComplEx, entity and relation embeddings  $h, r, t$  no longer lie in a real space but a complex space, say  $\mathbb{C}^d$ . The score of a fact  $(h, r, t)$  is defined as:

$$f_r(h, t) = \text{Re}(\mathbf{h}^T \text{diag}(\mathbf{r})\bar{\mathbf{t}}) = \text{Re}\left(\sum_{i=0}^{d-1} [\mathbf{r}]_i \cdot [\mathbf{h}]_i \cdot [\bar{\mathbf{t}}]_i\right) \quad (2.9)$$

where  $\bar{\mathbf{t}}$  is the conjugate of  $\mathbf{t}$  and  $\text{Re}(x)$  means taking the real part of a complex value. This scoring function is not symmetric anymore, i.e.,  $f_r(h, t) \neq f_r(t, h)$ , and facts from asymmetric relations can receive different scores depending on the order of entities involved.

## 2.5 Applications of Knowledge Graphs

There are numerous applications of knowledge graphs in both research and industry. Their ability to provide semantically organized information has the potential to solve a wide range of problems, including question answering, recommendation, and information retrieval [56]. For our discussion here, we have chosen to focus on applications in the financial industry and the use of knowledge graphs as input to machine learning.

### 2.5.1 Organizational Data Governance and Integration Platform

Financial institutions face tremendous challenges in their legacy data environment as different parties participate in defining the data, resulting in data silos across their organizational units. There will be inconsistencies in the data quality and usefulness as different departments within a financial corporation generate new data over time. Thus, the significance of *data governance* is in managing these disparate contexts and data silos, tracking data lineage, ownership, and usage patterns while focusing on the value, quality, and usability of data.

Knowledge graphs can be used in *data governance* to centralize knowledge across “diverse datasets” and keep them up-to-date as more data comes in. They can act as a semantic layer,

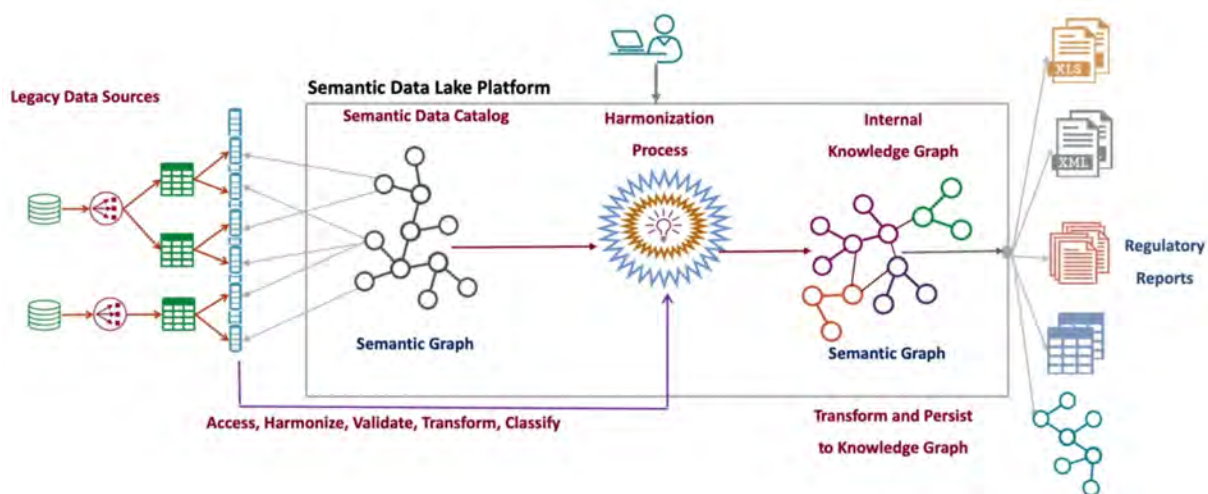


Figure 2.9: Knowledge graph provides a platform to integrate disparate data

modeling metadata and adding rich descriptive meaning to data elements. The combined metadata and relationships form a semantic layer that fully describes the meaning of the data and allows for visualization of all the data in its granularity. Knowledge graphs allow a user to identify duplicate or inconsistent data by visualization, as this data will have an interconnected relationship with other entities. These overlaps can alert the user to identify inconsistencies and make the necessary changes to ensure data quality. Knowledge graphs can also easily represent data ownership by mapping relationships across the different business domains and going back to the origin of the data. Last but not the least, patterns revealed by relationships may aid a company in developing analytics for a better understanding of the data.

*Data integration* is the process of combining data from various sources and presenting a consolidated view of the data to the user. A large fraction of the data in enterprises resides in relational databases [57]. One approach to data integration relies on a global schema that captures the interrelationships between the data items represented across these databases. Creating a global schema is an extremely difficult process because there are many tables and attributes; the experts who created these databases are usually not available; and because of a lack of documentation, it is difficult to understand the meaning of the data. Due to the difficulties inherent in developing a global schema, it is more convenient to bypass this issue by converting the relational data to a database with a generic triples schema, i.e., a knowledge graph. As shown in the Figure 2.9, knowledge graphs can provide a platform to integrate disparate data into a semantic data lake, where the whole process is to map the set of columns from the legacy sources to the common data catalog of a knowledge layer. Hence, knowledge graph now becomes a common harmonized model by which we can link and integrate disparate data; harmonize it, and then be able to export it to a variety of target sources, for example, regulatory reporting as per the central banks' guidelines.

## 2.5.2 KGE to Advance Precision for Machine Learning

The performance of a machine learning system is usually dependent heavily on the features engineered over the datasets, as opposed to the virtues of the machine learning algorithms themselves [14, 44]. Feature engineering tended to be manual and ad-hoc, and in the general case, there was no good reason to assume why one feature would perform better than another. Researchers also realized that the ‘goodness’ of a feature set also had a lot to do with the dataset itself, i.e., a particular set of features could perform well on one dataset but not another, all else being the same [44]. As mentioned in Section 2.4, KGE turns the symbolic representation of the graphs into a numeric representation for consumption by a machine learning algorithm. It is a way to input domain knowledge expressed in a knowledge graph into a machine learning algorithm providing far better improvements in feature engineering.

Knowledge graphs can be used for a variety of downstream tasks by leveraging their embedded representation, such as link prediction, triple classification, entity recognition, clustering, and relation extraction [2, 58]. Figure 2.10 illustrates this concept. We will go into more detail about *link prediction* and *triple classification* tasks because of their relevance in our work. These are in-KG applications; they are carried out within the scope of knowledge graph scope [5].

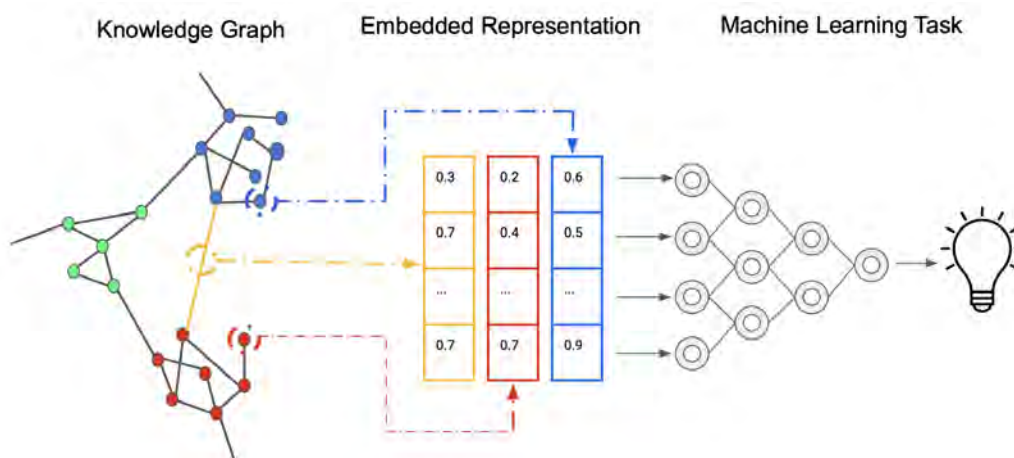


Figure 2.10: The embedded representation of a knowledge graph can be used for different machine learning applications [7]

**Link prediction** [5, 49] is the task of predicting whether a given entity has a specific relation with another entity. More formally, predicting  $h$  given  $(r, t)$  or conversely,  $t$  given  $(h, r)$ , with the former task denoted as head entity prediction  $(?, r, t)$ , and the latter as tail entity prediction  $(h, r, ?)$ . For instance,  $(Dhaka, isCapitalOf, ?)$  or  $(?, isCapitalOf, Bangladesh)$ . Link prediction exploits the existing facts in a knowledge graph to infer missing ones. The datasets for link prediction are constructed by sampling from the original knowledge graph. Then, the links can be removed to use in the validation set or the test set. With entity and relation representations learned during training, link prediction can be done using the ranking procedure. Take the prediction task  $(?, r, t)$  as an example, a ranking system can ‘predict’ the head entity by taking



every entity  $h'$  in the knowledge graph as a candidate answer and calculating a score for each  $(h', r, t)$ , using a scoring function. In descending order of scores, this yields a ranked list of candidate answers. If the embedding is 'good', the hope is that the correct prediction will be ranked nearer to the top of the list than incorrect predictions.

**Triple classification** [5, 59] is the problem of identifying whether an unseen triple fact  $(h, r, t)$  is true or not. It aims to give a yes-or-no answer to questions such as: *is Dhaka capital of Bangladesh?* This can be written in the form of a triple  $(Dhaka, isCapitalOf, Bangladesh)$ . A scoring function is used to calculate the score of a triple similar to the link prediction task. If the score is greater than a certain threshold, then it is considered a fact or true triple, otherwise it is a wrong triple.

## 2.6 Contextual AI

*Artificial intelligence (AI)* is powering more and more services and devices that we use daily, such as personal voice assistants, movie recommendation services, or driving assistance systems [60]. While AI has advanced, we all have moments when we wonder, "Why did I get this odd recommendation?" or "Why did the assistant do that?" One of the reasons for this mistrust is that most current AI systems operate as a "black box", with limited interaction capabilities, where the reasoning behind their decisions is indecipherable. In response to these limitations, a new phase of AI has emerged, named Contextual AI, to create a more collaborative partnership between humans and machines. Contextual AI does not refer to a specific algorithm or machine learning method - instead, it takes a human-centric view and approach to AI [60].

Predictions made by AI systems must be interpretable by the experts and explainable to the end-users, i.e., the system can show what it knows, how it knows, and what it's doing. In the absence of understanding how decisions were reached, users may reject recommendations outcomes that are counterintuitive. In systems where human safety is paramount, such as medical imaging or criminal facial recognition, explainability becomes a critical aspect of running a system that will not harm people. Explainability becomes a required component of AI, and being context-driven enhances explainability. Knowledge graph technology is the best way to maintain the context for explainability. It provides a human-friendly method for evaluating connected data, allowing the human to visualize the AI decision-making process. By better understanding the lineage of data (context of where it came from, cleansing methods used, and so forth), we can better evaluate and explain its influence on predictions made by the AI model [61].

# Chapter 3

## Related Work

Credit risk management is essential for financial institutions whose core business is lending. Thus, proper consumer or corporate credit assessment is critical, as financial institutions might incur considerable losses when borrowers default. Financial institutions must appropriately assess borrowers' credit risks to limit their losses from uncollectible accounts. As a result, they collect borrower data and develop numerous statistical and machine learning methods for objectively measuring and analyzing credit risk. Many studies [62–71] have been conducted on this subject due to its academic and practical significance.

Credit scoring and bankruptcy/default prediction are among the two most active research areas in the field of credit risk management [15]. The term “credit” refers to an amount of money that a financial institution lends to a consumer and which must be repaid in installments (usually at regular intervals) [62]. *Credit scoring* is a general term that relates to the risk assessment of individual borrowers (e.g., personal loans, home loans, car loans). On the other hand, *bankruptcy prediction* typically relates to the forecast of an organizational borrower's (e.g., a small business or corporation) insolvency. Generally, both credit scoring and bankruptcy prediction may be considered as binary classification tasks from a statistical modeling perspective [16]. However, the financial indicators utilized to perform these modeling tasks vary significantly.

In this chapter, we review the current research on credit risk prediction tasks and discuss how different techniques have evolved over time. Finally, we formulate the research queries to be addressed in our thesis based on the limitations found in the current literature.

### 3.1 Statistical Methods in Consumer Credit Scoring

The process of giving out credit leads to two options: granting a loan to a new customer or declining the application. Historically, the purpose of credit evaluation has been to compare a customer's features or characteristics to those of previous customers who have already repaid their loans [63]. If a customer's attributes are sufficiently similar to those of customers who

have been granted credit and then defaulted, the application for the loan will be denied. If the customer's characteristics match those of borrowers who have not defaulted, the application will normally be granted. Credit approval decisions were undertaken with the judgmental process whereby a subjective evaluation was carried out by the decision-maker, i.e., the credit analyst [63]. The success of a judgmental process relies on the knowledge and experience of the credit analyst. Over the years, statistical methods have gradually replaced the judgmental process with credit scoring models. The classification of good and bad credit is critical and is, in fact, the purpose of a credit score model [64]. The need for an appropriate classification technique is thus evident to determine the categorization of a new applicant. Table 3.1 lists the characteristics of loan applicants that are commonly employed in developing consumer credit scoring models [64–68].

Table 3.1: Loan applicant's characteristics used in building typical credit scoring

Characteristics	Attributes
Age	18-25, 26-40, 41-55, 55+ years
Annual income	0-10,000, 11,000-20,000, 21,000+ USD
Time at present address	0-1, 1-2, 3-4, 5+ years
Home status	Owner, tenant, other
Marital status	Married, divorced, single, widow, other
Purpose of loan	Coded (as per the organization's specifications)
Loan amount	$X$ USD
Loan duration	0-12, 13-36, 36+ months
Has credit card	Yes, no
Type of occupation	Coded (as per the organization's specifications)
Time with employer	$X$ years
Time with bank	$X$ years
Type of bank accounts	Current and/or savings, none
Has guarantees	Yes, no

Statistical models, called scorecards or classifiers, use predictor variables from application forms and other sources to produce estimates of the probabilities of default (POD). An accept or reject decision is made by comparing the estimated POD to a certain threshold. Discriminant analysis [63, 72, 73], linear regression [73], logistic regression [69], and decision trees [63, 69] are some of the most common statistical methods used in the financial industry to make credit scorecards, but there are many more.

*Discriminant analysis* is an established statistical technique to classify customers as good credit or bad credit and has long been applied in credit scoring applications [63, 73]. A well-known application is corporate bankruptcy prediction, the first operational scoring model based on five financial ratios, taken from eight variables from corporate financial statements [72].

*Logistic regression* is also one of the most widely used statistical techniques in the credit scoring field due to its simplicity and transparency in predictions [69]. What differentiates a logistic regression model from a linear regression model is that the outcome variable in logistic regression is dichotomous (a 0/1 outcome) [63]. Logistic regression assumes a linear relationship between the inputs and the log-likelihoods. However, the assumption of linearity does not always hold, as there are instances when the relationship between the independent variables and the log odds is non-linear [69]. In recent years, various machine learning methods have superseded traditional statistical techniques to improve the accuracy of prediction tasks. The following section discusses the most popular machine learning methods applied to credit risk predictions.

## 3.2 Machine Learning Methods for Managing Credit Risk

Banks and financial institutions usually conduct credit evaluations for individuals and small business owners using conventional statistical techniques such as linear discriminant analysis, logistic regression, and decision trees. Generally, statistical learning methods assume formal relationships between variables in the form of mathematical equations, while machine learning methods can learn from data without requiring any rules-based programming. Machine learning methods are particularly powerful in capturing non-linear relationships [69]. With the advancement of machine learning and artificial intelligence techniques, financial risk management researchers have increasingly used machine learning techniques for managing credit risk in recent years [74]. Based on the learning methods, machine learning classifiers used for credit risk prediction can be categorized into two families: *supervised* and *unsupervised*.

### 3.2.1 Supervised Learning Methods

Supervised learning algorithms use labeled datasets for training. The trained model can be used to make predictions for unlabeled samples. The effectiveness of using single classifiers to predict credit risks (bankruptcy or credit scoring) has been demonstrated in numerous studies. Support Vector Machines (SVM) [75–77] and Neural Networks [78] are the two most commonly used single classifiers for this task. Hybrid SVM models have been proposed to improve the performance by adding methods for the reduction of the feature subset. However, these only classify and don't provide an estimation of the probability of default [79]. Neural Networks have strong nonlinear fitting abilities; they can map any complex nonlinear relationship, and their learning rules are simple and easy to apply on a computer. However, the primary limitation of neural networks is their inability to learn more dense numerical features [80].

Many studies have shown that a single classifier does not solve credit assessment problems well [81]. Because various data sets often have diverse structures and features, credit assessment data usually has a class imbalance problem, i.e., the number of defaulters is much lower than

the number of good credit users [74]. At the same time, there are often a large number of sparse classification features in the credit data set, such as information on occupation and region. Therefore, using the integrated approach through different basic classification models, features can be effectively extracted from unbalanced and sparse data, resulting in better classification results. According to the research in [70, 71], the ensemble method (bagging, boosting, and stacking) performs better than single machine learning and statistical techniques.

The Gradient Boosting Decision Tree (GBDT) [82] method is an ensemble technique that has gained popularity due to its stability and performance in numerous data mining and machine learning competitions. GBDT has performed well in various machine learning tasks, including multi-class classification, click prediction, and learning to rank [80]. The difference with the Random Forests (RF) algorithm is that RF builds each tree independently and combines results at the end of the process (by averaging or “majority rules”), while GBDT builds one tree at a time and combines results along the way. GBDT reduces variance because multiple models are used (bagging), and it reduces bias by telling the next model what errors the prior models made (boosting) [83]. It has two clear advantages: it can handle dense numerical features well, and it requires less tuning time but has higher prediction accuracy. Because the GBDT learning tree is not differentiable, it is difficult to update the GBDT model in online mode, requiring repeated retraining from scratch. This flaw also prohibits GBDT from learning huge data sets since it is challenging to load them into memory [84]. Extreme Gradient Boosting (XGBoost) [85] is an advanced implementation of the gradient boosting algorithm, offering increased efficiency, accuracy, and scalability over simple bagging algorithms. It supports fitting various kinds of objective functions, including regression, classification, and ranking. XGBoost offers increased flexibility since optimization is performed on an extended set of hyper-parameters while it fully supports online training [83].

### 3.2.2 Unsupervised Learning Methods

Unsupervised learning refers to the task of detecting patterns from unlabeled data. In this setting, no labeled data is available. Clustering methods can also be used to determine the risk of bankruptcy or credit default. These methods can help to identify groups of loan applicants or enterprises with similar characteristics [10]. A cluster-based dynamic scoring model with the K-means algorithm can help the lender to identify the individual’s credibility at an earlier stage of the loan period without losing its accuracy [86]. Since the evaluation of clustering algorithms typically involves multiple criteria, it can be modeled as a multiple-criteria decision-making (MCDM) problem to rank a selection of popular clustering algorithms in the domain of financial risk analysis [87]. The comparative survey of different clustering methods concludes that no algorithm can achieve the best performance across all measurements for any data set [87].

### 3.3 Deep Learning Techniques for Default Prediction

Deep learning algorithms have been successfully applied in literature since the 1980s in an attempt to improve classification accuracy [69]. Additionally, deep learning models with optimal hidden layers have been designed to discover information that is difficult to identify using conventional statistical and machine learning methods. There is an emerging trend to substitute statistical and classical machine learning methods with deep learning techniques in credit risk management [69].

A mortgage default prediction model using Convolutional Neural Networks (CNN) has been proposed in [17], utilizing time-series data from client transactions in current accounts, savings accounts, and credit cards. In their study, the CNN model outperformed the Random Forest classifier. Deep belief networks showed better performance compared to well-known credit scoring models such as logistic regression, multi-layer perceptron, and support vector machine using Corporate Default Swaps (CDS) data [88].

A deep learning framework DeepGBM [80] has been proposed for credit assessment that consists of two parts: CatNN and GBDT2NN, which are used to deal with sparse categorical features and dense numerical features, respectively.

The Deep CNN model was compared with the multi-layer perceptron using German and Australian credit datasets, and the results showed a superior overall accuracy rate for CNN [89]. These studies have shown the superiority of deep learning models in credit scoring. However, the performance of deep learning models is dependent on the choice of activation function, the number of hidden layers, and the dropout rate. The results in [90] showed a better performance for ensemble methods, such as boosting and bagging, when compared with deep neural networks using the Taiwan credit dataset.

In [91], a deep learning model integrated with knowledge graph technology has been proposed to forecast bond defaults. They constructed a knowledge graph with the publicly available bond dataset of China. In their work, they optimized Deep Factorization Machines (DeepFM) to learn higher-order features automatically, and the bond knowledge graph was used as prior knowledge to expand higher-order cross-features. Their proposed deep learning model outperforms the traditional machine learning models in terms of prediction accuracy.

### 3.4 Research Queries

Based on the discussions in the above sections, we can easily conclude that the researchers' contributions to the credit risk management field are substantial due to the practical significance of the problem. However, most of the models in the current researches only concentrate on prediction tasks, but relatively a few try to establish the causal relationship between attributes and model prediction. Credit approval is a critical decision for banks and financial institutions,

and both model creators and regulators want a causal explanation of the prediction model. Based on the limitations observed in the current literature, we can formulate the following research queries to be addressed in our thesis:

**Explainable Prediction Model:** The problem with deep learning and machine learning models is interpretability, as they are not transparent in nature. Since banks and financial institutions are governed by regulators, they are required to be transparent in their credit decision process. Furthermore, a bank should be able to tell the borrower why their loan application has been rejected. Hence, the need to make the model transparent in the credit risk prediction task is of paramount importance. As explained in Section 2.6, knowledge graph technology is the best way to make the prediction model interpretable. It provides a human-friendly method for evaluating related data, allowing humans to visualize the decision-making process taken by the system. The influence of knowledge graphs on financial services is only at the beginning of the process, where the knowledge scientists can play a vital role in building bridges between business needs, queries, and data.

**Improve Model's Precision with Knowledge Graph Embedding:** The performance of most machine learning algorithms is heavily dependent on the features engineered over the datasets. As discussed in Section 2.5.2, knowledge graph embedding is a way to input domain knowledge expressed in a knowledge graph into a machine learning algorithm, providing far better improvements in feature engineering. Since knowledge graph embedding preserves the semantic information and structural content, this can surely boost the performance of conventional machine learning models in loan default prediction tasks.

# Chapter 4

## Methodology

In this chapter, we present an overview of our approach and the relevant technical details. Figure 4.1 shows the technology architecture of the proposed model for loan default prediction. The model is generic in nature and can be applied to all banks and financial institutions.

Our proposed methodology can be broken down into the following steps:

- Design and develop a *semantic data model* with an ontology. A semantic data model has the advantage of connecting and integrating disparate data sources, harmonizing them, and then exporting them to a variety of target sources.
- Construct the *knowledge graph* using the credit dataset (publicly available) mapped with our semantic data model to obtain entities, entity features, and relationships among the entities. Nodes represent entities in the graph, and the edges connecting nodes indicate their relationships. After knowledge extraction and validation, we stored the knowledge graph in the graph database.
- Machines cannot directly access the knowledge graph represented by symbols to perform computations. So, we adopted appropriate *knowledge graph embedding (KGE)* techniques to represent knowledge in a low-dimensional vector space that embeds entities and relationships from the knowledge graph. The vectors preserve the original graph's semantic and structural content.
- Formulate the loan default risk prediction as a *binary classification problem* within the KGE space by computing similarities (link prediction/triple classification) between loan applicants.
- Input the vectors extracted from the graph embedding as features to the *machine learning classifier* to forecast loan default.

The specific steps of the proposed methodology are elaborated below:



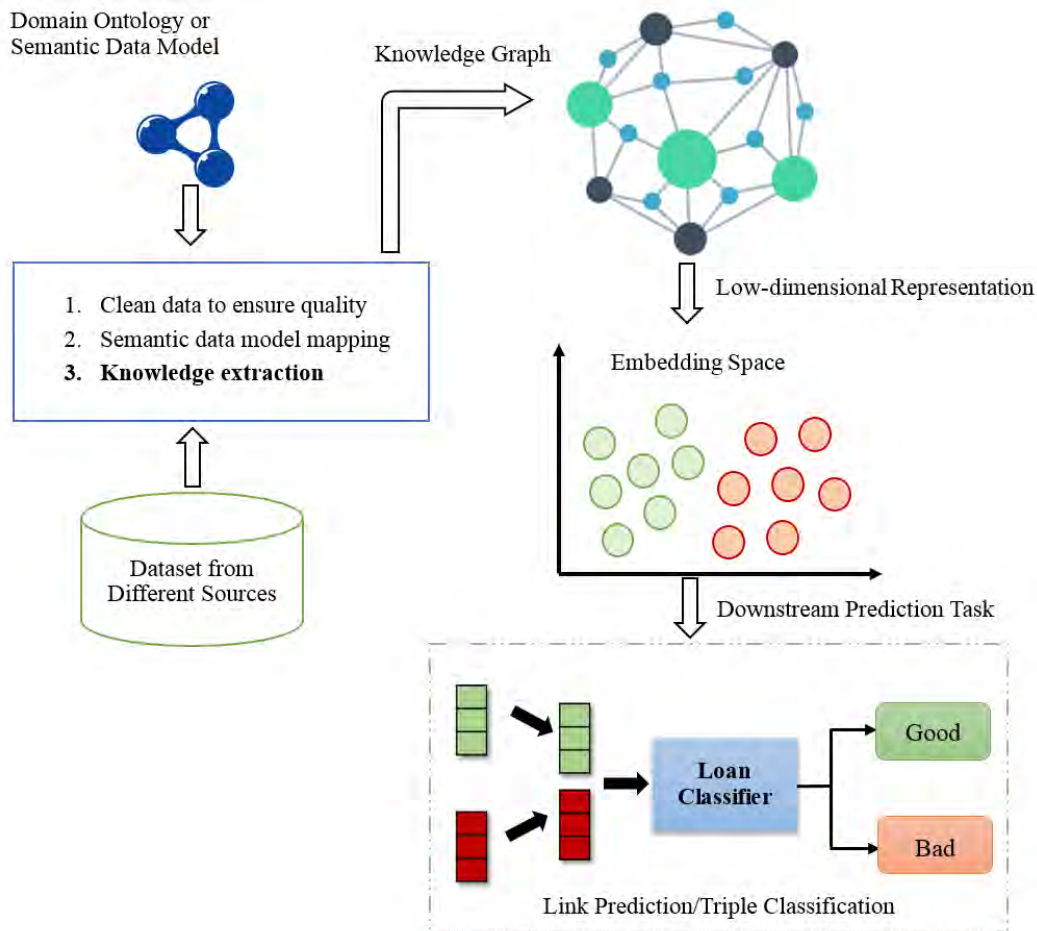


Figure 4.1: Technology architecture for loan default prediction model

## 4.1 Knowledge Extraction

Financial institutions create massive amounts of data via digital financial services offered to their customers (from client data to campaigns, social media, and emails) [92]. However, aggregating data remains isolated in a department or database and is not accessible to the entire organization for decision-making. They also consume data produced by third-party providers; many data providers collect their data through the processing of unstructured sources and devote significant effort to offering it in a structured form for usage by others. External data must be linked to the company's internal data to be used effectively. Such data integration enables many popular use cases, such as 360-degree views of a customer, fraud detection, risk assessment, loan approval, etc. These companies may use knowledge graphs in a semantic manner to gain new insights and commercial opportunities from data to assist them in breaking down silos by making data visible, meaningful, and even real-time.

Knowledge extraction is the process of obtaining knowledge from structured (relational databases, XML, CSV) and unstructured (text, documents, images) sources [93]. The resulting knowledge needs to be in a machine-readable and machine-interpretable format and must represent

knowledge in a way that unambiguously defines its meaning and promotes inference. This thesis mainly uses the structured data sources published by Home Credit Group [8] as a benchmark dataset where the customers' identities are anonymized. This dataset contains demographic and historical financial attributes (including prior loans, monthly loan installment, past payments) of loan applicants. The data preprocessing techniques are discussed in Section 5.1. The description and key attributes of the dataset from different sources are shown in Table 4.1. A schema definition or semantic data model is required to combine data from different sources into a knowledge graph. The following section discusses the design and development of a semantic data model for knowledge graph construction.

Table 4.1: Dataset description and key attributes

Data Source	Description	Key attributes
application.csv	Information about the loan applicant (anonymized) and loan at application time	Demographic information (i.e., age, gender, family status, etc.), employment type, years in business/employment, income, loan information (loan type, requested amount), external source's data, the target variable 1 for the client with payment difficulties and 0 for all other cases.
bureau.csv	Borrower's previous credits provided by other banks and financial institutions	Number of loans active/close, total loan exposure, total overdue amount, remaining term, number of defaults.
bureau_balance.csv	Borrower's monthly data of prior credits in the bureau.	Monthly status of availed credits , i.e., regular/overdue payments
previous_application.csv	Information on previous loan applications and their status for the applicants	Loan amount, loan type, loan duration, decision (approve/reject)
POS_CASH_balance.csv	Monthly information on current customers' prior points of sale or cash loans	Monthly balance, term of cash loan, loan status
credit_card_balance.csv	Monthly balance snapshots of previous credit cards	Credit limit, utilized amount, receivable amount, payments
installments_payments.csv	Payment history for previous loans	Installment size, last paid amount, overdue amount, status

## 4.2 Semantic Data Modeling

Semantic data modeling is a method of structuring data to represent it in a specific logical way. An ontology acts as a semantic data model that contains a formal, explicit definition of the concepts in our domain of interest and properties that describe each concept's different attributes or characteristics [94]. With a semantic data model powered by ontology, we understand a model of knowledge in a particular area (in our case, it is credit risk) that promotes the integration of heterogeneous resources at the conceptual level, providing a unified approach to the description of their semantics. The process of developing ontologies relating to the financial world was described in [95]. The Financial Industry Business Ontology (FIBO) [96] is an example of a conceptual business model for the financial industry. It shares a common vocabulary and meaning for the financial industry and regulators. We designed and developed a semantic data model considering the FIBO's principles and standards, as illustrated in Figure 4.2. A semantic data model has the advantage of representing data in a single interchangeable format such as RDF, so that both machines and humans can understand it.

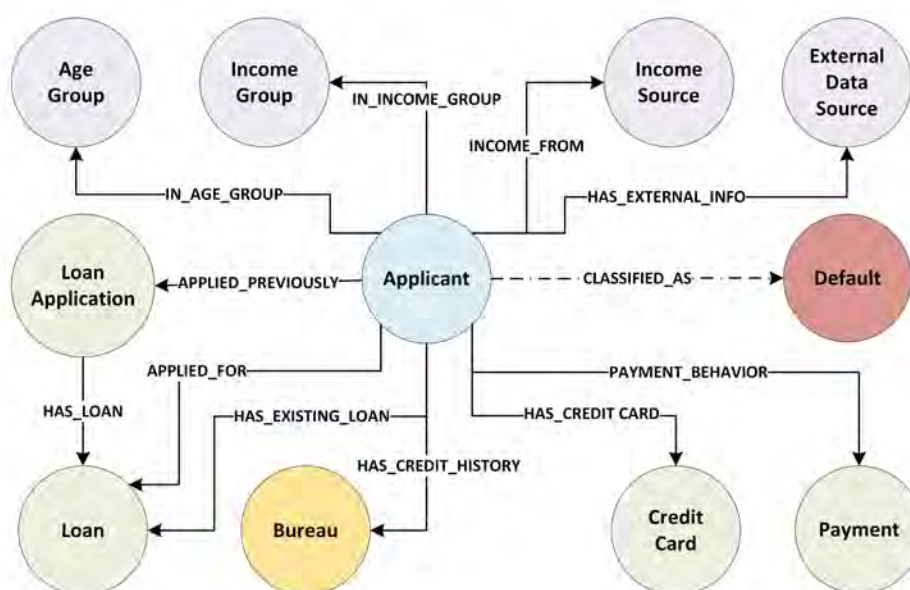


Figure 4.2: Semantic data model

As shown in Figure 4.2, twelve relationships exist in our semantic data model for a particular loan applicant. From the study of the literature (discussed in Chapter 3) in the credit risk management domain, we defined these relationships. This semantic data model provides the terminology definition for the knowledge graph. Our objective is to predict whether a loan applicant has the “Classified\_As” relationship with the “Default” concept. The semantic data model can be evolved at the pace of business demands so that financial firms can include additional business requirements, data sources, and other models. It also allows easy maintenance of data consistency when data is updated.

## 4.3 Knowledge Graph Construction

A knowledge graph is a conceptual model of the world that is also intuitive to understand. Knowledge can be stated using factual triples such as *(subject, predicate, object)* or *(head, relation, tail)*, for example, *(Credit Risk, Type Of, Financial Risk)*.  $G = E, R, F$  is a formal definition of a knowledge graph, where  $E$ ,  $R$ , and  $F$  sets of entities, relations, and facts. A fact is indicated by the triple  $(h, r, t) \in F$  [2], where  $h$ ,  $r$ , and  $t$  represent the head, relation, and tail, respectively.

There are two main ways to construct a knowledge graph: top-down and bottom-up (see Section 2.3.1). Since the ultimate goal of building a knowledge graph from a credit dataset is to provide knowledge for loan default prediction, we create the knowledge graph using structured data published by Home Credit as the data source and employ the top-down technique for graph construction. Nodes represent entities (such as Applicants, Loans, Bureaus, etc.) in the knowledge graph, and the edges connecting nodes indicate their relationships (e.g., Applied\_For, Classified\_As). We map our semantic data model to obtain entities, entity features, and connections within entities. After knowledge extraction and validation, we complete the knowledge graph construction, and the end outcome is a directed graph. Figure 4.3 depicts a simplified illustration of the knowledge graph construction process from disparate data sources.

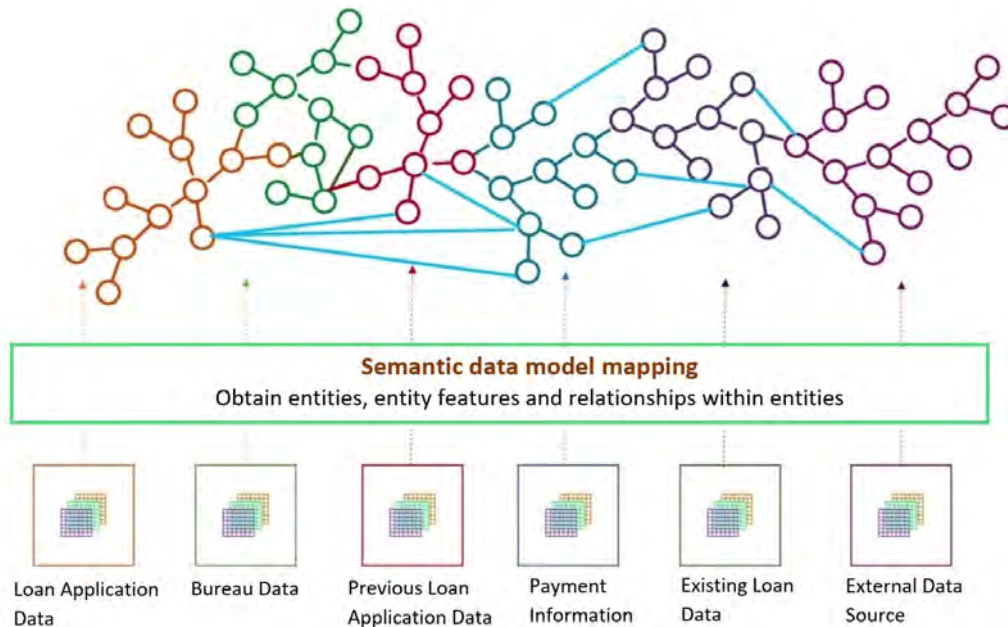


Figure 4.3: Simple illustrations of the knowledge graph construction process from disparate data sources

Knowledge graphs are usually stored in a graph or NoSQL database. A graph database uses highly interlinked data structures built from nodes, relationships, and properties. In turn, these graph structures support sophisticated, semantically rich queries at scale. Neo4j [97] is a popular

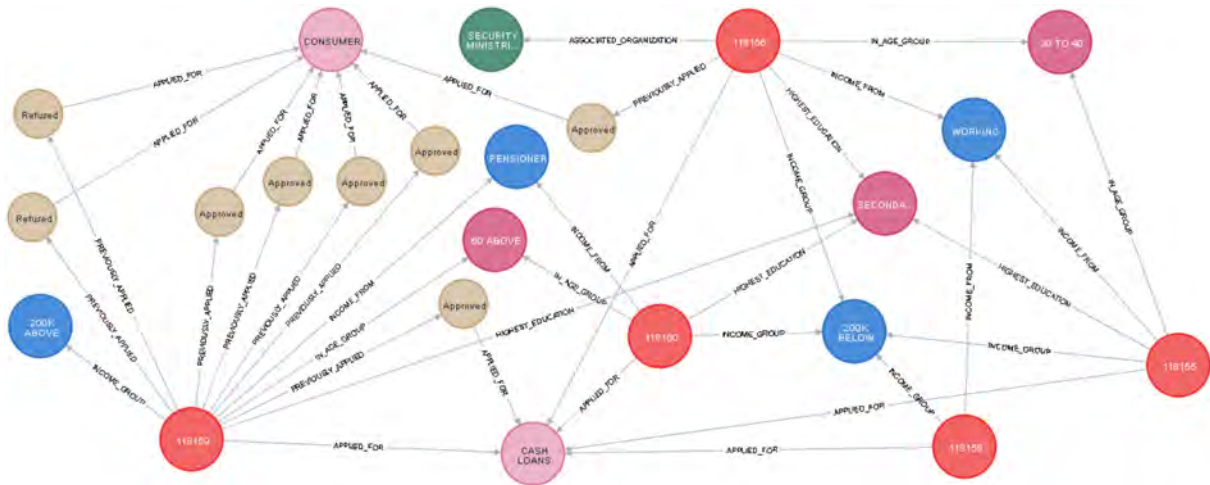


Figure 4.4: Part of a knowledge graph for loan default prediction

graph database with an open-source community version that includes native graph storage and processing features. We use the Neo4j graph database to store the built knowledge graph, containing 16,33,661 nodes and 12 relationships. Figure 4.4 shows a portion of the knowledge graph.

## 4.4 Knowledge Graph Embedding Model Selection

It is difficult for machines to directly access the knowledge graph represented by symbols to perform computational operations. Knowledge graph embeddings (KGEs) are now a widely adopted technique for representing knowledge that embeds entities and connections in low-dimensional vector spaces. They can be a beneficial source of features for a subsequent machine learning classification task. The vectors preserve the original graph’s semantic information and structure. There are many knowledge representation models such as TransE [18], TransH [3], TransR [19], which are translation based methods. The KGE models define various score functions, and they are used to quantify the distance between two entities with respect to their connections in the low-dimensional embedding space. These scoring functions are employed during the training of KGE models to determine the entities that are closest to each other. The unconnected entities, on the other hand, are a greater distance apart (see Section 2.4).

An entity may have numerous semantic characteristics related to various relationships in our processed knowledge graph. For instance, multiple applicants may apply for the same type of loan; on the other hand, occupations and income sources can also be similar for the different loan applicants. TransE learns only one aspect of similarity. It has difficulty handling relationships that are not one-to-one. TransH maps the entities with their associated relation hyperplanes. TransR employs a relationship-specific area to deal with various connections. Although both TransH and TransR overcome the limitations of TransE, they still cannot handle multiple types

of relations determined by each relation's head and tail entities [98]. For example, consider two triples of  $(Branch, Part\_of, Bank)$  and  $(Revenue, Part\_of, Income\ Statement)$ . Both triples have a relation *Part\_of* in common, but the relationship in each triple should be perceived differently.

Our research uses the ComplEx [20] model, a bilinear diagonal model, which can handle various binary relations; symmetric and antisymmetric relations exist in the knowledge graph. Furthermore, this model is scalable to enormous datasets since it retains linearity in both space and time while delivering state-of-the-art prediction capabilities to the user. We extract the triples from the knowledge graph and train the ComplEx embedding model on those triples, i.e., translating the loan applicants' attributes and financial states into a vector space. Each loan applicant has a unique signature in the vector space. We utilize this vector representation of the knowledge graph to predict the loan default by computing the semantic similarity among the borrowers in the graph embedding space.

## 4.5 Loan Classification Using Knowledge Graph Embedding

We consider loan default prediction to be a binary classification task, and it can be solved by applying KGE using either link prediction [5] or triple classification [2]. The objective of *link prediction* is to forecast one entity's relationship with another entity, such as predicting  $h$  with  $(r, t)$  known or predicting  $t$  with  $(h, r)$  known. The first is denoted by  $(?, r, t)$ , whereas the second is indicated by  $(h, r, ?)$  [5]. For example,  $(Applicant\_A, Classified\_As, ?)$  means predicting whether *Applicant\_A* is classified as default or regular. In contrast, *triple classification* tries to determine if an undiscovered triple  $(h, r, t)$  is a true or false fact, e.g.,  $(Applicant\_A, Classified\_As, Default)$  is a true or false fact. These are essentially knowledge graph completion tasks, i.e., adding unseen or novel connections to the graph, and have been extensively studied in the prior literature [4, 18].

Both *link prediction* and *triple classification* are downstream machine learning tasks. The performance of most machine learning algorithms is highly dependent on feature engineering, which is a very time-consuming process. A knowledge graph is also a key enabler for machine learning applications. As illustrated in Figure 4.5, using a knowledge graph, it is possible to generate feature sets based on the knowledge layer by linking and harmonizing data from various sources. Because we understand the higher-level concepts, we can tell the knowledge graph to pull together all of the disparate data sources to make the feature set for machine learning out of them. Then, we carry out data validation and consistency checking to ensure the quality of the data from these disparate sources is flawless. Next is the data classification and inference process. The knowledge graph can draw inferences about certain relationships that exist in the data to get a much more robust and meaningful output, allowing us to improve the performance of machine learning. Thus, we can see that knowledge graphs can further ingest machine learning output.

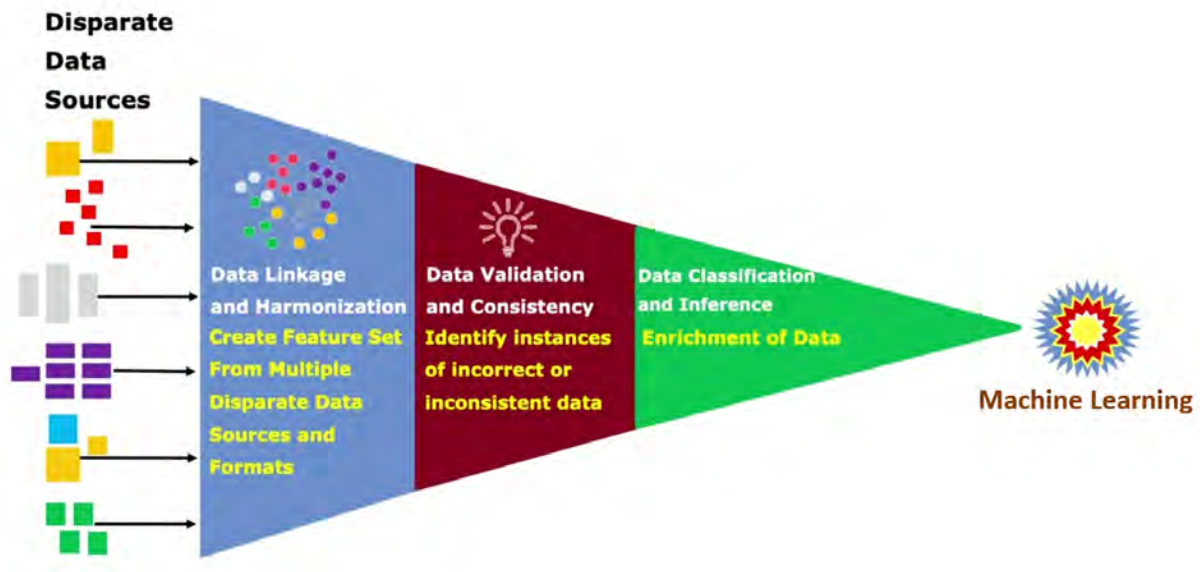


Figure 4.5: A knowledge graph enables better feature engineering for machine learning

We train the ComplEx embedding model on the triples or facts extracted from the knowledge graph and get the vector representations. We apply these vectors as features to the conventional machine learning classifiers for the risk prediction of loan default. Based on the literature review in Chapter 3, we employ four popular machine learning classification models that exhibit strong performance in credit default prediction tasks: Logistic Regression (LR) [69], Random Forest (RF) [82], Light Gradient Boosting Machine (LightGBM) [84], and Extreme Gradient Boosting algorithm (XGBoost) [85].

## 4.6 Explanations for Loan Default Prediction

As discussed in Section 2.6, machine learning models have difficulty with interpretability because they are not transparent in nature. Since banks and financial institutions are highly regulated entities, they are obligated to be transparent in their credit decision process. Furthermore, a financial institution should be able to tell the applicants why their loan application has been refused. Therefore, it is critical to make the credit risk prediction model understandable to both the model creators and regulators. Knowledge graph technology is the best way to make the prediction model explainable. It provides a human-friendly method for evaluating related data, allowing humans to visualize the decision-making process taken by the system.

We can query the knowledge graph to generate explanations for the predicted triples. Explanations are valuable when KGEs are implemented in real applications, as they help improve the reliability and people’s trust in predicted results. Table 4.2 shows the explanations with the supports (related attributes) for a predicted triple (*Applicant\_A*, *Classified\_As*, *Default*) generated by our model.

Table 4.2: Explanations and supports for prediction made by the model

<b>Predicted Triple: (Applicant_A, Classified_As, Default)</b>				
<b>Explantion</b>			<b>Support</b>	
<i>Head</i>	<i>Relation</i>	<i>Tail</i>	<i>Attribute Name</i>	<i>Value</i>
Applicant_A	In_Age_Group	30_to_40	Age (in years)	37
Applicant_A	In_Income_Group	300K_to_500K	Monthly income	40000
Applicant_A	Income_From	Working	Associated organization	Commercial
Applicant_A	Applied_For	Cash_Loan	Loan amount	300000
			Loan term (in months)	24
Applicant_A	Applied_Previously	Revolving_Loan	Loan amount	300000
			Decision	Refused
Applicant_A	Has_Existing_Loan	Cash_Loan	Loan amount	50000
			Loan term	36
			Remaining term	16
			Overdue Amount	40000
Applicant_A	Has_Credit_History	Bureau	Number of loans active	2
			Total loan exposure	700000
			Total overdue amount	90000
			Number of defaults	1
Applicant_A	Has_Credit_Card	Credit_Card	Credit limit	100000
			Utilized amount	60000
			Overdue amount	10000
Applicant_A	Payment_Behavior	Irregular	Monthly installment amount	20000
			Number of unpaid installment	4



# Chapter 5

## Experimental Result and Analysis

In this chapter, we present our experimental work in detail and discuss the outcomes of our experiments with a comparative analysis. First, we discuss experimental data, preprocessing techniques, experimental settings, and training of knowledge graph embedding for loan default prediction model. Then, we provide a comparative analysis of our experimental results for various machine learning models with and without knowledge graph embedding features.

### 5.1 Experimental Data

As mentioned earlier, we choose the Home Credit Default Risk dataset [8] for the experiment. There are seven different data sources in the dataset that contain information about loan applications (with anonymized identities of the applicants), prior credits with other institutions, previous applications, and the payment history of earlier loans. Each data source is described briefly below [8].

- *application*: main data source that contains information about each loan application at Home Credit. Every loan has its own row and is identified by the feature `SK_ID_CURR` which is a loan id. The application data comes with the `TARGET` column indicating 0: the loan was repaid or 1: the loan was not repaid.
- *bureau*: data concerning the client's previous credits from other financial institutions. Each previous credit has its own row in the bureau, but one loan in the application data can have multiple previous credits.
- *bureau\_balance*: monthly data about the previous credits in the bureau. Each row represents one month of a previous credit, and a single previous credit can have multiple rows, one for each month of the credit length.
- *previous\_application*: previous applications for loans at Home Credit of clients who have loans in the application data. Each current loan in the application data can have multiple

previous loans. Each previous application has one row and is identified by the feature `SK_ID_PREV`.

- *POS\_CASH\_BALANCE*: monthly data about the previous point of sale (POS) or cash loans clients have had with Home Credit. Each row is one month of a previous point of sale or cash loan, and a single previous loan can have many rows.
- *credit\_card\_balance*: monthly data about previous credit cards clients have had with Home Credit. Each row represents one month of a credit card balance, and a single credit card can have many rows.
- *installments\_payment*: payment history for previous loans at Home Credit. There is one row for every payment made on time and one row for every missed payment.

Figure 5.1 shows how all of the data sources are related.

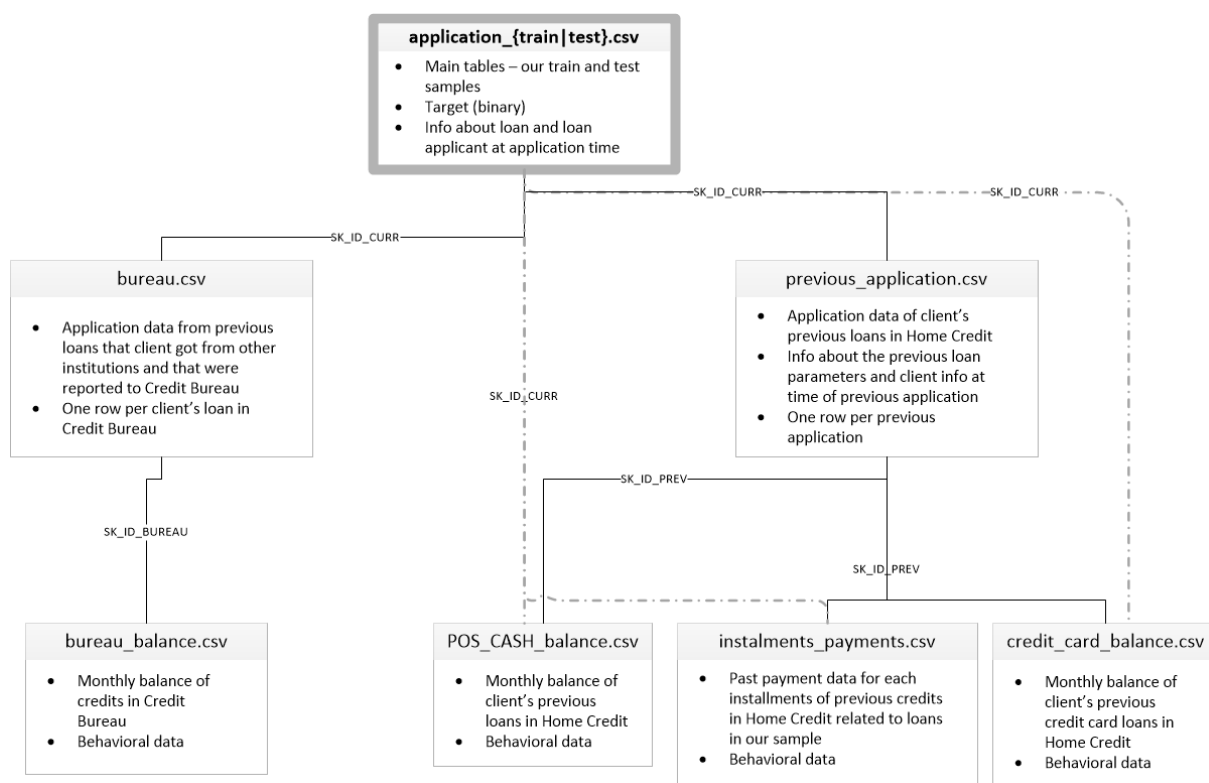


Figure 5.1: Relationship diagram of the Home Credit Dataset [8]

The dataset has 307,511 samples with 218 characteristics. The defaulted loan in the sample is denoted by 1, while the remaining loan applications are denoted by 0, indicating that they were repaid on time.

### 5.1.1 Exploratory Data Analysis

Exploratory data analysis (EDA) [99, 100] is a task performed by a data scientist to find trends, anomalies, patterns, or relationships within the data. This statistical approach relies on data visualization techniques, enabling us to determine how best to manipulate data sources to get the answers we need, making it easier to discover patterns, spot anomalies, test hypotheses, or check assumptions. It also takes advantage of several quantitative methods to describe the data.

All the datasets together have around 218 variables or features. Visualizing and analyzing every one of them in this thesis would make it difficult to read. Therefore, we will discuss a small set of features based on the importance of the variable related to the target, i.e., the loan default prediction task. We considered what it looks like to help understand the data and the business based on intuition and domain expertise.

**The distribution of loan repayment status:** The repayment status is the “target” variable in the application data set. According to Figure 5.2(a), a total of 282,686 loans were repaid on time, while 24,825 loans defaulted (unpaid). As shown in Figure 5.2(b), 8.07% of the customers have payment difficulties (encoded 1), which means that the client had a late payment for more than X (e.g., 90) days on at least one of the first installments of the loan. 91.9% is for customers with regular payments. We can see from these counts and plot that the dataset is imbalanced.

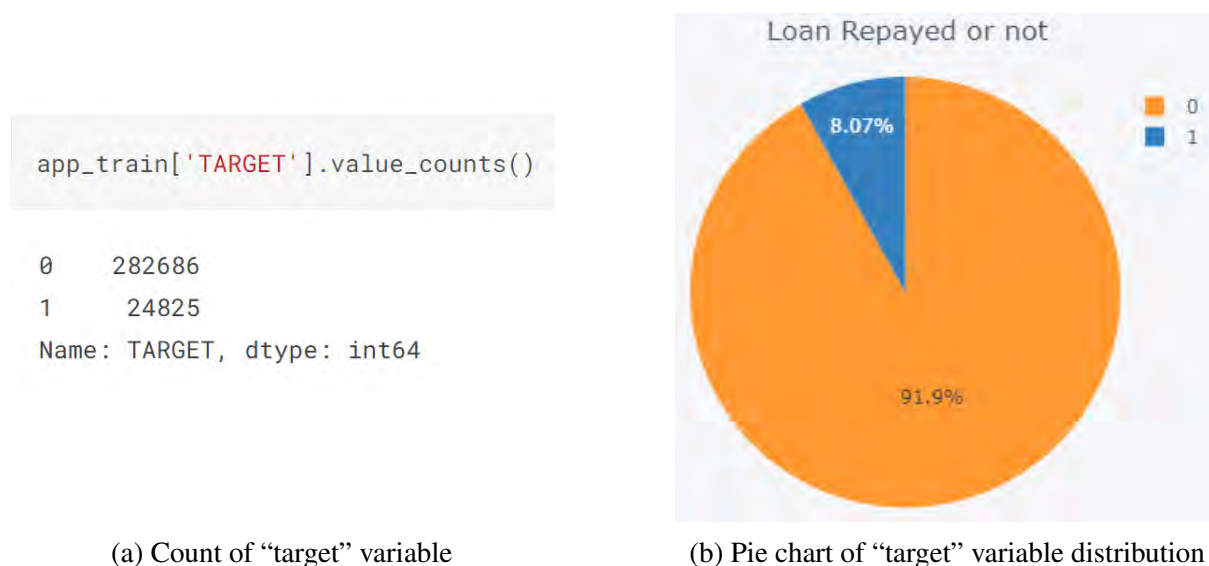


Figure 5.2: The distribution of loan repayment status

**Exploration of income sources:** The analysis of the distribution of the applicants’ income sources does not tell us precisely what kind of job the applicant is performing. It can give us an idea of the nature of the income. Figure 5.3 shows us that the largest segment of the dataset is made up of customers who mentioned that they were working when they applied for a loan, with a percentage of 51.6%. The second important section is made up of commercial associates at 23.3%, followed by pensioners at 18%. The fourth segment is devoted to state servants. And the

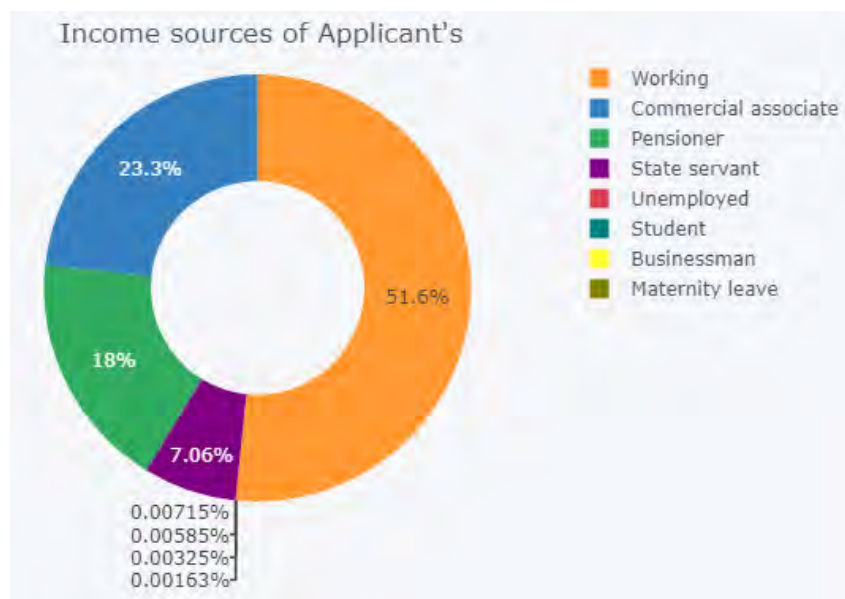


Figure 5.3: The distribution of income sources of applicants

last part has only 0.04%, and it consists of unemployed people, businesspeople, students, and people on maternity leave.

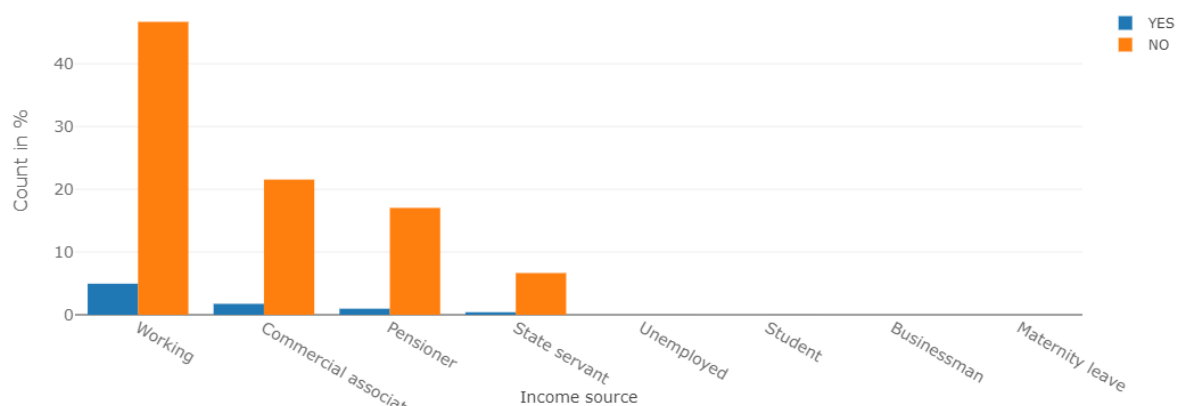


Figure 5.4: Applicants' income sources in terms of repaid or not (in percent)

As we can see clearly in Figure 5.4, more than 40% of applicants with the status of working have difficulty repaying the loan. But again, as we do not know exactly what the income source for this status is, it is difficult to explain the reason behind this high percentage. More than 20% of commercial associates have payment issues, followed by pensioners (more than 15%). However, it is low for state servants (around 5%). Perhaps the stability of their salaries explains this payment behavior. For the rest of the income sources, the situation is unknown.

**Impact of the applicant's age on repayment:** From the study of the literature (discussed in Chapter 3), we observed that an applicant's age is a vital characteristic for developing the

consumer credit scoring model. To better understand the distribution of applicants' ages in the context of loan repayment, we first categorized the applicant's age into groups of spans of 5 years each. Then, for each group, we calculated the average value of the target, which tells us the ratio of loans that were not repaid in each age category. As illustrated in Figure 5.5, there is a clear trend: as the client gets older, there is a negative linear relationship with the target, meaning that as clients get older, they tend to repay their loans on time more often. The rate of failure to repay is above 10% for the youngest three age groups and below 5% for the oldest age group.

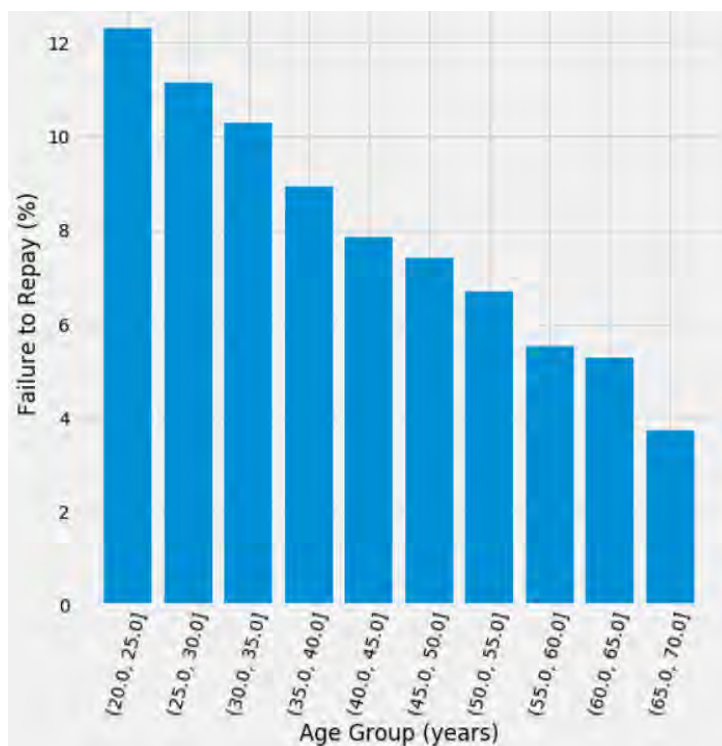


Figure 5.5: Default in repayment by age group

**Exploration of previous application data:** Old loans could influence the current loans or even explain them better. As usual, the present is constructed on the foundation of past experiences. One of the exciting features in the dataset is the status of the previous applications. As we can see in Figure 5.6, there is one of four situations for each previous application:

- *Approved* means that the loan application has been accepted, with a percentage of 62.1% in the data set.
- *Canceled* means that Home Credit or the client canceled the loan, which has a percentage of 18.9% in the data set.
- *Refused* means that the loan application has been denied to provide the credit. This status has a percentage of 17.4%.
- *An unused offer* signifies that the loan has been granted, but the applicant has yet to utilize it; it's either pending or suspended. This status has a percentage of only 1.58%.

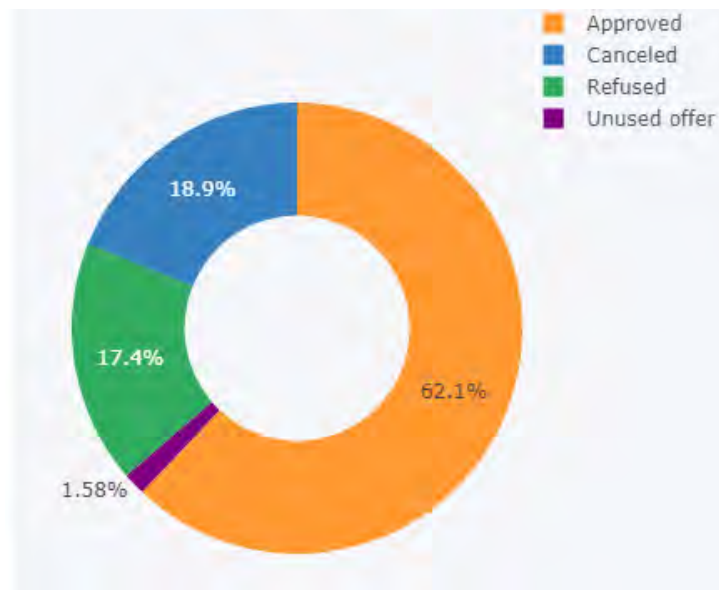


Figure 5.6: The contract status in previous applications

The reasons for rejecting previous loans have been encoded, which makes it difficult to understand. However, the distribution is understandable, and we can get a broad idea of the previous application. As Figure 5.7 shows, more than 1.3 million loans have XAP symbols that mean "not applicable," which may mean that this label may contain accepted loans. HC (symbol) was the rejection reason for around two hundred thousand loans. Other reasons for rejecting previous applications remain low in general.



Figure 5.7: Top reasons for previous applications' rejection

### 5.1.2 Data Preprocessing

Now that we have got some more understanding of the data, it is time to go further and do the data preprocessing to ensure the quality of the data. We performed various preprocessing tasks to control the distribution of each attribute in the dataset to improve the accuracy of the prediction model.

**Data cleansing** is the process of detecting and correcting (or removing) corrupt or inaccurate records from a table or database [99]. After cleaning, a data set should be consistent with other similar data in the data set. The inconsistency of the data may have been caused initially by user entry errors or corruption in transmission or storage. We cleaned the data to avoid the influence of data format, missing data, and value range on subsequent experiments. In our experimental data, we had to deal with two problems: *infinity values* and *missing values*.

- Since not all algorithms are capable of dealing with *infinity values* in the same way, it is critical to keep this in mind when developing the model. The first step in our data cleansing method is to deal with infinity values. The process of correcting infinity values is pretty straightforward; we simply convert infinity values to missing ones (NaNs) to rectify them in the following phase.
- Dropping features with many *missing values* may impact the accuracy of the model. Removing samples with missing values could do the same, as the data set will be too small. Thus, we fill in these missing values (known as imputation). For numeric variables, we impute the missing values with the median or average values in each attribute, and for categorical variables, we impute them with the most frequent category [91, 100].

**Normalization** is the process of transforming numeric data to the same scale as other numeric data [99]. When a dataset contains a wide range of values, it is essential to normalize the data to train models faster and avoid saturation. We used Min-Max-Scaler [101] to normalize our data. In this type of normalization, for each value in a feature, the Min-Max scaler subtracts the minimum value from the actual value and then divides it by the range. The range is the difference between the original maximum and the original minimum. This technique preserves the shape of the actual distribution and is calculated using the formula below:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (5.1)$$

where  $X_{norm}$  represents the normalized result of the data,  $X$  represents the value before normalization,  $X_{min}$  and  $X_{max}$  represent the minimum and maximum values of samples in a given feature, respectively.

**Encoding categorical variables** [102]: There are a few categorical features (e.g., gender, occupation, type of loan, education, etc.) in our dataset. We utilized the encoding technique to handle these categorical features. We chose *label encoding* for variables with only two categories and *one-hot encoding* for all other cases, i.e., columns with more than two categories. For label encoding, we assigned each category a unique integer based on the alphabetic ordering. On the other hand, each category's distinct value was added as a feature/column for one-hot encoding, resulting in a 1 in its category column and a 0 in the remaining new columns. We grouped the applicant's age and income into different segments (e.g., age: 20-30 years, income: below

20,000) and added the following features as domain knowledge in the credit risk evaluation [103]:

- *Debt\_to\_Income\_Ratio*: Percentage of total monthly obligations (repayment of loans and monthly expenses) related to the applicant’s gross monthly income. Borrowers that have a low debt-to-income ratio are more likely to pay on time.

$$\text{Debt\_to\_Income\_Ratio} = \frac{\text{Total\_of\_Monthly\_Debt\_Payments}}{\text{Gross\_Monthly\_Income}} \quad (5.2)$$

- *Debt\_to\_Credit\_Ratio*: The credit utilization rate or debt to credit rate is the amount of revolving credit (credit card) divided by the overall credit limits (amount of credit available).

$$\text{Debt\_to\_Credit\_Ratio} = \frac{\text{Utilized\_Revolving\_Credit\_Amount}}{\text{Total\_of\_Revolving\_Credit\_Limits}} \quad (5.3)$$

## 5.2 Experimental Settings

Among the 307,511 loan records, there were 24,825 defaulted loans in the dataset, which is around 8% of the total. The loan default samples are unbalanced, which means that the number of repaid loans is more than the number of defaulted loans by a significant margin. As a result, the classification of this data is skewed toward the majority class, severely reducing the minor class’s prediction ability. While dividing the training set and test set, we performed 5-fold cross-validation to ensure accurate results. The oversampling approach duplicates a positive sample for each training set, resulting in a final training set with a positive sample to negative sample ratio of around 1:10 (positive to negative). We used Python environment for data preprocessing, machine learning, and graph embedding-related tasks. The Neo4j graph database [97] was used for storing the knowledge graph.

## 5.3 Training Knowledge Graph Embedding

We used the preprocessed data and mapped it with our semantic data model to get the entity, its properties or attributes, and its relationship with other entities. The end outcome of knowledge extraction and validation is a directed graph. Nodes symbolize the entities in the knowledge graph, and the edges linking nodes indicate the connections within the entities. Figure 5.8 shows two sample loan applicants in our knowledge graph exported from the Neo4j graph database.

We further divided our training dataset into two groups: *training* and *validation*. The training set was used for knowledge embedding training and the validation set was applied for its evaluation. The validation set differs from the conventional sampling method because our data points are two entities connected by some relationship. So, we ensured that all entities were represented by





Figure 5.8: Knowledge graph query results for two sample loan applicants

at least one triple in both the training and validation sets. We used AmpliGraph [104], a Python library, to train knowledge graph embedding with the ComplEx model [20] known to bring state-of-the-art predictive power. The selection of hyper-parameters was based on the best results obtained by applying the ComplEx model to particular benchmark datasets commonly used in the knowledge graph embedding community. Table 5.1 shows the values of hyper-parameters used for the training. The embeddings are used directly as features to various machine learning classifiers.

Table 5.1: Hyper-parameters used for the training of KGE

Parameter Name	Value
batches_count	50
epochs	300
k: <i>the dimensions of the graph embedding space.</i>	100
eta: <i>number of false triples required to generate for each true triple.</i>	20
optimizer	‘adam’ with learning rate 1e-4
loss	‘multiclass_nll’
regularizer	l3 regularization. lambda: 1e-5

## 5.4 Evaluation Metrics

In any predictive modeling task, the evaluation of the model is of utmost importance. Binary classification divides data into two distinct categories: positives (P) and negatives (N). This classification generates four sorts of results: two types of accurate classes, true positives (TP) and true negatives (TN), and two types of inaccurate categories, false positives (FP) and false negatives (FN). The confusion matrix of these four outcomes is shown in Table 5.2.

Table 5.2: Confusion matrix

	<i>Predicted Positive</i>	<i>Predicted Negative</i>
<i>Actual Positive</i>	True Positives (TP)	False Negatives (FN)
<i>Actual Negative</i>	False Positives (FP)	True Negatives (TN)

This confusion matrix serves as the basis for the evaluation of the binary classifier. In our case:

- *TP* denotes the number of loans in our samples predicted to be Bad (will default), and they are Bad actually.
- *TN* denotes the number of loans in our samples assumed to be Good and found Good.
- *FP* signifies the number of loans anticipated to be Bad but was Good.
- *FN* indicates the number of loans expected to be Good but actually Bad.

We used the accuracy, precision, recall, F1 score, Matthews correlation coefficient (MCC), and receiver operator characteristic (ROC) curve [105] as evaluation metrics to compare the different machine learning methods to the proposed method. The ROC curve for a binary classification problem represents the TP proportion as a function of the FP ratio.

**Accuracy** is defined as the ratio of the number of samples correctly classified by the classifier to the total number of samples for a given test data set. Typically, accuracy is used to assess the effectiveness of a model with the help of the confusion matrix. The accuracy of a model is calculated through:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5.4)$$

**Precision** is calculated by comparing the number of true positives to the total number of true positives and false positives. To put it another way, precision estimates how many of the instances the classifier predicted as positive were actually positive. The precision of a model is calculated through:

$$Precision = \frac{TP}{TP + FP} \quad (5.5)$$

**Recall**, also known as *True Positive Rate (TPR)*, compares the number of true positives to the number of true positives and false negatives. Thus, it is the fraction of all positive instances that the classifier correctly identifies as positive. Recall is computed through:

$$\text{Recall or TPR} = \frac{TP}{TP + FN} \quad (5.6)$$

**False Positive Rate (FPR)** is calculated as the ratio between the number of negative instances wrongly classified as positive (false positives) and the total number of actual negative instances in a given dataset. So, the formula to calculate FPR is:

$$FPR = \frac{FP}{FP + TN} \quad (5.7)$$

**F1-score**, also called a balanced *F-score* or *F-measure*, is defined as the harmonic mean of precision and recall. It is a measure of the overlapping between the actual and predicted classes. The value ranges from 0 to 1 and is calculated with the below formula.

$$F1\text{-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5.8)$$

**Matthews correlation coefficient (MCC)** [106] is used in machine learning to assess the quality of binary (two-class) classifications. It is generally recognized as a balanced measure that can be employed even when the distribution of classes is unequal. The MCC is, in essence, a correlation coefficient between the observed and predicted binary classifications; it returns a value between -1 and +1. A coefficient of +1 represents a perfect prediction, 0 means no better than a random prediction, and -1 indicates total disagreement between prediction and observation. The MCC can be calculated from the confusion matrix using the following formula.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (5.9)$$

Although accuracy and F1-score are popular in binary classification, they may produce misleading outcomes for unbalanced datasets due to the unknown probability distribution of positive and negative factors. The MCC is a more reliable metric that only delivers high scores if the prediction achieves good results in all four confusion matrices (TP, FN, TN, and FP) [107].

**Receiver operating characteristic (ROC)** [108] is a two-dimensional graphical plot that illustrates the performance of a binary classifier model. The curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at all classification thresholds. The ROC curve can intuitively represent the performance of a classifier. Figure 5.9 shows an example of the ROC curve.

**Area under the curve (AUC)** represents the area under the ROC curve in the given testing

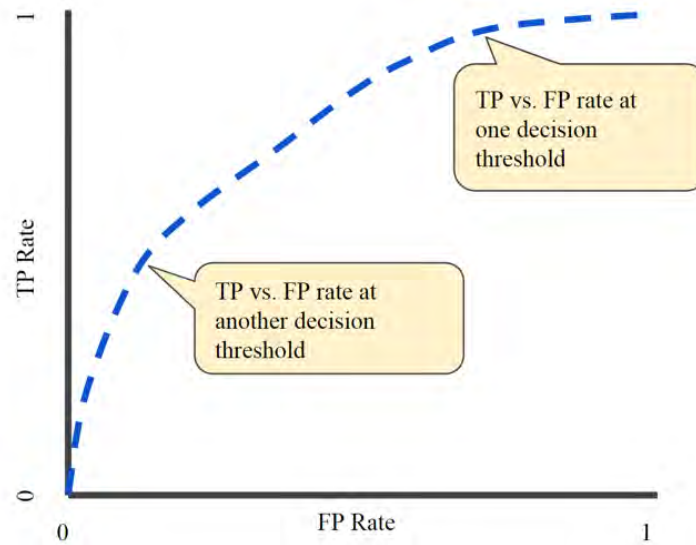


Figure 5.9: A simple illustration of the ROC curve

dataset. Assume that the ROC curve is formed by the sequential connection of points with coordinates of  $(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_m, y_m)$ . AUC can be calculated as,

$$AUC = \frac{1}{2} \sum_{i=1}^{m-1} (x_{i+1} - x_i) \cdot (y_i + y_{i+1}) \quad (5.10)$$

where AUC values range from 0.5 to 1.0, and a classifier with a larger AUC value has better performance.

Table 5.3 lists the basic evaluation measures along with their formulas computed from the confusion matrix.

Table 5.3: A summary of evaluation metrics based on the confusion matrix

Evaluation Metric	Formula
Accuracy	$(TP + TN)/(TP + TN + FN + FP)$
Precision	$TP/(TP + FP)$
Recall or True Positive Rate (TPR)	$TP/(TP + FN)$
False Positive Rate (FPR)	$FP/(FP + TN)$
F1-score	$(2 \times Precision \times Recall)/(Precision + Recall)$
Matthews Correlation Coefficient (MCC)	$(TP \times TN - FP \times FN)/((TP + FP)(TP + FN)(TN + FP)(TN + FN))^{1/2}$

## 5.5 Experimental Result

For performance evaluation, we considered four popular ML classification models that exhibited strong performance in credit default prediction tasks: Logistic Regression (LR) [69], Random Forest (RF) [82], Light Gradient Boosting Machine (LightGBM) [84], and Extreme Gradient Boosting algorithm (XGBoost) [85]. We reserved 20% of the dataset for testing purposes. There are several hyper-parameters in each classification model that requires fine-tuning. We utilized the grid-search algorithm to do hyper-parameter tuning for each machine learning model. We used the 5-fold cross-validation technique to determine the optimum hyper-parameter.

The experimental results of various machine learning models with and without knowledge graph embedding features are shown in Table 5.4 Model names with +KGE mean that knowledge graph embeddings are used as features for model training. It is obvious from Table 5.4 that the machine learning models' performance improved significantly while knowledge graph embeddings were used directly as features. LR and RF have the lowest predictive power among these models because they are highly dependent on robust feature engineering.

Table 5.4: The performance comparison of machine learning model using knowledge graph embeddings

ML Model	Accuracy (%)	Precision (%)	Recall (%)	FPR (%)	F1 score	MCC	ROC AUC	AUC Gain(%)
LR	70.00	10.71	37.48	27.18	0.166	0.062	0.607	–
LR + KGE	72.28	13.87	45.22	25.29	0.212	0.123	0.627	3.29
RF	74.57	16.29	52.60	23.52	0.235	0.248	0.651	–
RF + KGE	88.96	34.39	44.73	10.93	0.377	0.318	0.695	6.76
LightGBM	83.21	20.80	39.13	12.96	0.271	0.198	0.719	–
LightGBM + KGE	94.02	70.29	43.78	5.91	0.539	0.525	0.796	10.71
XGBoost	94.10	74.65	49.78	5.18	0.519	0.518	0.759	–
<b>XGBoost + KGE</b>	<b>96.79</b>	<b>80.83</b>	<b>78.75</b>	<b>1.62</b>	<b>0.79</b>	<b>0.78</b>	<b>0.836</b>	<b>10.14</b>

Figure 5.10 shows the comparison of ROC curves of different classifiers. Including knowledge graphs' semantic and structural information as features considerably improves the AUC score of the LightGBM and XGBoost models by 10.71% and 10.14%, respectively, while moderately enhancing the performance of the LR and RF methods by 3.29% and 6.76%, respectively. XGBoost + KGE exhibited strong performance in all evaluation measures. Figure 5.11 depicts the confusion matrix of the XGBoost + KGE model. We set the number of trees to 500, the maximum depth of each tree to 5, the learning rate to 0.001, and gamma to 0.1 as XGBoost

hyper-parameters while training the model with the knowledge embedding features.

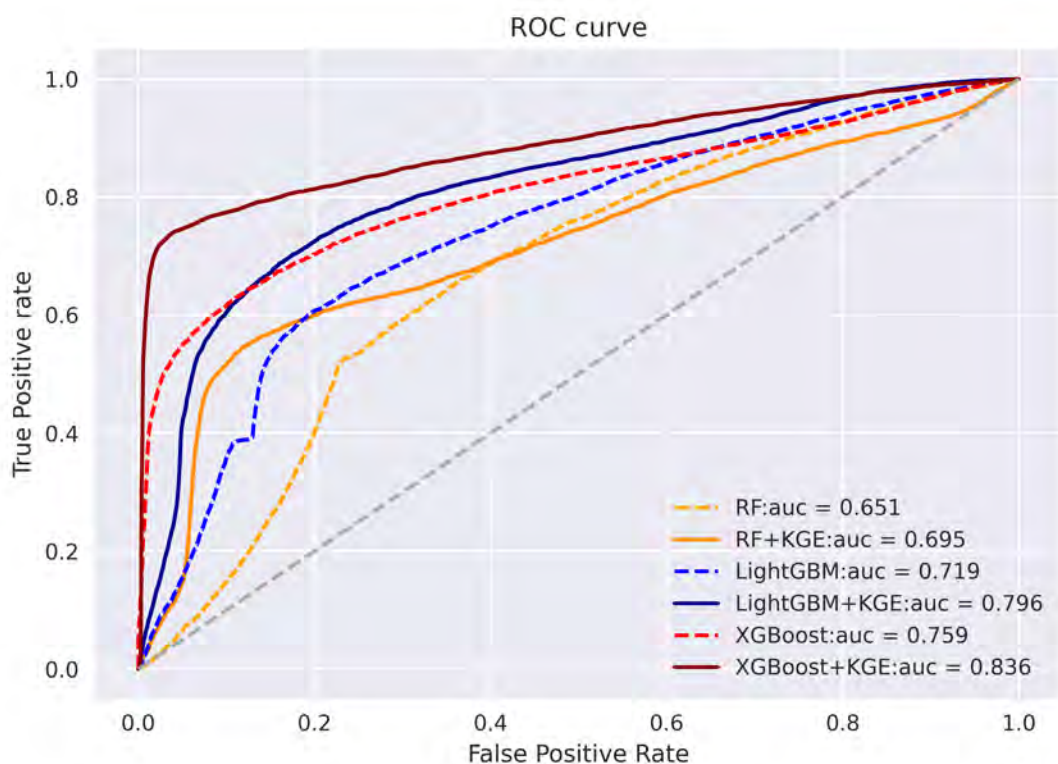


Figure 5.10: Comparison of the ROC curves of different ML classification models with and without KGE features

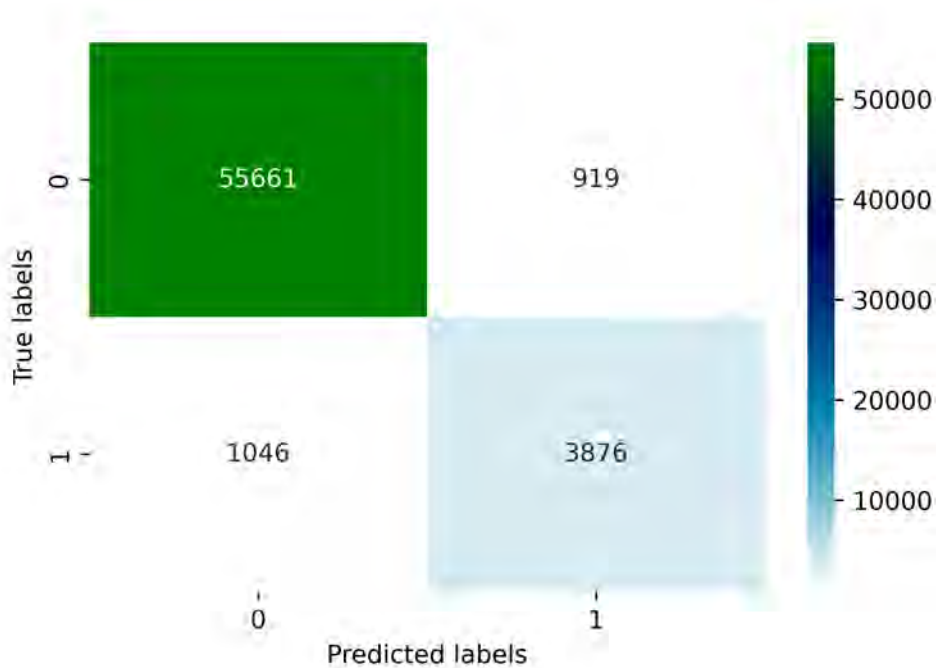


Figure 5.11: Confusion matrix for 'XGBoost + KGE' model

## 5.6 Analytical Discussion

Credit risk management is critical for lending-based financial institutions. Financial institutions may suffer significant financial losses when borrowers default, i.e., fail to repay the loan. Hence, they rely on statistical and machine learning approaches for assessing credit risk objectively. These machine learning applications concentrate only on the prediction tasks and are not interpretable. Credit approval is a critical decision for financial institutions, and both model creators and regulators want a causal explanation of the prediction model. Knowledge graph technology is the best way to make the prediction model interpretable as it provides the semantic context of the data (see Section 2.6). Thus, machine learning applications can unlock their full potential while integrated with knowledge graph technology.

The main objective of this study is to formulate the loan default risk prediction as a classification problem within the knowledge graph embedding space by computing similarities between loan applicants. To achieve this, we first investigated the datasets (published by Home Credit) by employing exploratory data analysis techniques, including data preprocessing activities, to ensure their quality for knowledge graph construction. Then, we mapped the consumer attributes and financial states with the semantic data model and put them into a knowledge graph embedding space for the consumption of machine learning models.

Knowledge graph embedding is a way to input domain knowledge expressed in a knowledge graph into a machine learning algorithm, providing far better improvements in feature engineering (see Section 2.5.2). We trained conventional machine learning models with the knowledge graph embeddings as the input features. The experimental results demonstrated that the machine learning models' performance improved significantly while knowledge graph embeddings were used directly as features. Since it preserves both semantic information and structural content of the knowledge graph, utilizing these embeddings improved the AUC score of the LightGBM and XGBoost models by 10.71% and 10.14%, respectively (see Table 5.4). However, the performance of the XGBoost model in all evaluation measures is higher than that of traditional machine learning-based models.

The integration of knowledge graph technology into machine learning models facilitates the explainability and fairness of their results. Regulators expect banks and financial organizations to be transparent in their credit decision-making process. Moreover, a borrower should know why their loan application has been declined. The knowledge graph provides a human-friendly way of evaluating related data, allowing humans to visualize the decision-making process taken by the system.

# Chapter 6

## Conclusion and Future Work

This final chapter summarizes our research and related outcomes associated with the thesis, followed by some directions for future research to advance the performance and rationality of the loan default prediction model.

### 6.1 Summary and Conclusion

In this thesis, we proposed a loan default prediction model using knowledge graph technology to reduce the credit risk of lending organizations. We developed a semantic data model and constructed a knowledge graph with publicly available credit data. Next, we formulate the loan default risk prediction as a binary classification problem within the knowledge graph embedding space by computing similarities among loan applicants. For this, we mapped the applicant attributes and financial states into a knowledge graph embedding space for the consumption of machine learning models. We used the knowledge graph embeddings as features in the machine learning classifier to forecast the loan default. The experimental results indicated that incorporating knowledge graph embeddings as features can significantly improve the prediction performance compared to conventional machine learning classifiers. We used accuracy, precision, recall, F1 score, MCC, and ROC AUC as evaluation metrics to measure the performance of different machine learning classifiers. The “XGBoost + KGE” model demonstrated strong performance in all evaluation measures, with a ROC AUC of 0.836 (a 10.14% gain compared to the traditional approach).

Banks and financial institutions can lend money instantly with a reliable loan default prediction model. Consequently, more people could have access to credit. Using knowledge graph technology, financial institutions can view and analyze all their customers’ information, risk dimensions, laws, and regulations in one location, all correlated based on their significance for in-depth analysis.



## 6.2 Future Research Direction

From this thesis, we can highlight some of the scopes of future work to step forward:

- LightGBM and XGboost were trained for the boosting models, although other boosting approaches or machine learning algorithms (e.g., AdaBoost, Support Vector Machine) can also be used to compare the performance.
- The financial market changes rapidly every day, and people's economic status and performance are affected by the market all the time. We can add more dimensions to the knowledge graph from various unstructured data sources, e.g., social media, sentiment analysis, macroeconomic indicators, and other alternative data.
- We can use natural language processing (NLP) and deep learning technology to extract entities and relations from the unstructured data sources to enrich the knowledge graph, which will further enhance the performance and rationality of the loan default prediction model.
- Knowledge graph embedding is developing rapidly, and new models are being proposed every year. Other knowledge graph embedding models can be tested to find the most appropriate method.
- The proposed approach is based on the semantic data model, which can be improved further by combining it with the domain expert's knowledge and experience. Moreover, automated tools can be explored and used for knowledge extraction from graphs.
- Fuzzy logic can also be employed, utilizing the knowledge graph technology to evaluate a loan applicant's creditworthiness and predict their probability of default into three classes: high risk, medium risk, and low risk. It is a form of approximate reasoning which is based on "degrees of truth" as opposed to binary (0 or 1) classification. The fuzzy inference system (FIS) enables domain specialists to articulate their knowledge in the form of fuzzy rules, allowing machine prediction to be combined with human judgment.

# Publications

M. N. Alam and M. M. Ali, “Loan default risk prediction using knowledge graph,” in *2022 14th International Conference on Knowledge and Smart Technology (KST'2022)*, IEEE, Jan. 2022, held at Burapha University, Chonburi, Thailand.

# References

- [1] “Semantic web stack.” Last accessed on January 19, 2022, at 02:35:00PM. [Online]. Available: [https://en.wikipedia.org/wiki/Semantic\\_Web\\_Stack](https://en.wikipedia.org/wiki/Semantic_Web_Stack).
- [2] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, “A survey on knowledge graphs: Representation, acquisition, and applications,” *IEEE Transactions on Neural Networks and Learning Systems*, 2 2021.
- [3] Z. Wang, J. Zhang, J. Feng, and Z. Chen, “Knowledge graph embedding by translating on hyperplanes,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 28, (Québec, Canada), p. 1112–1119, AAAI Press, 2014.
- [4] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, “Learning entity and relation embeddings for knowledge graph completion,” in *Twenty-ninth AAAI conference on artificial intelligence*, 2015.
- [5] Q. Wang, Z. Mao, B. Wang, and L. Guo, “Knowledge graph embedding: A survey of approaches and applications,” 12 2017.
- [6] M. Nickel, L. Rosasco, and T. Poggio, “Holographic embeddings of knowledge graphs,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, 2016.
- [7] “Knowledge graph embedding.” Last accessed on January 19, 2022, at 02:02:00PM. [Online]. Available: [https://en.wikipedia.org/wiki/Knowledge\\_graph\\_embedding](https://en.wikipedia.org/wiki/Knowledge_graph_embedding).
- [8] “Home credit default risk,” 2018. Last accessed on January 2, 2022, at 11:30:00PM. [Online]. Available: <https://www.kaggle.com/c/home-credit-default-risk/data>.
- [9] M. Qamruzzaman and W. Jianguo, “Financial innovation and economic growth in bangladesh,” *Financial Innovation*, vol. 3, no. 1, p. 19, 2017.
- [10] A. Mashrur, W. Luo, N. A. Zaidi, and A. Robles-Kelly, “Machine learning for financial risk management: A survey,” *IEEE Access*, vol. 8, pp. 203203–203223, 11 2020.
- [11] “Knowledge graphs for financial services,” 2020. Last accessed on January 21, 2022, at 02:08:00PM. [Online]. Available:

- <https://www2.deloitte.com/content/dam/Deloitte/nl/Documents/risk/deloitte-nl-risk-knowledge-graphs-financial-services.pdf>.
- [12] L. Ehrlinger and W. Wöß, “Towards a definition of knowledge graphs.,” *SEMANTiCS (Posters, Demos, SuCCESS)*, vol. 48, no. 1-4, p. 2, 2016.
- [13] S. B. Obenland Philipp, Schoof Ulrich, “Wisdom of enterprise knowledge graphs,” 2019. Last accessed on January 29, 2022, at 08:30:00PM. [Online]. Available: <https://www2.deloitte.com/content/dam/Deloitte/de/Documents/operations/knowledge-graphs-pov.pdf>.
- [14] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.
- [15] W. Y. Lin, Y. H. Hu, and C. F. Tsai, “Machine learning in financial crisis prediction: A survey,” 2012.
- [16] C.-F. Tsai, “Financial decision support using neural networks and support vector machines,” *Expert Systems*, vol. 25, no. 4, pp. 380–393, 2008.
- [17] H. Kvamme, N. Sellereite, K. Aas, and S. Sjørusen, “Predicting mortgage default using convolutional neural networks,” *Expert Systems with Applications*, vol. 102, pp. 207–217, 2018.
- [18] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, “Translating embeddings for modeling multi-relational data,” *Advances in neural information processing systems*, vol. 26, p. 2787–2795, 2013.
- [19] C. Moon, P. Jones, and N. F. Samatova, “Learning entity type embeddings for knowledge graph completion,” in *Proceedings of the 2017 ACM on conference on information and knowledge management*, (Singapore), pp. 2215–2218, ACM, 2017.
- [20] T. Trouillon, J. Welbl, S. Riedel, E. Ciaussier, and G. Bouchard, “Complex embeddings for simple link prediction,” in *33rd International Conference on Machine Learning, ICML 2016*, vol. 5, (New York City, NY, USA), pp. 3021–3032, JMLR.org, 2016.
- [21] P. O. Kelliher, D. Wilmot, J. Vij, and P. J. Klumpes, “A common risk classification system for the actuarial profession,” *British Actuarial Journal*, vol. 18, no. 1, pp. 91–121, 2013.
- [22] M. Beyhaghi and J. P. Hawley, “Modern portfolio theory and risk management: assumptions and unintended consequences,” *Journal of Sustainable Finance & Investment*, vol. 3, no. 1, pp. 17–37, 2013.

- [23] “Principles for the sound management of operational risk.” Last accessed on January 12, 2022, at 04:09:00PM. [Online]. Available: <https://www.bis.org/publ/bcbs195.pdf>.
- [24] “Credit risk analysis models.” Last accessed on January 17, 2022, at 04:02:00PM. [Online]. Available: <https://corporatefinanceinstitute.com/resources/knowledge/credit/credit-risk-analysis-models/>.
- [25] J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare, “Credit card fraud detection using machine learning techniques: A comparative analysis,” in *2017 International Conference on Computing Networking and Informatics (ICCNi)*, pp. 1–9, IEEE, 2017.
- [26] Z. Chen, L. D. Van Khoa, E. N. Teoh, A. Nazir, E. K. Karuppiah, and K. S. Lam, “Machine learning techniques for anti-money laundering (aml) solutions in suspicious transaction detection: a review,” *Knowledge and Information Systems*, vol. 57, no. 2, pp. 245–285, 2018.
- [27] K. Nian, H. Zhang, A. Tayal, T. Coleman, and Y. Li, “Auto insurance fraud detection using unsupervised spectral ranking for anomaly,” *The Journal of Finance and Data Science*, vol. 2, no. 1, pp. 58–75, 2016.
- [28] C. Phua, V. Lee, K. Smith, and R. Gayler, “A comprehensive survey of data mining-based fraud detection research,” *arXiv preprint arXiv:1009.6119*, 2010.
- [29] T. Berners-Lee, J. Hendler, and O. Lassila, “The semantic web,” *Scientific american*, vol. 284, no. 5, pp. 34–43, 2001.
- [30] L. Yu, *A developer’s guide to the semantic Web*. Springer Science & Business Media, 2011.
- [31] “Semantic web.” Last accessed on January 19, 2022, at 02:05:00PM. [Online]. Available: [https://www.semanticweb.org/wiki/Main\\_Page.html](https://www.semanticweb.org/wiki/Main_Page.html).
- [32] “Semantic web-w3c.” Last accessed on January 19, 2022, at 02:05:00PM. [Online]. Available: <https://www.w3.org/standards/semanticweb/>.
- [33] G. Antoniou and F. Van Harmelen, *A semantic web primer*. MIT press, 2004.
- [34] “Rdf 1.1 primer.” Last accessed on January 19, 2022, at 01:10:00PM. [Online]. Available: <https://www.w3.org/TR/rdf11-primer/>.
- [35] A. Maedche and S. Staab, “Ontology learning for the semantic web,” *IEEE Intelligent systems*, vol. 16, no. 2, pp. 72–79, 2001.
- [36] G. Antoniou and F. Van Harmelen, “Web ontology language: Owl,” in *Handbook on ontologies*, pp. 67–92, Springer, 2004.

- [37] N. F. Noy, "Semantic integration: a survey of ontology-based approaches," *ACM Sigmod Record*, vol. 33, no. 4, pp. 65–70, 2004.
- [38] "google blog." Last accessed on January 19, 2022, at 01:12:00PM. [Online]. Available: <https://blog.google/products/search/introducing-knowledge-graph-things-not/>.
- [39] A. Bordes, J. Weston, R. Collobert, and Y. Bengio, "Learning structured embeddings of knowledge bases," in *Twenty-fifth AAAI conference on artificial intelligence*, 2011.
- [40] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 1247–1250, 2008.
- [41] D. Vrandečić and M. Krötzsch, "Wikidata: a free collaborative knowledgebase," *Communications of the ACM*, vol. 57, no. 10, pp. 78–85, 2014.
- [42] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "Dbpedia: A nucleus for a web of open data," in *The semantic web*, pp. 722–735, Springer, 2007.
- [43] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: a core of semantic knowledge," in *Proceedings of the 16th international conference on World Wide Web*, pp. 697–706, 2007.
- [44] M. Kejriwal, *Domain-specific knowledge graph construction*. Springer, 2019.
- [45] B. Abu-Salih, "Domain-specific knowledge graphs: A survey," *Journal of Network and Computer Applications*, vol. 185, p. 103076, 2021.
- [46] Z. Zhao, S.-K. Han, and I.-M. So, "Architecture of knowledge graph construction techniques," *International Journal of Pure and Applied Mathematics*, vol. 118, no. 19, pp. 1869–1883, 2018.
- [47] X. Ma, "Knowledge graph construction and application in geosciences: A review," *EarthArXiv Online*, 2021.
- [48] T. Tudorache, N. F. Noy, S. Tu, and M. A. Musen, "Supporting collaborative ontology development in protégé," in *International Semantic Web Conference*, pp. 17–32, Springer, 2008.
- [49] A. Rossi, D. Barbosa, D. Firmani, A. Matinata, and P. Merialdo, "Knowledge graph embedding for link prediction: A comparative analysis," *ACM Transactions on Knowledge Discovery from Data*, vol. 15, 2 2021.
- [50] Y. Dai, S. Wang, N. N. Xiong, and W. Guo, "A survey on knowledge graph embedding: Approaches, applications and benchmarks," *Electronics*, vol. 9, no. 5, p. 750, 2020.

- [51] L. Cai and W. Y. Wang, “Kbgan: Adversarial learning for knowledge graph embeddings,” *arXiv preprint arXiv:1711.04071*, 2017.
- [52] M. Nickel, V. Tresp, and H.-P. Kriegel, “A three-way model for collective learning on multi-relational data,” in *Icml*, 2011.
- [53] R. Jenatton, N. Roux, A. Bordes, and G. R. Obozinski, “A latent factor model for highly multi-relational data,” *Advances in neural information processing systems*, vol. 25, 2012.
- [54] B. Yang, W.-t. Yih, X. He, J. Gao, and L. Deng, “Embedding entities and relations for learning and inference in knowledge bases,” *arXiv preprint arXiv:1412.6575*, 2014.
- [55] G. Bouchard, S. Singh, and T. Trouillon, “On approximate reasoning capabilities of low-rank vector spaces,” in *2015 AAAI Spring Symposium Series*, 2015.
- [56] X. Zou, “A survey on application of knowledge graph,” in *Journal of Physics: Conference Series*, vol. 1487, p. 012016, IOP Publishing, 2020.
- [57] “Knowledge graphs.” Last accessed on January 19, 2022, at 02:02:00PM. [Online]. Available: [https://web.stanford.edu/class/cs520/2020/notes/What\\_is\\_a\\_Knowledge\\_Graph.html](https://web.stanford.edu/class/cs520/2020/notes/What_is_a_Knowledge_Graph.html).
- [58] B. Abu-Salih, M. Al-Tawil, I. Aljarah, H. Faris, P. Wongthongtham, K. Y. Chan, and A. Beheshti, “Relational learning analysis of social politics using knowledge graph embedding,” *Data Mining and Knowledge Discovery*, vol. 35, no. 4, pp. 1497–1536, 2021.
- [59] D. Q. Nguyen, “An overview of embedding models of entities and relationships for knowledge base completion,” *arXiv preprint arXiv:1703.08098*, 2017.
- [60] “Contextual AI: The next frontier of artificial intelligence.” Last accessed on January 29, 2022, at 05:05:00PM. [Online]. Available: <https://business.adobe.com/blog/perspectives/contextual-ai-the-next-frontier-of-artificial-intelligence>.
- [61] J. W. Jesus Barrasa, Amy E. Hodler, *Knowledge Graphs*. O’Reilly Media, Inc, 2021.
- [62] D. J. Hand and W. E. Henley, “Statistical classification methods in consumer credit scoring: a review,” *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, vol. 160, no. 3, pp. 523–541, 1997.
- [63] H. A. Abdou and J. Pointon, “Credit scoring, statistical techniques and evaluation criteria: a review of the literature,” *Intelligent systems in accounting, finance and management*, vol. 18, no. 2-3, pp. 59–88, 2011.

- [64] T.-S. Lee, C.-C. Chiu, C.-J. Lu, and I.-F. Chen, "Credit scoring using the hybrid neural discriminant technique," *Expert Systems with applications*, vol. 23, no. 3, pp. 245–254, 2002.
- [65] T.-S. Lee and I.-F. Chen, "A two-stage hybrid credit scoring model using artificial neural networks and multivariate adaptive regression splines," *Expert systems with applications*, vol. 28, no. 4, pp. 743–752, 2005.
- [66] M. Šušteršič, D. Mramor, and J. Zupan, "Consumer credit scoring models with limited data," *Expert Systems with Applications*, vol. 36, no. 3, pp. 4736–4744, 2009.
- [67] M.-C. Chen and S.-H. Huang, "Credit scoring and rejected instances reassigning through evolutionary computation techniques," *Expert Systems with Applications*, vol. 24, no. 4, pp. 433–441, 2003.
- [68] C.-L. Chuang and R.-H. Lin, "Constructing a reassigning credit scoring model," *Expert Systems with Applications*, vol. 36, no. 2, pp. 1685–1694, 2009.
- [69] X. Dastile, T. Celik, and M. Potsane, "Statistical and machine learning models in credit scoring: A systematic literature survey," *Applied Soft Computing*, vol. 91, p. 106263, 2020.
- [70] S. Lessmann, B. Baesens, H.-V. Seow, and L. C. Thomas, "Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research," *European Journal of Operational Research*, vol. 247, no. 1, pp. 124–136, 2015.
- [71] G. Wang, J. Hao, J. Ma, and H. Jiang, "A comparative assessment of ensemble learning for credit scoring," *Expert systems with applications*, vol. 38, no. 1, pp. 223–230, 2011.
- [72] E. I. Altman, "Financial ratios, discriminant analysis and the prediction of corporate bankruptcy," *The journal of finance*, vol. 23, no. 4, pp. 589–609, 1968.
- [73] H. A. Abdou and J. Pointon, "Credit scoring and decision making in egyptian public sector banks," *International Journal of Managerial Finance*, 2009.
- [74] T. M. Alam, K. Shaukat, I. A. Hameed, S. Luo, M. U. Sarwar, S. Shabbir, J. Li, and M. Khushi, "An investigation of credit card default prediction in the imbalanced datasets," *IEEE Access*, vol. 8, pp. 201173–201198, 2020.
- [75] H. S. Kim and S. Y. Sohn, "Support vector machines for default prediction of smes based on technology credit," *European Journal of Operational Research*, vol. 201, no. 3, pp. 838–846, 2010.



- [76] A. Chaudhuri and K. De, “Fuzzy support vector machine for bankruptcy prediction,” *Applied Soft Computing*, vol. 11, no. 2, pp. 2472–2486, 2011.
- [77] Y.-C. Lee, “Application of support vector machines to corporate credit rating prediction,” *Expert Systems with Applications*, vol. 33, no. 1, pp. 67–74, 2007.
- [78] D. West, “Neural network credit scoring models,” *Computers & operations research*, vol. 27, no. 11-12, pp. 1131–1152, 2000.
- [79] A. Keramati and N. Yousefi, “A proposed classification of data mining techniques in credit scoring,” in *the Proceeding of 2011 International Conference of Industrial Engineering and Operations Management, Kuala Lumpur, Malaysia, Jurnal*, pp. 22–4, 2011.
- [80] X. Chen, X. Liu, Z. Liu, P. Song, and M. Zhong, “A deep learning approach using deepgbm for credit assessment,” in *ACM International Conference Proceeding Series*, pp. 774–779, Association for Computing Machinery, 9 2019.
- [81] Y. Xia, C. Liu, Y. Li, and N. Liu, “A boosted decision tree approach using bayesian hyperparameter optimization for credit scoring,” *Expert Systems with Applications*, vol. 78, pp. 225–241, 2017.
- [82] L. Zhu, D. Qiu, D. Ergu, C. Ying, and K. Liu, “A study on predicting loan default based on the random forest algorithm,” *Procedia Computer Science*, vol. 162, pp. 503–513, 2019.
- [83] A. Petropoulos, V. Siakoulis, E. Stavroulakis, A. Klamargias, *et al.*, “A robust machine learning approach for credit risk analysis of large loan level datasets using deep learning and extreme gradient boosting,” *IFC Bulletins chapters*, vol. 49, 2019.
- [84] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, “Lightgbm: A highly efficient gradient boosting decision tree,” *Advances in neural information processing systems*, vol. 30, 2017.
- [85] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” *New York, NY, USA: ACM*, vol. 10, no. 2939672.2939785, pp. 785–794, 2016.
- [86] M. K. Lim and S. Y. Sohn, “Cluster-based dynamic scoring model,” *Expert Systems with Applications*, vol. 32, no. 2, pp. 427–431, 2007.
- [87] G. Kou, Y. Peng, and G. Wang, “Evaluation of clustering algorithms for financial risk analysis using MCDM methods,” *Information Sciences*, vol. 275, pp. 1–12, 2014.
- [88] C. Luo, D. Wu, and D. Wu, “A deep learning approach for credit scoring using credit default swaps,” *Engineering Applications of Artificial Intelligence*, vol. 65, pp. 465–470, 2017.

- [89] V.-E. Neagoie, A.-D. Ciotec, and G.-S. Cucu, “Deep convolutional neural networks versus multilayer perceptron for financial prediction,” in *2018 International Conference on Communications (COMM)*, pp. 201–206, IEEE, 2018.
- [90] S. Hamori, M. Kawai, T. Kume, Y. Murakami, and C. Watanabe, “Ensemble learning or deep learning? application to default risk analysis,” *Journal of Risk and Financial Management*, vol. 11, no. 1, p. 12, 2018.
- [91] M. Chi, S. Hongyan, W. Shaofan, L. Shengliang, and L. Jingyan, “Bond default prediction based on deep learning and knowledge graph technology,” *IEEE Access*, vol. 9, pp. 12750–12761, 2021.
- [92] K. Hussain and E. Prieto, “Big data in the finance and insurance sectors,” in *New horizons for a data-driven economy*, pp. 209–223, Springer, Cham, 2016.
- [93] J. Unbehauen, S. Hellmann, S. Auer, and C. Stadler, “Knowledge extraction from structured sources,” *Search computing*, pp. 34–52, 2012.
- [94] A. Gangemi and V. Presutti, “Ontology design patterns,” in *Handbook on ontologies*, pp. 221–243, Springer, 2009.
- [95] P. Castells, B. Foncillas, R. Lara, M. Rico, and J. L. Alonso, “Semantic web technologies for economic and financial information management,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3053, pp. 473–487, 2004.
- [96] Enterprise Data Management Council, “The financial industry business ontology,” 2021. Last accessed on January 20, 2022, at 10:30:00PM. [Online]. Available: <https://spec.edmcouncil.org/fibo/>.
- [97] “The neo4j graph data platform.” Last accessed on January 21, 2022, at 9:24:00PM. [Online]. Available: <https://neo4j.com/product/>.
- [98] H.-G. Yoon, H.-J. Song, S.-B. Park, and S.-Y. Park, “A translation-based knowledge graph embedding preserving logical property of relations,” in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 907–916, 2016.
- [99] A. Nabil, *Data Science in FinTech: credit risk prediction using Deep Learning*. PhD thesis, ETSI-Informatica, 2020.
- [100] “Exploratory-data-analysis.” Last accessed on January 29, 2022, at 02:05:00PM. [Online]. Available: <https://www.kaggle.com/willkoehrsen/start-here-a-gentle-introduction#Exploratory-Data-Analysis>.

- [101] “Min-max-scaler.” Last accessed on January 29, 2022, at 02:25:00PM. [Online]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html>.
- [102] “Encoding categorical variables.” Last accessed on February 21, 2022, at 07:05:00PM. [Online]. Available: <https://kiwidamien.github.io/encoding-categorical-variables.html>.
- [103] R. Emekter, Y. Tu, B. Jirasakuldech, and M. Lu, “Evaluating credit risk and loan performance in online peer-to-peer (p2p) lending,” *Applied Economics*, vol. 47, no. 1, pp. 54–70, 2015.
- [104] L. Costabello, S. Pai, C. L. Van, R. McGrath, N. McCarthy, and P. Tabacof, “Ampligraph: a library for representation learning on knowledge graphs,” 3 2019.
- [105] D. M. Powers, “Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation,” *arXiv preprint arXiv:2010.16061*, 2020.
- [106] “Matthews correlation coefficient(mcc).” Last accessed on January 31, 2022, at 05:05:00PM. [Online]. Available: [https://en.wikipedia.org/wiki/Phi\\_coefficient](https://en.wikipedia.org/wiki/Phi_coefficient).
- [107] D. Chicco and G. Jurman, “The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation,” *BMC genomics*, vol. 21, no. 1, pp. 1–13, 2020.
- [108] T. Fawcett, “An introduction to roc analysis,” *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.

Generated using Postgraduate Thesis L<sup>A</sup>T<sub>E</sub>X Template, Version 1.03. Department of  
Computer Science and Engineering, Bangladesh University of Engineering and  
Technology, Dhaka, Bangladesh.

This thesis was generated on Thursday 24<sup>th</sup> March, 2022 at 4:56am.