# GENERIC PREDICTIVE COMPUTATIONAL MODELLING FOR FINANCIAL DATA USE CASE: APPLYING ONTOLOGIES AS FINANCIAL DATA PRE-PROCESSING TOOL

**Natalia Yerashenia, MSc**
*Computer Science PhD Researcher*
*University of Westminster, London*

**Dr. Alexander Bolotov**
*Principal Lecturer, School of Computer Science and Engineering*
*University of Westminster, London*

UNIVERSITY OF WESTMINSTER

Stanford University

# Overview

## GENERIC MODEL:

- Main Aim
- Input **financial/market data**: Big Four 'V' + 'R'
- Ontology as the core data pre-processing component of the Model
- Essential Definitions
- **Graph Database** Component Development
- **Machine Learning Engine** Overview (inc. Feedback Loop)

## USE CASE:

- The features of model building and implementation
- **Bankruptcy Prediction Computational Model** Components and Dataflow
- **Ontology of Bankruptcy Prediction** Development
- **Graph Database** Component Development
- **Machine Learning Engine** Overview
- Accomplished to date & Contribution to knowledge
- Published Papers to date

## Main Aim

To form a Use Case based on Generic Computation Model for Data Analysis

**USE CASE:**

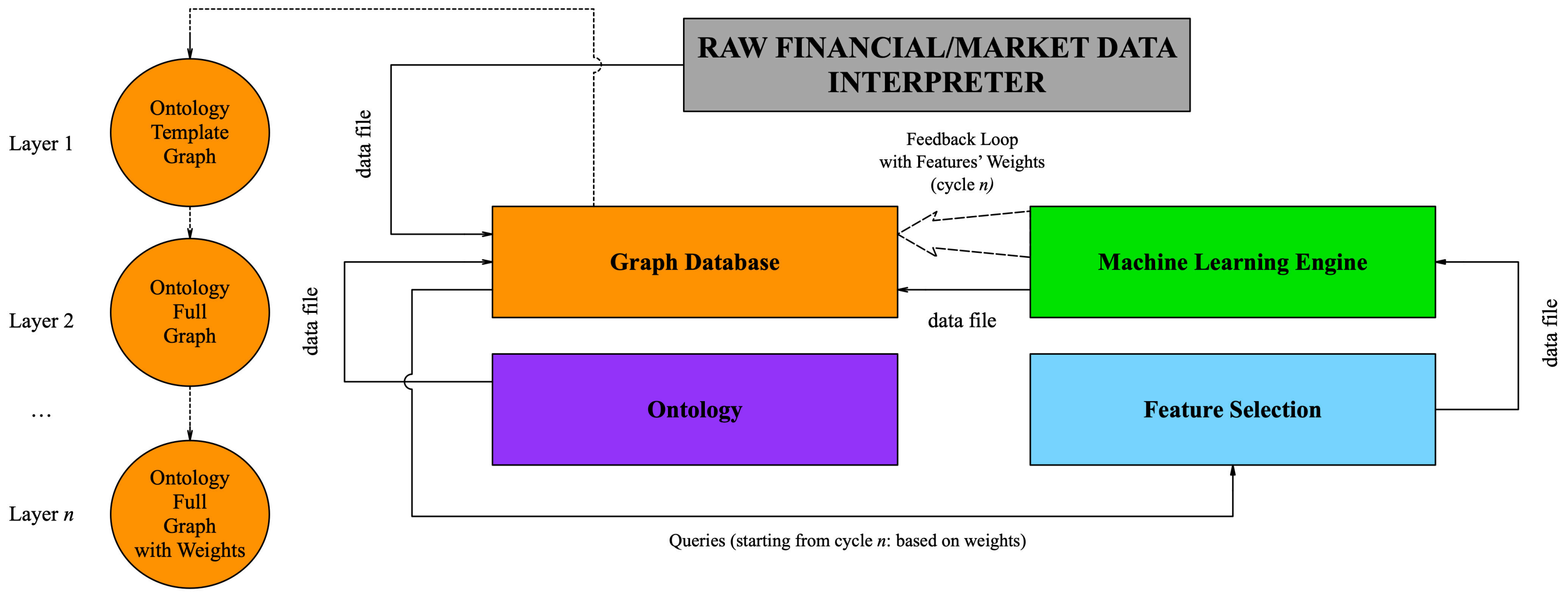
**Bankruptcy Prediction Computational Model (BPCM)**

BPCM processes financial information of a company and carries out a comprehensive investigation of companies' financial activities during a particular time period.

**This involves:** modern methods of searching, processing and storing a heterogeneous data, advanced machine learning methods.

We aim to create a generic methodology of data pre-processing and analysis, which can be applied to various financial/market data types

# Generic Computational Model Components and Dataflow



# Essential Definitions

**An Ontology full graph with weights** is a labelled graph

$$G_w = <V,E,L_w>$$

- **V** is the set of vertices (features),
- **E** is the set of edges (relationships),
- **$L_w$** is the set of four types of labels: {value:0, value:i, weight:0, weight:k}
  - *value:0* the value is not yet identified
  - *value:i* *i* ranges over the real dataset values
  - *weight:0* the feature is irrelevant to this particular model's cycle'
  - *weight: k* ranges over the weights (0,1] which were added as the result of analysis executed during the previous model's cycle; *k=1* indicates a direct dependence between the final outcome of the model and the feature's value.

**EXAMPLE:** **ratioID:** Gearing **value:** 31.16429818 **weight:** 0.5771

*taken from the use case, Gearing feature node, Neo4j environment

# Essential Definitions

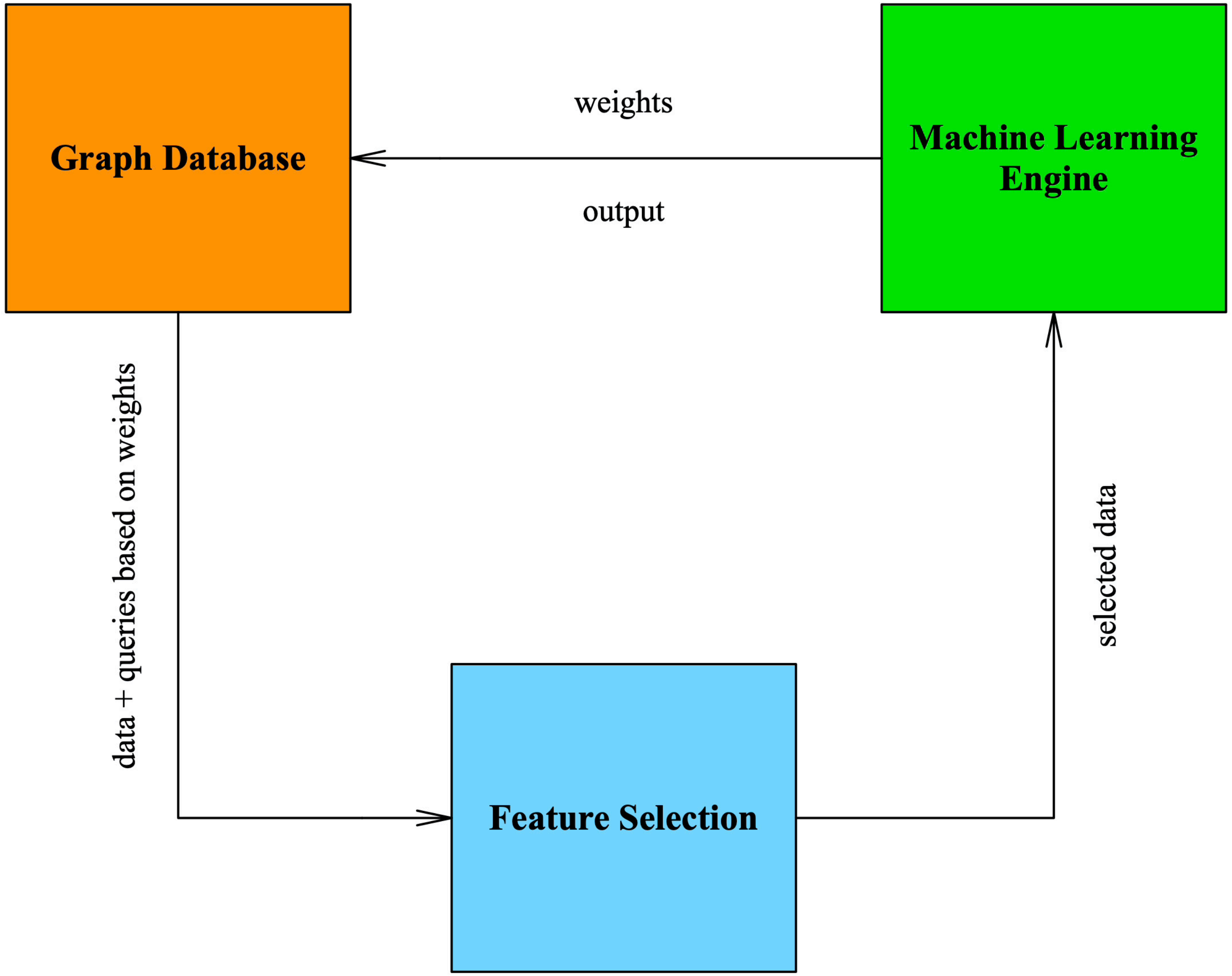
**An Ontology full graph** is a labelled graph

G= <V,E,L>, where

- *V* is the set of vertices,
- *E* is the set of edges, and
- *L* is the set of two types of labels {*value:0, value:i*}
  - *value:0* indicates ‘the value is not yet identified’ and
  - *i* ranges over the real dataset values

**An Ontology template graph** is a labelled graph which, as the full graph, contains a full set of vertices, edges, and labels exported from OBP ontology, however the values in labels are not identified. In other words, it’s a graph which contains no information about a particular company.

# Feedback Loop



# Financial / Market Data Features

## Big Data

Variety
Velocity
Veracity
Value
Volume

**5V**

## Financial Data

**(input/output data for BPCM)**

Variety
Velocity
Veracity
Value
~~Volume~~
Relationships

**4V + R**

# Why do we use Ontology?

## Ontology Features:

- Formal description of knowledge as a set of concepts
  - Synonyms
  - Definitions
  - Meaning of Data
- Relationships between concepts: hierarchical & non-hierarchical
- Mapping and integration
- Reasoning  (e.g. detects concentrations of concepts)

**Inspired by** FIBO

# Generic Steps of Ontology Creating


*Source:* Aranguren, M.E., Wroe, C., Goble, C. and Stevens, R., 2008. In situ migration of handcrafted ontologies to reason-able forms. *Data & knowledge engineering*, *66*(1), pp.147-162.

# Steps of developing a Graph Database

**STEP 1**
Importing the Ontology files (e.g. OWL or RDF format) into the Graph Database

**STEP 2**
Creating hierarchical & non-hierarchical connections between the ontologies/taxonomies (if there are two and more ontology files)

**STEP 3**
Matching the node labels with the indicators' names in the input data file taken from the external data sources (e.g. CSV file from traditional database).

**STEP 4**
The quantitative input data from the external data sources is transferred to graph database as ”value” attributes of corresponded nodes.

**STEP 5**
Calculating the value attributes of the features (if required) by using graph data base built-in math formulae

**STEP 6**
The Graph Database is ready to be used for Feature Selection and analytical purposes by means of queries

**STEP 7**
Exporting the values of the features (e.g. as CSV file)

**STEP 8**
Receiving the file with weights of the features (e.g. CSV file) from ML engine

# Building Machine Learning Engine

**Step 1**

Importing the files:

- Training Data Input & Output Data
- Input Data: exported from the graph database

**Step 2**

Designing and Training the Neural Network relevant for particular data type

**Step 3**

Usage and diagnostics

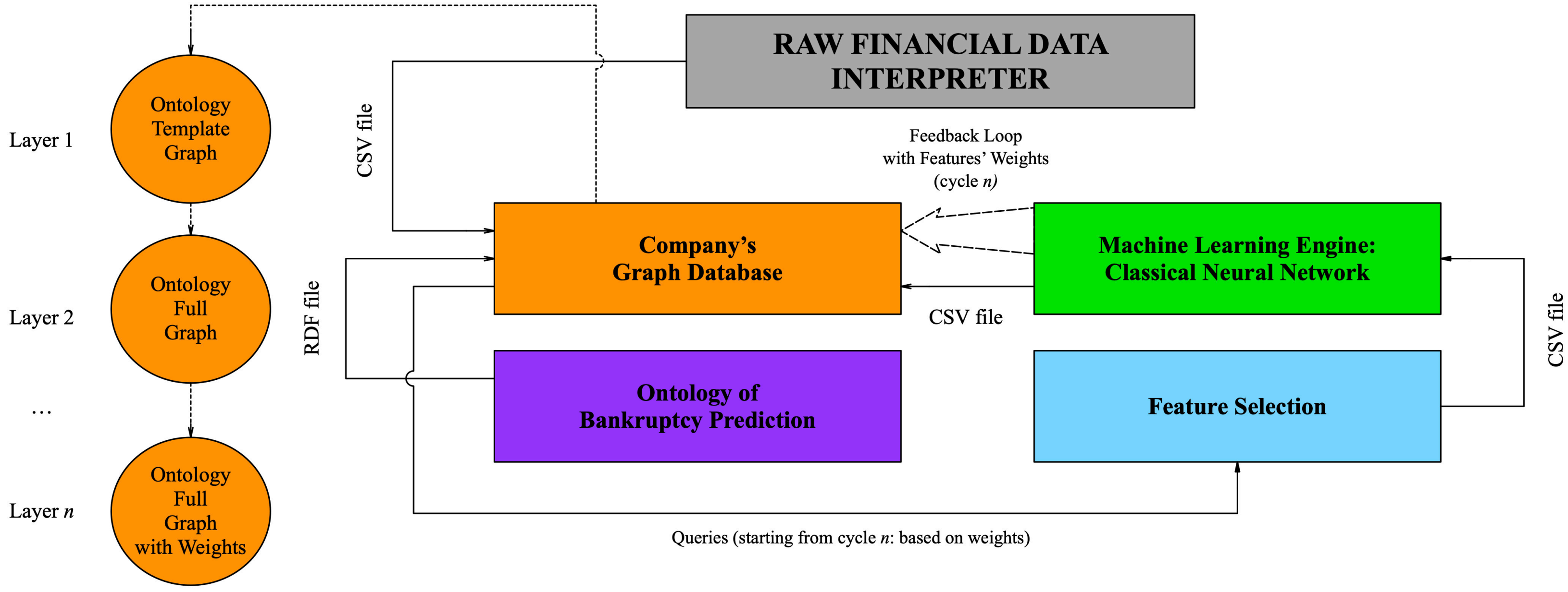
**Result:**

- Output Data
- Used Features' Weights, required for the next cycle

**Step 4**

Transfer the Features' Weights using Feedback Loop

# Bankruptcy Prediction Computational Model Components and Dataflow



# Financial Data used for the Use Case

| | Company name | Number of employees Last avail. yr | Increase(Decrease) Cash & Equiv. th GBP Last avail. yr | Turnover th GBP Last avail. yr | Cost of Sales th GBP Last avail. yr | Gross Profit th GBP Last avail. yr | Interest Paid th GBP Last avail. yr | Profit (Loss) before Interest paid th GBP Last avail. yr | Profit (Loss) after Tax th GBP Last avail. yr | Trade Debtors th GBP Last avail. yr | Stock & W.I.P. th GBP Last avail. yr | Current Assets th GBP Last avail. yr | Net ass th GBP Last |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1. | | 153.00 | 1469.21 | 44156.53 | -37676.83 | 6479.70 | -10.66 | 3867.86 | 3179.22 | 394.67 | 276.73 | 21778.06 | 12277.8 |
| 2. | | 143.00 | 1639.29 | 40289.57 | -35287.93 | 5001.64 | -10.66 | 3081.57 | 2536.09 | 16.83 | 276.73 | 21049.32 | 12093.9 |
| 3. | | 100.00 | 470.20 | 35074.63 | -22270.24 | 12804.39 | -9.27 | 1081.87 | 753.30 | 7028.41 | 9944.43 | 22216.02 | 25944.5 |
| 4. | | 71.00 | 492.29 | 29639.57 | -27133.50 | 2506.08 | -2.69 | 1386.08 | 1104.45 | 2908.14 | 943.26 | 5432.96 | 10175.8 |
| 5. | | 135.00 | 670.31 | 28002.46 | -21851.05 | 6151.41 | -1.31 | 1661.69 | 1346.59 | 4876.37 | 3446.85 | 11501.34 | 9640.3 |
| 6. | | 229.00 | 4293.66 | 26333.42 | -6411.22 | 19922.20 | -138.81 | 8324.92 | 7272.43 | 3794.49 | 2396.95 | 21358.48 | 31006.3 |
| 7. | | 94.00 | 804.49 | [illegible] | -21085.01 | [illegible] | [illegible] | 1246.94 | 1038.59 | 5866.62 | 3431.87 | 9616.18 | 11423.8 |
| 8. | | 188.00 | 1587.83 | 23871.38 | -15454.34 | 8417.04 | -15.50 | 1962.92 | 1569.06 | 2042.51 | 630.38 | 7444.37 | 7355.6 |
| 9. | | 72.00 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 9557.57 | 10910.5 |
| 10. | | 24.00 | 623.71 | 16685.27 | -13669.29 | 3015.97 | -21.88 | 844.08 | 657.44 | 2280.04 | 1433.56 | 5284.47 | 3318.7 |
| 11. | | 114.00 | 875.20 | 15547.87 | -10396.36 | 5151.51 | -20.67 | 1715.21 | 1482.69 | 2285.05 | 485.54 | 6920.77 | 8741.9 |
| 12. | | 178.00 | 586.09 | 14802.86 | -9027.35 | 5775.51 | -4.56 | 1776.36 | 1565.08 | 1571.60 | 2150.98 | 6153.15 | 7855.3 |
| 13. | | 96.00 | 1085.65 | 13423.82 | -10668.17 | 2755.65 | -11.99 | 1942.75 | 1557.51 | 4068.05 | 25.00 | 9481.04 | 6612.0 |
| 14. | | 104.00 | 263.56 | 13242.91 | -8781.59 | 4461.32 | -0.81 | 1885.84 | 1517.50 | 3075.38 | 896.82 | 4993.62 | 21183.0 |
| 15. | | 64.00 | 627.35 | 11404.10 | -8480.43 | 2923.67 | -0.65 | 512.26 | 405.91 | 1905.08 | 37.18 | 3574.31 | 2636.1 |
| 16. | | 72.00 | 1614.11 | 11340.23 | -5431.14 | 5909.09 | -0.11 | 1429.97 | 1265.23 | 2367.64 | 1448.89 | 7908.07 | 3517.3 |
| 17. | | 105.00 | 2180.69 | 11337.06 | -7019.60 | 4317.45 | -1.50 | 2421.62 | 1928.99 | 1392.40 | 1089.30 | 7123.93 | 8158.0 |
| 18. | | 57.00 | 1024.35 | 10821.84 | -6060.91 | 4760.93 | -1.80 | 408.79 | 258.03 | 606.04 | 2041.02 | 6186.66 | 5221.5 |

**Financial Records of 45 UK middle-size companies**

## Data Sources:

- Fame Bureau Van Dijk (https://fame.bvdinfo.com/)
- Annual reports and Independent Auditors' Reports, provided at Companies House website (https://beta.companieshouse.gov.uk).

# Ontology of Bankruptcy Prediction (OBP) Content

**Based on:**

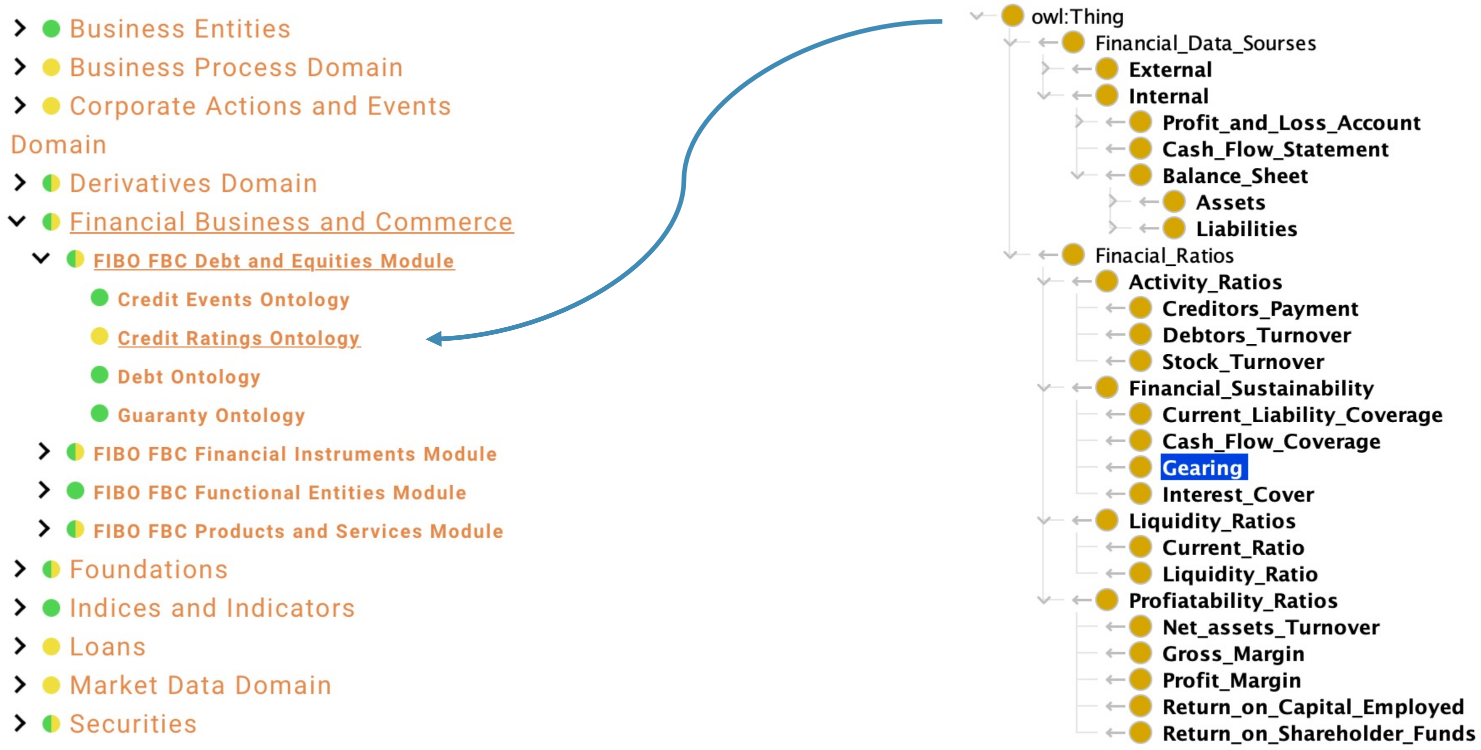
UK and international legislation which set the principles for the preparation and analysis of financial statements

- IFRS Standards https://www.ifrs.org
- UK Companies Act 2006 http://www.legislation.gov.uk/ukpga/2006/46/contents
- International Standards on Auditing (ISA) https://www.iaasb.org/standards-pronouncements
- etc

# OBP vs FIBO

## FIBO

## OBP

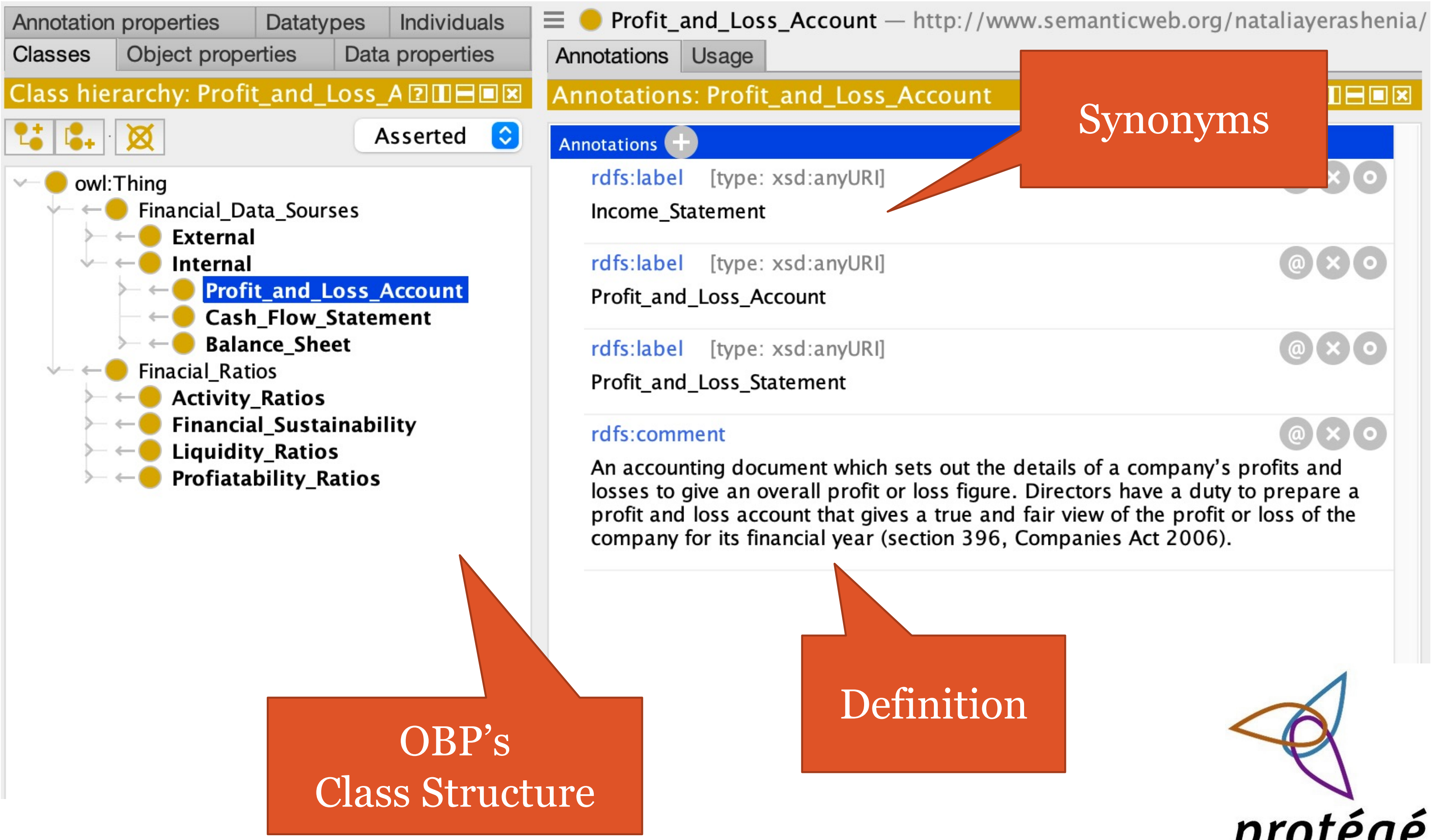



**Aim:** machine-readable and unambiguous data standard of the financial terminology in general

**Aim:** machine-readable and unambiguous data standard of the financial terminology on companies' financial analysis

# OBP Overview

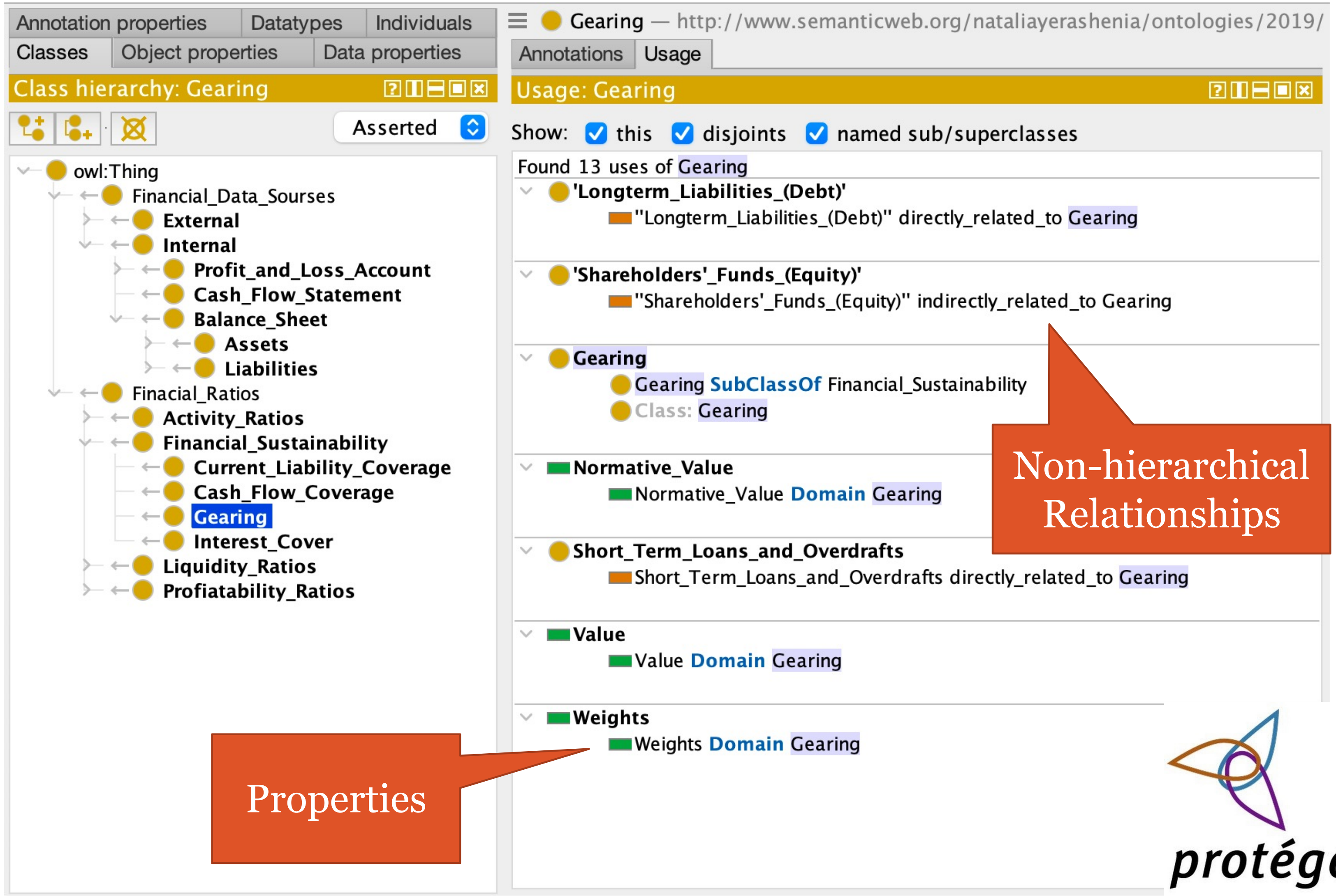
Formal Description of Concepts




protégé

# OBP Overview

## Concept's Relationships

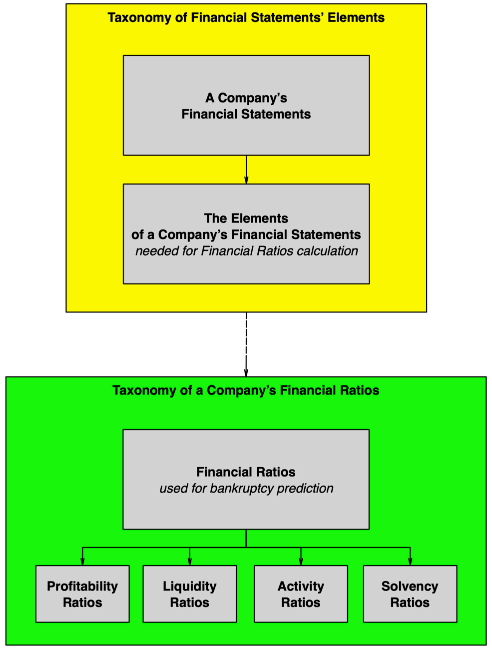



protégé

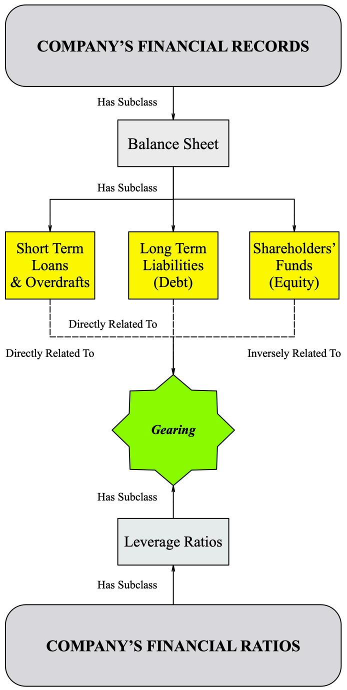
# Ontology of Bankruptcy Prediction as a core component of the system

## Steps of developing an OBP Ontology

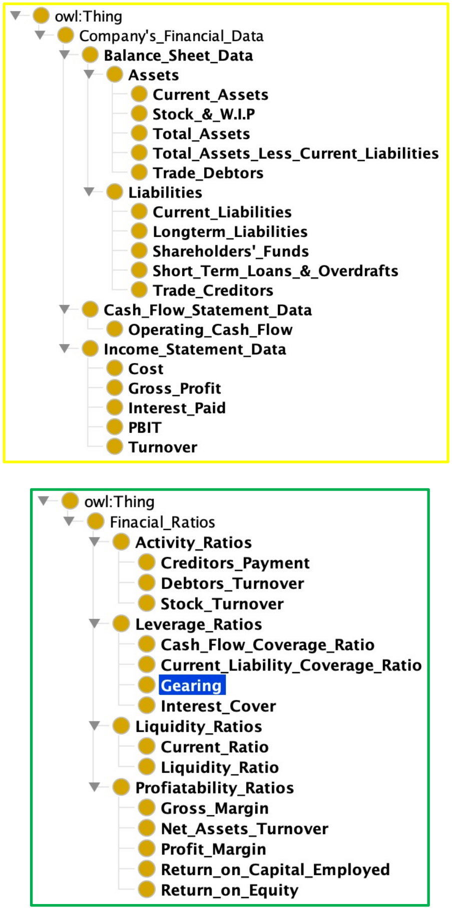
***Step 1.*** Creating a Conceptual Representation of OBP Ontology




***Example.*** Conceptual Representation of Gearing Ratio in OBP Ontology



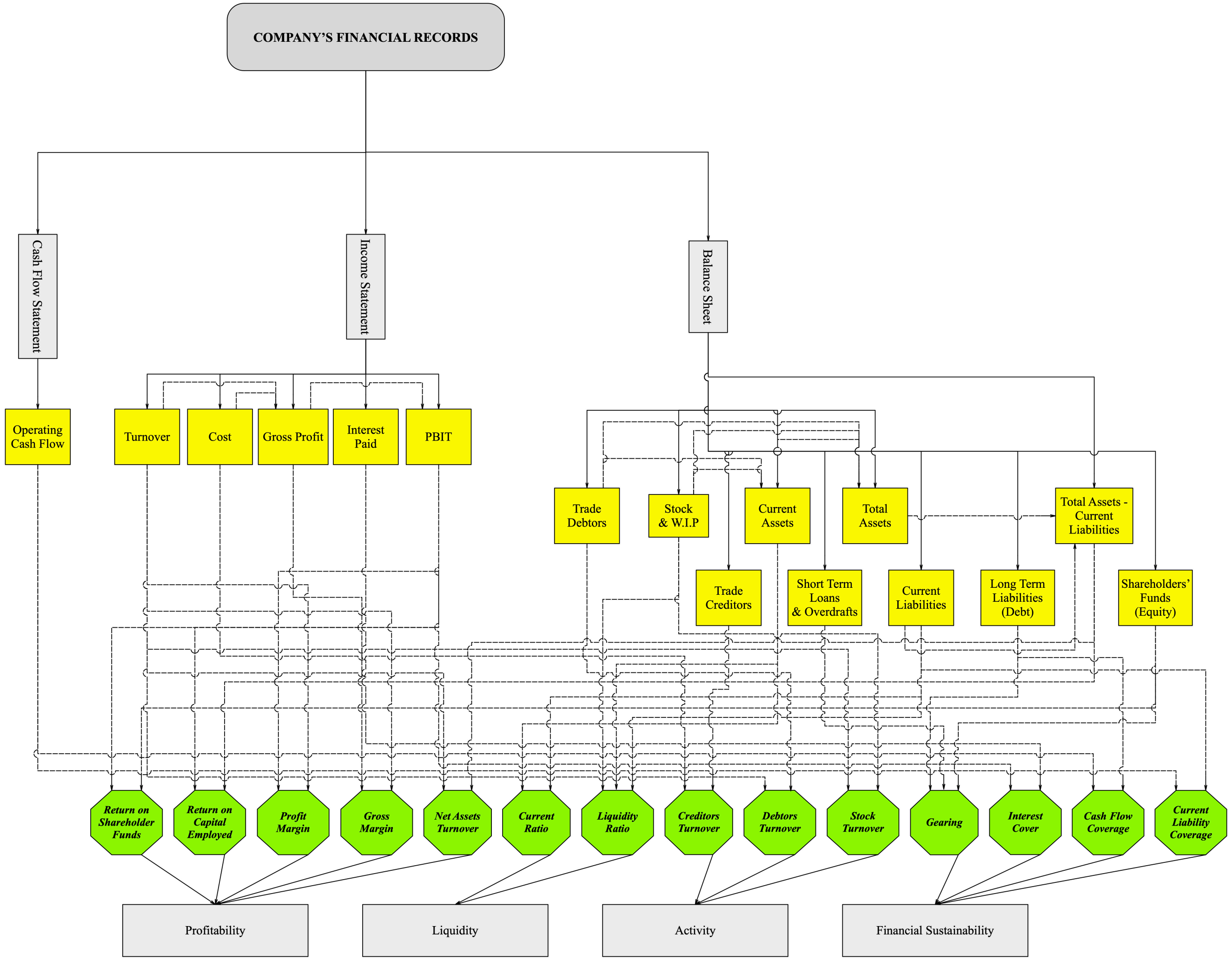

***Step 2.*** Creating Formal Physical Representation of OBP Ontology




*Note:* The composite Ontology of Bankruptcy Prediction content is based on the IFRS Standards and the UK Companies Act 2006

# Full version of Ontology of Bankruptcy Prediction

## Conceptual Representation




*Note:* This image shows only the objects of the OBP ontology which are involved in this particular Use Case

# Protégé as an environment to develop and edit OBP ontology

* In this project Protégé was used to create OWL files only, so in the long run, it doesn't matter what ontology development tool is used.

The ontology editor can be used to create a new ontology framework as the system evolves and adapt it to the new requirements.

Both new ontology made from scratch or third-party ready made RDF file can be used to in BPCM model

protégé

**Two files in Protégé:**

- 'BPCM ontology part 1.owl'
  Taxonomy of Financial Statements' Elements OWL file
- 'BPCM ontology part 2.owl'
  Taxonomy of Financial Ratios OWL file

# Data Files

## Training Data (45 UK Companies)

**'Training Data 2019 45.csv'** – CSV file containing financial ratios of 45 UK companies is used as training data for the Neural Network in Python.

**'Training_Output_2019_45.csv'** – CSV file containing information about the bankruptcy level of 45 UK companies is used as training data for the Neural Network in Python.

## Testing Company A

**'Company_A_Fin_Elements.csv'** – CSV file with financial data of Company A including the Elements of Cash Flow Statement, Income statement and Balance Sheet was taken from Fame Bureau Van Dijk database. The file is used as testing input data for Neo4j Graph.

**'Ratios_Export_NN.csv'** – CSV file automatically built in Neo4j graph database containing values of 14 financial ratios of Company A is used as testing input data for the Neural Network in Python.

# Steps of developing a Graph Database in Neo4j

neo4j

STEP 1

Importing the OBP Ontology OWL files made in Protégé into Neo4j

STEP 2

Creating non-hierarchical connections between the ‘Elements of a Company’s Financial Statements’ and ‘Financial Ratios’ taxonomies

STEP 3

Exporting the CSV file containing info about the company's financial records. Matching the node labels of “Elements of Financial Statements” taxonomy with the elements’ names in the CSV file.

STEP 4

The quantitative data from CSV file containing a company’s records is transferred to Neo4j as ”value” attributes of corresponded nodes.

STEP 5

Calculating the value attributes of the Ratios by using built-in math formulae

STEP 6

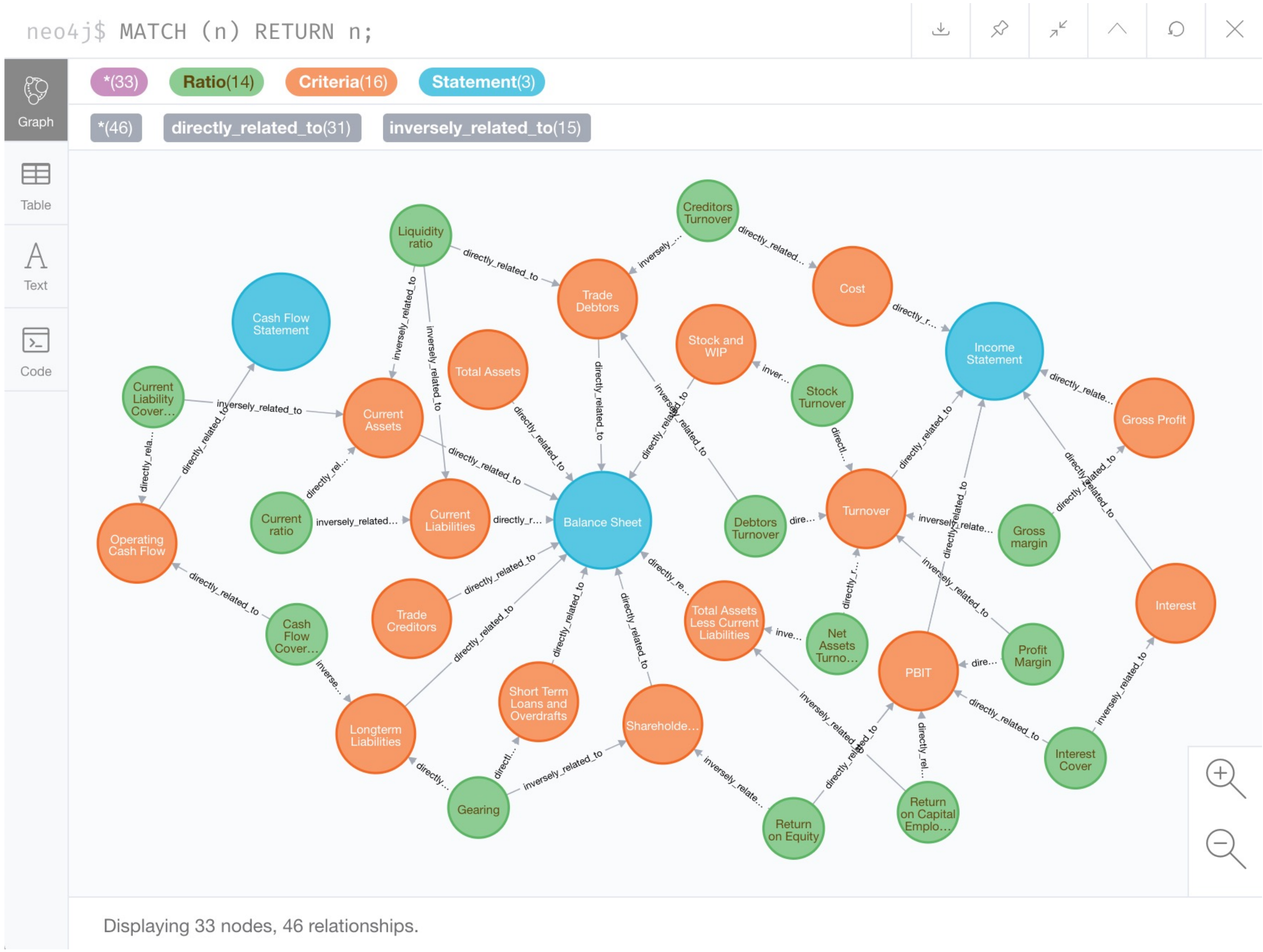
A company’s graph database is ready to be used for Feature Selection and analytical purposes (by means of Cypher queries)

STEP 7

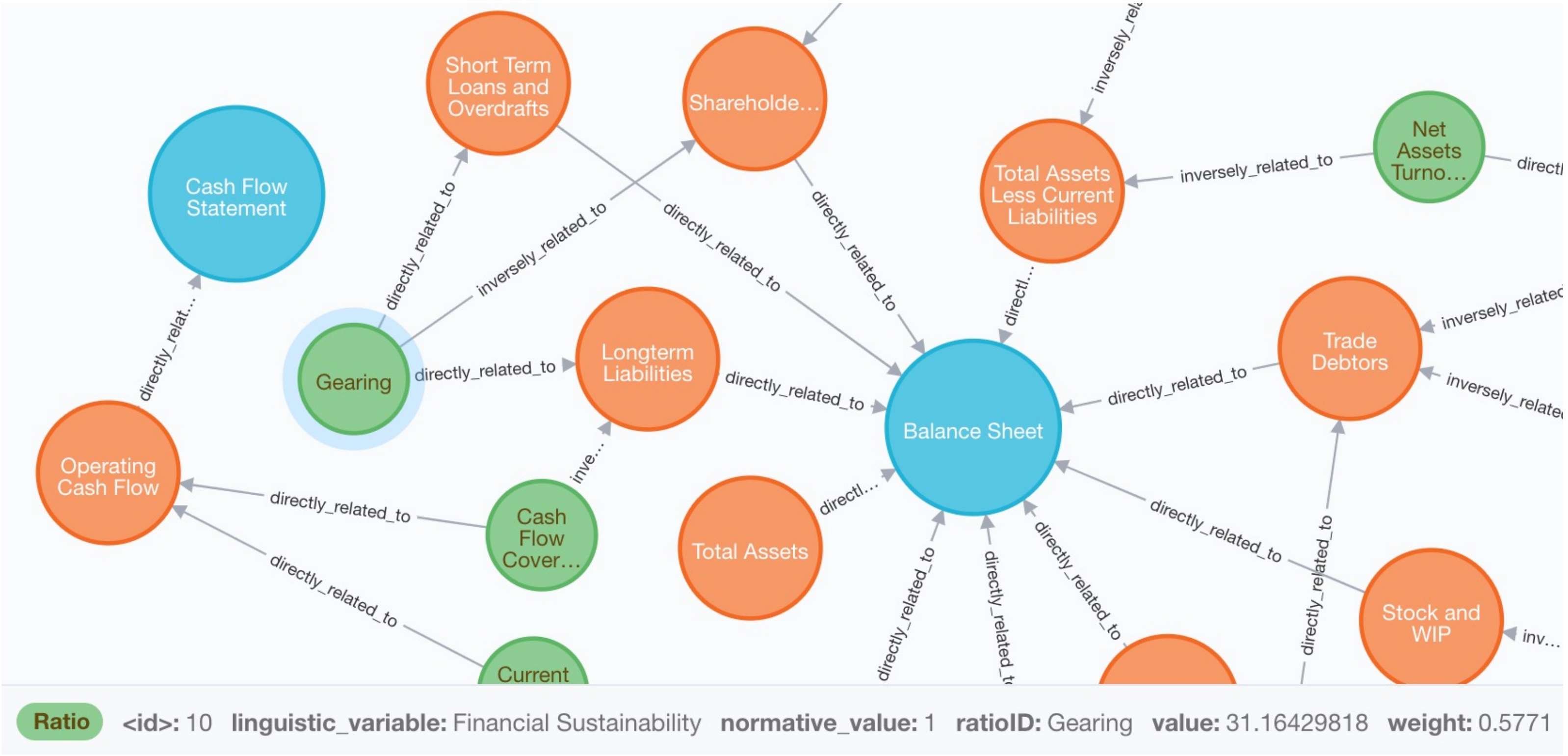
Exporting the values of the Ratios as CSV file

# OBP Ontology Representation In Neo4j Environment

- **33 nodes** divided into three categories: Ratio, Criteria (financial indicator), Statement
- **46 relationships** between them (of two types – direct and inverse)



# Part of a Neo4j representation graph with Gearing Ratio Components ontological path developed for the tested Company A



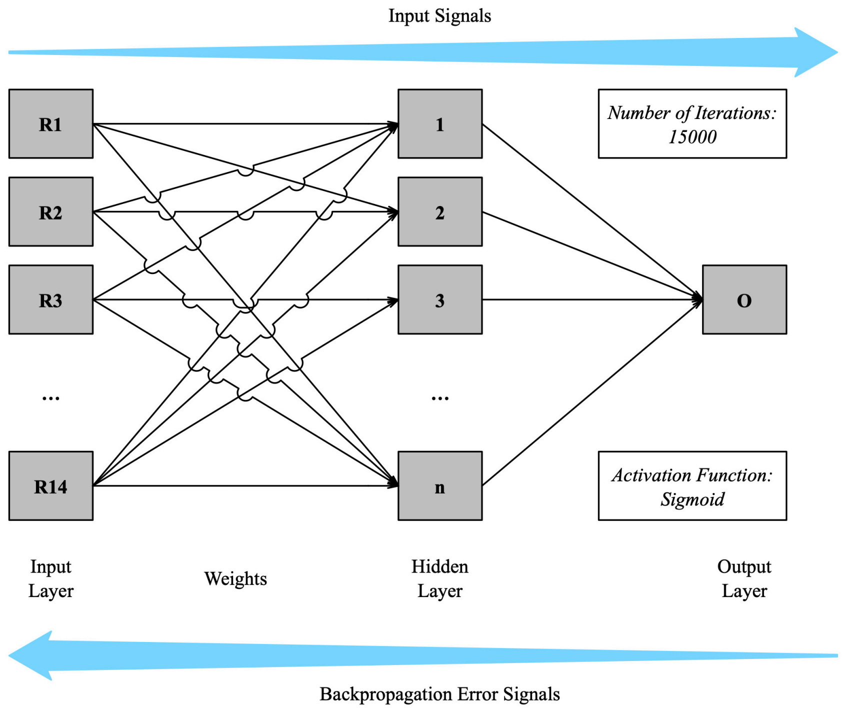

The value of Gearing Ratio for company A (31.16) is calculated using the in-built formula.

# Building Neural Network in Python using Semantic Database System Data

**Step 1.** Importing the files:

- Training Data
- Training Output Data
- Input Data (Exported from Neo4j)

**Step 2.** Designing and Training the Classical Neural Network for Bankruptcy Prediction

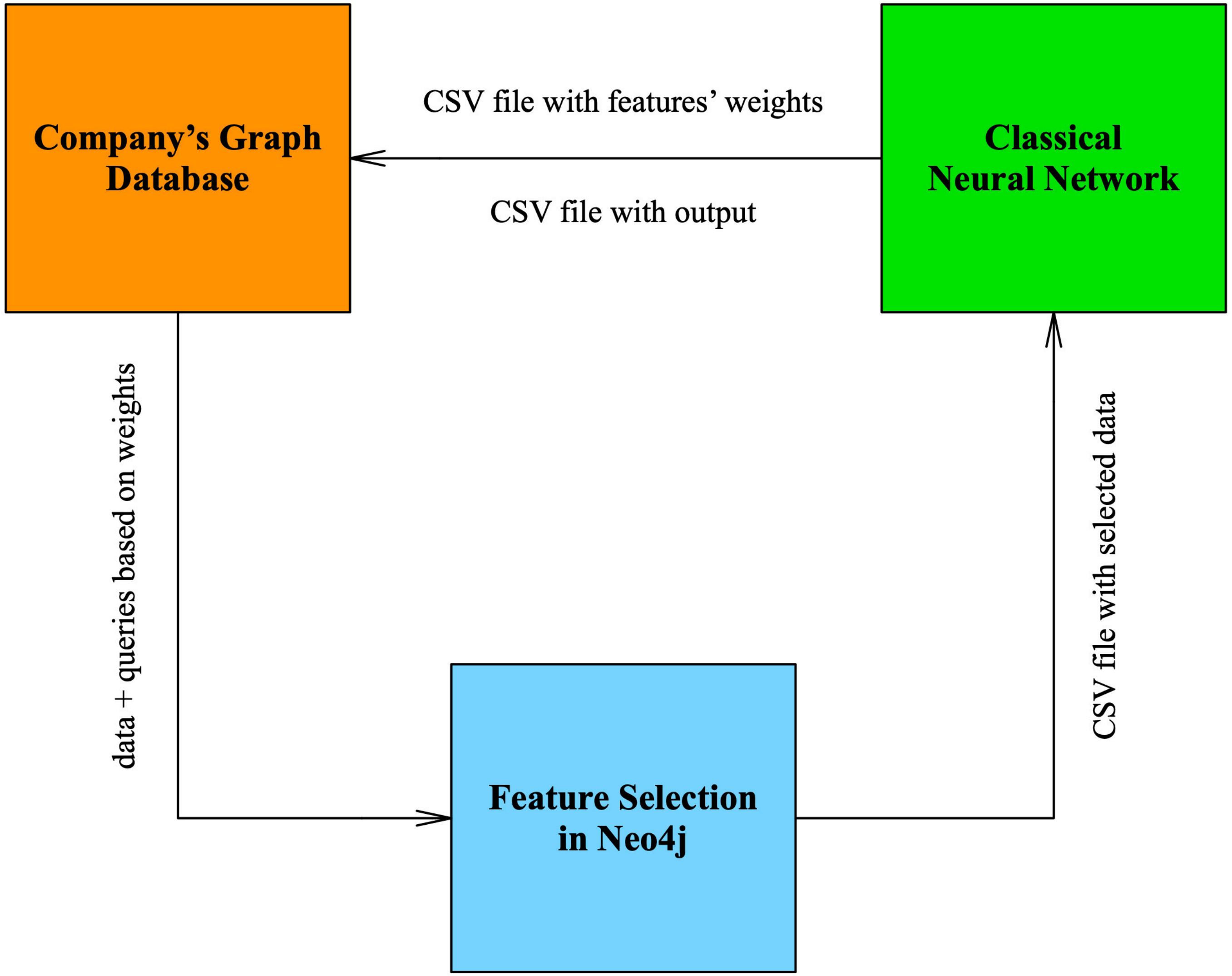



**Step 3.** Usage and diagnostics.

**Output:**
[1] – high risk of bankruptcy, [0] – the company is stable.

**Weights:**
Final *weights* of the analysed features ready to be transmitted to the Company's Graph Database

# Feedback Loop



# Accomplished to date & Contribution to knowledge

- A novel component-based architecture of **BPCM Model** was introduced, which integrates the Semantic Database System and a set of modern machine learning algorithms
- A conceptual representation of the **OBP Ontology** was designed and described
- A formal conceptual representation of the **OBP Ontology** through using Protégé environment
- The code of the **Graph DB** was built using *Cypher* query language and *Neo4j* environment
- A **Graph DB** for financial data of companies in *Neo4j* environment and input data values was implemented
- **Neural Network** in *Python* which uses the structured data from the Graph DB as its input was developed
- A **mechanism of transferring the data** between components was established: *Protégé -> Neo4j, Neo4j -> Python NN*

**Result:** The created methodology of developing a computational model based on semantic approach, with the embedded ontology and graph database. The methodology allows *to work with metadata* stored in the model. The generic nature allows *to upgrade existing and plug in new components on demand* (e.g. to embed different ontologies and datasets)

# Published Papers & Conferences to date

- Yerashenia, N., and Bolotov, A. ***Computational modelling for bankruptcy prediction: Semantic data analysis integrating graph database and financial ontology***. In 2019 IEEE 21st Conference on Business Informatics (CBI) (2019), vol. 1, IEEE, pp. 84–93.
- Yerashenia, N. and Bolotov, A., ***Creating an Intelligent System for Bankruptcy Detection: Semantic data Analysis Integrating Graph Database and Financial Ontology***. In Proceedings of the Automated Reasoning Workshop 2019: Bridging the Gap between Theory and Practice (2019), p. 11.
- Yerashenia, N., Bolotov, A., Pierantoni, G., and Chan, D. ***Semantic data pre-processing for machine learning based bankruptcy prediction computational model***. In 2020 IEEE 22nd Conference on Business Informatics (CBI) (2020), IEEE.

# Thank you!

**Natalia Yerashenia, Msc**

w1578366@my.westminster.ac.uk

**Dr Alexander Bolotov**

A.Bolotov@westminster.ac.uk

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

UNIVERSITY OF WESTMINSTER

115 NEW CAVENDISH STREET
LONDON W1W 6UW

UNITED KINGDOM