Healthcare Analytics - Load and analyse FDA Adverse Event Reporting System Data with Neo4j

*Emma Zhang, Ricky Wong, Xiaoxing Bian*

[**Problem 2**](#_hxr1crultf0v)

[**Data Model 3**](#_fc7dni1ztv53)

[Graph Nodes Explained 3](#_4ac7vft2mjx0)

[**Graph Creation 4**](#_sp93hdzl0gs)

[Preparing the schema 4](#_gvkmcs74ntzk)

[Load cases, manufacturers, and relate them 4](#_3aukwe2anmt3)

[Load other information related to the events 5](#_kzzintjozfpe)

[Load drugs and therapies 6](#_ocj79wgpy4c8)

[**Analysis 8**](#_m11ksyhsvq8h)

[Question 1: Which outcomes are most "influential" or central in the network of cases involving Myocardial Infarction? 8](#_o7z74ennamyu)

[Question 2: What demographic factors (e.g., age group, gender) are most commonly associated with cardiovascular-related drug reactions? 9](#_yv0wlou7e0h)

[Question 3: What are the top 10 drugs reported with cardiovascular-related side effects? 11](#_ik9z3ngd52oe)

[Question 4: What are the most common drug-drug interactions associated with cardiovascular-related side effects, reported in the FAERS dataset? 12](#_sj7vny7rlhxl)

[Question 5: How can the company predict the likelihood of cardiovascular-related adverse reactions? 13](#_w8e9q0b8nzzg)

[Question 6: How can the company accurately predict and classify the severity of adverse events (outcomes) reported to guide strategic decisions in drug development? 17](#_enlwh38yjg82)

[**Conclusion 21**](#_l2fq5a6ykgp4)

[**References 22**](#_ndfj35t4z20i)

Remark: Please note that all references to PharmaCARE, including its operations, research activities, and associated data analyses, are **purely fictional** and created for **illustrative purposes** only. The scenarios, data, and outcomes presented in this document are hypothetical and not based on real events or entities. Any resemblance to actual companies, operations, or results is purely coincidental. This fictional setup is intended to demonstrate the potential application of Neo4j graph databases and machine learning in pharmaceutical analytics.

# Problem

PharmaCARE, a pharmaceutical company, prides itself on its commitment to patient safety and innovation in drug development. However, like many companies in the industry, PharmaCARE faces the continuous challenge of monitoring and responding to adverse events associated with its products, especially with a growing portfolio of drugs on the market. The company seeks a comprehensive analysis to support strategic decisions on the next drug to develop. Based on previous research, the company decided to focus on drug development related to cardiovascular diseases, which remain a leading cause of mortality and morbidity worldwide.

To achieve this goal, the company decided to analyze the FDA Adverse Event Reporting System (FAERS) dataset. This extensive dataset contains reports of adverse events and medication errors from across the United States. The FDA uses FAERS to monitor new adverse events and medication errors that might occur with these products. It is a system that measures occasional harms from medications to ascertain whether the risk-benefit ratio is high enough to justify continued use of any drug and to identify correctable and preventable problems in healthcare delivery (such as the need for retraining to prevent prescribing errors).

This dataset offers invaluable insights that can inform the entire drug development process. By analyzing FAERS data, PharmaCARE aims to identify patterns and potential risks associated with existing treatments, thereby guiding the development of a safer and more effective cardiovascular drug. The comprehensive report on the FAERS data, utilizing Neo4j for graph-based analytics, will provide PharmaCARE with the insights needed to inform and guide its drug development strategies, with a particular focus on addressing cardiovascular-related issues and adverse events.

# Data Model

#### Fig. Data model of the database.

## Graph Nodes Explained

|  |  |  |
| --- | --- | --- |
| Entity | Label | Description |
| Demographic | Case | This is the demographic information of a person involved in the adverse event report. |
| Drug | Drug | Drug involved in the adverse event. A drug can be a primary suspect, secondary suspect, concomitant or interacting drug responsible for the adverse effect. This suspect type is identified by the relationship between Case and Drug Nodes. |
| Reaction | Reaction | This is the reaction that the person (Case) developed after consumption of the respective drug, like 'Pain', 'Body temperature increased' or 'Insomnia' or 'Memory Loss' etc. |
| Outcome | Outcome | This is the long term outcome of the case after the adverse event, for example 'Hospitalization: Initial or Prolonged', 'Disability' or 'Death' |
| Report\_Source | ReportSource | This is the reported of the adverse event, for example 'Health Professional', 'Consumer', 'User Health Facility' etc who has reported the event to FDA system. |
| Therapy | Therapy | For some cases, they receive drug as a part of a therapy. This is the therapy details for the case. |
| Indication | - | This is the medical term for drug indication that has the details like drug sequence, indication point. We have not used a separate Node label for this, instead we have covered required details as Case to Drug relationship properties. |
| Demographics | AgeGroup | Demographics table in the FEARS data has age information that we turned into a separate node for Age Group reporting. |

# Graph Creation

## Preparing the schema

##### // Constraints

##### CREATE CONSTRAINT constraint\_drug\_name IF NOT EXISTS FOR (n: `Drug`) REQUIRE n.`name` IS UNIQUE;

##### CREATE CONSTRAINT constraint\_case\_primaryid IF NOT EXISTS FOR (n: `Case`) REQUIRE n.`primaryid` IS UNIQUE;

##### CREATE CONSTRAINT constraint\_reaction\_description IF NOT EXISTS FOR (n: `Reaction`) REQUIRE n.`description` IS UNIQUE;

##### CREATE CONSTRAINT constraint\_reportsource\_code IF NOT EXISTS FOR (n: `ReportSource`) REQUIRE n.`code` IS UNIQUE;

##### CREATE CONSTRAINT constraint\_outcome\_code IF NOT EXISTS FOR (n: `Outcome`) REQUIRE n.`code` IS UNIQUE;

##### CREATE CONSTRAINT constraint\_therapy\_primaryid IF NOT EXISTS FOR (n: `Therapy`) REQUIRE n.`primaryid` IS UNIQUE;

##### CREATE CONSTRAINT constraint\_manufacturer\_name IF NOT EXISTS FOR (n: `Manufacturer`) REQUIRE n.`manufacturerName` IS UNIQUE;

##### 

##### // indexes

##### CREATE INDEX index\_case\_age IF NOT EXISTS FOR (n: `Case`) ON (n.`age`);

##### CREATE INDEX index\_case\_ageUnit IF NOT EXISTS FOR (n: `Case`) ON (n.`ageUnit`);

##### CREATE INDEX index\_case\_gender IF NOT EXISTS FOR (n: `Case`) ON (n.`gender`);

##### CREATE INDEX index\_case\_eventdate IF NOT EXISTS FOR (n: `Case`) ON (n.`eventDate`);

##### CREATE INDEX index\_case\_reportdate IF NOT EXISTS FOR (n: `Case`) ON (n.`reportDate`);

## Load cases, manufacturers, and relate them

##### LOAD CSV WITH HEADERS FROM "https://raw.githubusercontent.com/neo4j-graph-examples/healthcare-analytics/main/data/csv/demographics.csv" AS row

##### 

##### //Conditionally create Case nodes, set properties on first create

##### MERGE (c:Case { primaryid: toInteger(row.primaryid) })

##### ON CREATE SET

##### c.eventDate= date(row.eventDate),

##### c.reportDate= date(row.reportDate),

##### c.age = toFloat(row.age),

##### c.ageUnit = row.ageUnit,

##### c.gender = row.sex,

##### c.reporterOccupation = row.reporterOccupation

##### 

##### //Conditionally create Manufacturer

##### MERGE (m:Manufacturer { manufacturerName: row.manufacturerName } )

##### 

##### //Relate case and manufacturer

##### MERGE (m)-[:REGISTERED]->(c)

##### 

##### //Conditionally create age group node and relate with case

##### MERGE (a:AgeGroup { ageGroup: row.ageGroup })

##### 

##### //Relate case with age group

##### MERGE (c)-[:FALLS\_UNDER]->(a)

##### 

##### RETURN count (c);

## Load other information related to the events

### Load outcomes and link them with cases

##### LOAD CSV WITH HEADERS FROM "https://raw.githubusercontent.com/neo4j-graph-examples/healthcare-analytics/main/data/csv/outcome.csv" AS row

##### 

##### // Conditionally create outcome node

##### MERGE (o:Outcome { code: row.code })

##### ON CREATE SET

##### o.outcome = row.outcome

##### 

##### WITH o, row

##### 

##### // Find the case to relate this outcome to

##### MATCH (c:Case {primaryid: toInteger(row.primaryid)})

##### 

##### // Relate

##### MERGE (c)-[:RESULTED\_IN]->(o)

##### 

##### RETURN count(o);

### Load reactions and link them with cases

##### LOAD CSV WITH HEADERS FROM "https://raw.githubusercontent.com/neo4j-graph-examples/healthcare-analytics/main/data/csv/reaction.csv" AS row

##### 

##### //Conditionally create reaction node

##### MERGE (r:Reaction { description: row.description })

##### 

##### WITH r, row

##### 

##### //Find the case to relate this reaction to

##### MATCH (c:Case {primaryid: toInteger(row.primaryid)})

##### 

##### //Relate

##### MERGE (c)-[:HAS\_REACTION]->(r)

##### 

##### RETURN count(r);

### Load report sources and link them with cases

##### LOAD CSV WITH HEADERS FROM "https://raw.githubusercontent.com/neo4j-graph-examples/healthcare-analytics/main/data/csv/reportSources.csv" AS row

##### 

##### // Conditionally create reportSource node

##### MERGE (r:ReportSource { code: row.code })

##### ON CREATE SET

##### r.name = row.name

##### 

##### WITH r, row

##### 

##### // Find the case to relate this report source to

##### MATCH (c:Case {primaryid: toInteger(row.primaryid) })

##### 

##### WITH c, r

##### 

##### // Relate

##### MERGE (c)-[:REPORTED\_BY]->(r)

##### 

##### RETURN count(r);

## Load drugs and therapies

### Load drugs with indications and link them with cases using relationships based on their roles for the cases

##### :auto LOAD CSV WITH HEADERS FROM "https://raw.githubusercontent.com/neo4j-graph-examples/healthcare-analytics/main/data/csv/drugs-indication.csv" AS row

##### 

##### CALL { WITH row

##### //Conditionally create Drug node

##### MERGE (d:Drug { name: row.name })

##### ON CREATE SET

##### d.primarySubstabce = row.primarySubstabce

##### 

##### WITH d, row

##### 

##### //Find the case to relate this drug based on the suspect type

##### MATCH (c:Case {primaryid: toInteger(row.primaryid)})

##### 

##### FOREACH (\_ IN CASE WHEN row.role = "Primary Suspect" THEN [1] ELSE [] END |

##### //Relate

##### MERGE (c)-[relate:IS\_PRIMARY\_SUSPECT { drugSequence: row.drugSequence, route: row.route, doseAmount: row.doseAmount, doseUnit: row.doseUnit, indication: row.indication }]->(d)

##### )

##### 

##### FOREACH (\_ IN CASE WHEN row.role = "Secondary Suspect" THEN [1] ELSE [] END |

##### //Relate

##### MERGE (c)-[relate:IS\_SECONDARY\_SUSPECT { drugSequence: row.drugSequence, route: row.route, doseAmount: row.doseAmount, doseUnit: row.doseUnit, indication: row.indication }]->(d)

##### )

##### 

##### FOREACH (\_ IN CASE WHEN row.role = "Concomitant" THEN [1] ELSE [] END |

##### //Relate

##### MERGE (c)-[relate:IS\_CONCOMITANT { drugSequence: row.drugSequence, route: row.route, doseAmount: row.doseAmount, doseUnit: row.doseUnit, indication: row.indication }]->(d)

##### )

##### 

##### FOREACH (\_ IN CASE WHEN row.role = "Interacting" THEN [1] ELSE [] END |

##### //Relate

##### MERGE (c)-[relate:IS\_INTERACTING { drugSequence: row.drugSequence, route: row.route, doseAmount: row.doseAmount, doseUnit: row.doseUnit, indication: row.indication }]->(d)

##### )

##### } IN TRANSACTIONS OF 5000 ROWS;

### Load therapies and link them with cases and drugs

##### LOAD CSV WITH HEADERS FROM "https://raw.githubusercontent.com/neo4j-graph-examples/healthcare-analytics/main/data/csv/therapy.csv" AS row

##### 

##### //Conditionally create therapy node

##### MERGE (t:Therapy { primaryid: toInteger(row.primaryid) })

##### 

##### WITH t, row

##### 

##### //Find the case to relate this therapy to

##### MATCH (c:Case {primaryid: toInteger(row.primaryid)})

##### 

##### //Relate case with therapy

##### MERGE (c)-[:RECEIVED]->(t)

##### 

##### WITH c, t, row

##### 

##### //Find drugs prescribed in the therapy

##### MATCH (d:Drug { name: row.drugName })

##### 

##### //Relate therapy and drugs

##### MERGE (t)-[:PRESCRIBED { drugSequence: row.drugSequence, startYear: coalesce(row.startYear, 1900), endYear: coalesce(row.endYear, 2021) } ]->(d);

# 

# Analysis

## Question 1: Which outcomes are most "influential" or central in the network of cases involving Myocardial Infarction?

To answer this question, a cypher query is used to project a graph from the existing data, focusing on cases, outcomes, and reactions related to Myocardial Infarction. Then PageRank algorithm is run to identify the most central or influential nodes in this network. Finally, the query retrieves and presents these nodes in a resultset, highlighting the key outcomes or reactions associated with Myocardial Infarction cases.

##### CALL gds.graph.project(

##### 'miOutcomeGraph',

##### ['Case', 'Outcome', 'Reaction'],

##### { resultedIn: {

##### type: 'RESULTED\_IN',

##### orientation: 'UNDIRECTED'

##### },

##### hasReaction: { type: 'HAS\_REACTION',

##### orientation: 'UNDIRECTED'

##### }

##### }

##### );

##### CALL gds.pageRank.stream('miOutcomeGraph')

##### YIELD nodeId, score

##### MATCH (n) WHERE id(n) = nodeId

##### RETURN

##### CASE

##### WHEN n.outcome IS NOT NULL THEN n.outcome

##### WHEN n.description IS NOT NULL THEN n.description

##### ELSE 'Unknown'

##### END AS OutcomeOrReaction,

##### score

##### ORDER BY score DESC

##### LIMIT 10;

#### Fig. Result set of the query.

High PageRank Scores indicate outcomes that are central and frequently connected within the network of Myocardial Infarction cases. These are likely to be the most critical outcomes to monitor or investigate further. Lower PageRank Scores, on the other hand, Indicate outcomes or reactions that are still important but perhaps occur less frequently or are less interconnected within the network. From the result set, the outcomes with the highest scores are “Other Serious (Important Medical Event)”, "Hospitalization - Initial or Prolonged", and "Death". These outcomes are the ones that are most frequently connected and therefore could be considered the most influential in the context of myocardial infarction cases. Therefore, these results can help the company prioritize areas for further analysis or intervention, focusing on the most central outcomes and reactions that are heavily associated with Myocardial Infarction cases.

## Question 2: What demographic factors (e.g., age group, gender) are most commonly associated with cardiovascular-related drug reactions?

Next, we want to identify demographic factors, specifically age group and gender, that are most commonly associated with cardiovascular-related drug reactions.

##### MATCH (c:Case)-[:FALLS\_UNDER]->(a:AgeGroup),

##### (c)-[:HAS\_REACTION]->(r:Reaction),

##### (c)-[:IS\_PRIMARY\_SUSPECT]->(d:Drug)

##### WHERE r.description IN ["Myocardial infarction", "Myocardial ischaemia", "Acute myocardial infarction", "Coronary artery disease", "Coronary artery stenosis", "Coronary arterial stent insertion", "Coronary angioplasty", "Atrial fibrillation", "Angina pectoris", "Deep vein thrombosis", "Thrombosis", "Pulmonary embolism", "Pulmonary oedema", "Hypotension", "Hypertension", "Aortic stenosis", "Atrioventricular block second degree", "Arrhythmia", "Cardiac arrest", "Cardiac failure", "Cardiac failure congestive", "Cardiac tamponade", "Cardiomyopathy", "Pericardial effusion", "Pleural effusion", "Electrocardiogram QT prolonged", "Electrocardiogram repolarisation abnormality", "Endocarditis bacterial", "Haemorrhage", "Lacunar stroke", "Ischaemic stroke", "Cerebrovascular accident", "Ventricular extrasystoles", "Haemodynamic instability"]

##### RETURN a.ageGroup AS AgeGroup, c.gender AS Gender, COUNT(c) AS CaseCount

##### ORDER BY CaseCount DESC;

The query highlights the demographic groups that are most frequently associated with serious cardiovascular drug reactions. We gathered the cardiovascular-related conditions to put them in the WHERE clause to ensure that the analysis is specific to cardiovascular-related conditions, making the results relevant to understanding which demographics are at higher risk for these types of reactions.

#### Fig. Result set of the cypher query.

The result set highlights that elderly females represent the highest number of cases associated with cardiovascular-related drug reactions. This suggests that elderly women are the demographic most frequently experiencing these types of reactions, potentially indicating a higher risk or more frequent exposure to cardiovascular-related issues in this group. In general, the elderly are more prone to cardiovascular-related conditions, compared with adults and adolescents.

## Question 3: What are the top 10 drugs reported with cardiovascular-related side effects?

By identifying drugs most frequently associated with cardiovascular complications, PharmaCARE can take proactive steps to improve drug formulations, adjust dosages, or provide clearer usage guidelines to reduce the risk of adverse events. The following cypher query is used to retrieved the required information:

##### MATCH (c:Case)-[:IS\_PRIMARY\_SUSPECT]->(d:Drug)

##### MATCH (c)-[:HAS\_REACTION]->(r:Reaction)

##### WHERE r.description IN [

##### "Myocardial infarction", "Myocardial ischaemia", "Acute myocardial infarction",

##### "Coronary artery disease", "Coronary artery stenosis", "Coronary arterial stent insertion", "Coronary angioplasty", "Atrial fibrillation", "Angina pectoris", "Deep vein thrombosis","Thrombosis", "Pulmonary embolism", "Pulmonary oedema", "Hypotension", "Hypertension", "Aortic stenosis", "Atrioventricular block second degree", "Arrhythmia", "Cardiac arrest", "Cardiac failure", "Cardiac failure congestive", "Cardiac tamponade", "Cardiomyopathy", "Pericardial effusion", "Pleural effusion", "Electrocardiogram QT prolonged", "Electrocardiogram repolarisation abnormality", "Endocarditis bacterial", "Haemorrhage", "Lacunar stroke", "Ischaemic stroke", "Cerebrovascular accident", "Ventricular extrasystoles", "Haemodynamic instability"]

##### WITH d.name as drugName, collect(r.description) as sideEffects, count(r.description) as totalSideEffects

##### RETURN drugName, sideEffects[0..5] as sideEffects, totalSideEffects

##### ORDER BY totalSideEffects DESC LIMIT 10;

The table below presents the top 10 drugs reported with cardiovascular-related side effects, ranked by the total number of adverse events recorded in the dataset.

#### Fig. The top 10 drugs reported with cardiovascular-related side effects.

Among the top 10 drugs most frequently reported with such side effects, REVLIMID stands out with a notably high number of reports (56), predominantly linked to thrombotic events. Other drugs, such as IMBRUVICA and ATEZOLIZUMAB, also show a considerable association with multiple cardiovascular side effects, including myocardial infarction, pleural effusion, and hypotension.

## Question 4: What are the most common drug-drug interactions associated with cardiovascular-related side effects, reported in the FAERS dataset?

Understanding the interactions between drugs that contribute to these side effects is vital for PharmaCARE to develop new drugs or update existing ones to reduce the risks associated with cardiovascular-related side effects. This analysis focuses on identifying the most common drug-drug interactions associated with cardiovascular-related side effects reported in the FAERS dataset. The ***Common Neighbors algorithm*** was employed to identify pairs of drugs that frequently co-occurred in these cases. This algorithm calculates a score representing the number of shared cases between two drugs, indicating potential interaction significance.

Step 1: Create a drug-interaction relationship associated with cardiovascular-related side effects:

##### MATCH (c:Case)-[:HAS\_REACTION]->(r:Reaction)

##### WHERE r.description IN ["Myocardial infarction", "Myocardial ischaemia", "Acute myocardial infarction","Coronary artery disease", "Coronary artery stenosis", "Coronary arterial stent insertion", “Coronary angioplasty", "Atrial fibrillation", "Angina pectoris", "Deep vein thrombosis", "Thrombosis", "Pulmonary embolism", "Pulmonary oedema", "Hypotension", "Hypertension", "Aortic stenosis", "Atrioventricular block second degree", "Arrhythmia", "Cardiac arrest", "Cardiac failure", "Cardiac failure congestive", "Cardiac tamponade", "Cardiomyopathy", "Pericardial effusion", "Pleural effusion", "Electrocardiogram QT prolonged", "Electrocardiogram repolarisation abnormality", "Endocarditis bacterial", "Haemorrhage", "Lacunar stroke", "Ischaemic stroke", "Cerebrovascular accident", "Ventricular extrasystoles", "Haemodynamic instability"]

##### MATCH (c)-[:IS\_PRIMARY\_SUSPECT|IS\_SECONDARY\_SUSPECT|IS\_CONCOMITANT|IS\_INTERACTING]->(d1:Drug)

##### MATCH (c)-[:IS\_PRIMARY\_SUSPECT|IS\_SECONDARY\_SUSPECT|IS\_CONCOMITANT|IS\_INTERACTING]->(d2:Drug)

##### WHERE d1 <> d2 AND d1.name < d2.name

##### MERGE (d1)-[i:DRUG\_INTERACTION]->(d2)

##### ON CREATE SET i.weight = 1

##### ON MATCH SET i.weight = i.weight + 1;

Step 2: Using graph algorithm ‘gds.alpha.linkprediction.commonNeighbors’ to calculate the Common Neighbors score between the two drug nodes, which reflects how many cases they are commonly associated with.

##### MATCH (d1:Drug)-[:DRUG\_INTERACTION]-(d2:Drug)

##### Where d1.name < d2.name

##### WITH d1, d2

##### RETURN d1.name AS Drug1, d2.name AS Drug2,

##### gds.alpha.linkprediction.commonNeighbors(d1, d2) AS score

##### ORDER BY score DESC

##### LIMIT 10;

#### Fig. The top 10 drug pairs, ranked by their common neighbors' scores

The results highlight specific drug combinations, such as Aspirin with Prednisone and Furosemide with Omeprazole, that are frequently implicated in adverse cardiovascular events. By focusing on the interactions identified in this study, PharamACARE can enhance drug safety profiles, refine clinical trial designs, and develop targeted strategies for mitigating adverse effects.

## Question 5: How can the company predict the likelihood of cardiovascular-related adverse reactions?

To strengthen its commitment to patient safety, PharamACARE developed an advanced link prediction model leveraging FAERS data. This model is specifically designed to predict the likelihood of cardiovascular-related adverse reactions, such as myocardial infarction, atrial fibrillation, and pulmonary embolism, among others. By identifying potential high-risk cases, the model aids in preemptive decision-making during drug development.

The model was developed using a RandomForest algorithm, trained within Neo4j’s Graph Data Science (GDS) Library. It incorporates multiple features as the specific drugs prescribed, the patient’s age group, and gender to accurately predict the likelihood of adverse reactions.

To train the model, we need to process the data. The first step is to encode the categorical variables and features in the nodes.

##### MATCH (c:Case)

##### WHERE c.gender = 'M'

##### SET c.genderEncoded = 0;

##### 

##### MATCH (c:Case)

##### WHERE c.gender = 'F'

##### SET c.genderEncoded = 1;

##### 

##### MATCH (d:Drug)

##### WITH d, id(d) AS drugId

##### SET d.drugEncoded = drugId;

##### 

##### MATCH (r:Reaction)

##### WITH r, id(r) AS reactionId

##### SET r.reactionEncoded = reactionId;

##### 

##### MATCH (a:AgeGroup)

##### WITH a, id(a) AS ageGroupId

##### SET a.ageGroupEncoded = ageGroupId;

##### 

##### MATCH (c:Case)-[:FALLS\_UNDER]->(a:AgeGroup)

##### SET c.ageGroupEncoded = a.ageGroupEncoded;

Each variable is encoded with a unique id or index.

Next, the encoded data is projected into an in-memory graph using Neo4j’s gds.graph.project method. A link prediction pipeline is created with several steps of configurations - Node Property Steps where two embeddings. The dataset is split into training, testing and validation sets using 3-fold cross-validation.

##### CALL gds.graph.project(

##### 'myGraph3',

##### {

##### Case: {

##### properties: ['genderEncoded', 'ageGroupEncoded']

##### },

##### Drug: {

##### properties: ['drugEncoded']

##### },

##### Therapy: {},

##### Reaction: {

##### properties: ['reactionEncoded']

##### }

##### },

##### {

##### HAS\_REACTION: {

##### orientation: 'UNDIRECTED'

##### },

##### FALLS\_UNDER: {

##### orientation: 'UNDIRECTED'

##### },

##### PRESCRIBED: {

##### orientation: 'UNDIRECTED'

##### },

##### RECEIVED: {

##### orientation: 'UNDIRECTED'

##### }

##### }

##### )

##### YIELD graphName, nodeCount, relationshipCount;

##### 

##### CALL gds.beta.pipeline.linkPrediction.create('caseReactionPipeline3')

##### YIELD name;

##### 

##### CALL gds.beta.pipeline.linkPrediction.addNodeProperty('caseReactionPipeline3', 'fastRP', {

##### mutateProperty: 'caseEmbedding',

##### embeddingDimension: 128,

##### randomSeed: 42

##### })

##### YIELD name, nodePropertySteps;

##### 

##### CALL gds.beta.pipeline.linkPrediction.addNodeProperty('caseReactionPipeline3', 'node2vec', {

##### mutateProperty: 'drugEmbedding',

##### embeddingDimension: 64,

##### randomSeed: 42

##### })

##### YIELD name, nodePropertySteps;

##### 

##### CALL gds.beta.pipeline.linkPrediction.addFeature('caseReactionPipeline3', 'hadamard', {

##### nodeProperties: ['caseEmbedding', 'drugEmbedding']

##### })

##### YIELD featureSteps;

##### 

##### CALL gds.beta.pipeline.linkPrediction.configureSplit('caseReactionPipeline3', {

##### testFraction: 0.25,

##### trainFraction: 0.6,

##### validationFolds: 3

##### })

##### YIELD splitConfig;

The pipeline is configured to use a RandomForest model with 50 decision trees. The model is trained to predict the “HAS\_REACTION” relationship between “CASE” and “REACTION” nodes, using the configured embeddings and features. The model’s performance is evaluated using metrics such as AUCPR (Area Under Precision-Recall Curve) and OUT\_OF\_BAG\_ERROR.

After training, the model produced an overall train score of 0.9257 and a test score of 0.70. The model’s performance is moderately strong predicting the reactions in new cases.

#### Fig. The metrics and scores of the trained model.

The model is able to predict the reaction of cases with probability. This model serves as a strong foundation for the company to control the risk of patients suffering from cardiovascular-related adverse events.

#### Fig. Example of predictions of reactions on cases with probability.

## Question 6: How can the company accurately predict and classify the severity of adverse events (outcomes) reported to guide strategic decisions in drug development?

In addition to predicting the likelihood of a certain reaction happening in a patient, PharmaCARE also developed a tri-class classification model leveraging the FAERS data. This model is designed to predict and classify adverse events a reported case could be suffering from. In the model, there are three distinct categories:

* severe outcomes which include life-threatening conditions and death,
* other outcomes where these were less severe but still significant adverse events,
* no outcome where cases where no adverse event was reported following drug usage.

The model was developed using a RandomForestClassifier, trained with Apache Spark MLlib, to handle the extensive FAERS data, which consists of 11,381 nodes, 61,453 relationships, and more than 10 relationship types. Given the large number of features in the dataset, careful feature selection was crucial. The key features identified for predicting outcomes include the specific drugs the patient is taking, along with the patient's age group and gender.

Data extraction for the model was achieved using a Cypher query, which pulled relevant information on each case, including drug names, reaction types, outcomes, and demographic details. The result sets are converted into PySpark DataFrame.

#### Fig. Cypher query used to extract relevant information from the database.

After the data exploration, however, the dataset presented a significant class imbalance, with the majority of instances falling into the "no outcome" category (113103 records of class 0, 14323 records of class 1 and 11935 records of class 2).

#### Fig. Significant class imbalance of the data.

Model trained with such imbalanced data, though with high accuracy, will suffer from low recall and precision.

#### Fig. Model evaluation on test set with Imbalanced dataset.

To address the problem of imbalance data, oversampling techniques were applied to the minority classes (other outcomes and severe outcomes) to match the size of the majority class.

#### Fig. The size of data in each class after rebalancing

##### indexers = [

##### StringIndexer(inputCol=col, outputCol=col+"\_index").fit(df)

##### for col in ["drug\_name", "suspect\_role", "reaction\_type", "age\_group", "gender", "tri\_class\_outcome"]

##### ]

##### 

##### pipeline = Pipeline(stages=indexers)

##### df\_prepared = pipeline.fit(df).transform(df)

##### 

##### # Balance the classes before feature assembly

##### class\_0 = df\_prepared.filter(col("tri\_class\_outcome\_index") == 0)

##### class\_1 = df\_prepared.filter(col("tri\_class\_outcome\_index") == 1)

##### class\_2 = df\_prepared.filter(col("tri\_class\_outcome\_index") == 2)

##### 

##### majority\_size = class\_0.count()

##### 

##### class\_1\_over = class\_1.sample(withReplacement=True, fraction=majority\_size / class\_1.count(), seed=1234)

##### class\_2\_over = class\_2.sample(withReplacement=True, fraction=majority\_size / class\_2.count(), seed=1234)

##### 

##### df\_balanced = class\_0.unionAll(class\_1\_over).unionAll(class\_2\_over)

##### 

##### # Verify the class distribution after balancing

##### df\_balanced.groupBy("tri\_class\_outcome\_index").count().show()

##### 

##### # Assemble feature columns into a single feature vector

##### assembler = VectorAssembler(

##### inputCols=["drug\_name\_index", "suspect\_role\_index", "reaction\_type\_index", "age\_group\_index", "gender\_index"],

##### outputCol="features"

##### )

##### 

##### df\_balanced = assembler.transform(df\_balanced)

##### df\_balanced = df\_balanced.select("tri\_class\_outcome\_index", "features")

##### 

##### # Split the balanced data into training and test sets

##### train\_data\_balanced, test\_data\_balanced = df\_balanced.randomSplit([0.8, 0.2], seed=1234)

##### 

##### rf\_balanced = RandomForestClassifier(labelCol="tri\_class\_outcome\_index", featuresCol="features", maxBins=3000)

##### model\_balanced = rf\_balanced.fit(train\_data\_balanced)

After training, the model produced an overall accuracy of approximately *63.8%*, with test weighted precision at *0.6387*, test weighted recall at *0.6383*, and a test F1 score of *0.6349*. These metrics indicate that the model effectively identifies severe adverse events while maintaining a reasonable balance between precision and recall.

#### Fig. Confusion matrix of the model.

With this trained RandomForestClassifier model, the company can predict the severity of adverse events, enabling the company to identify high-risk patient groups, such as older adults. The company can enter the data containing patient information and medication usage details. The model will predict the outcome for each patient. This strategic approach helps ensure the safety of medications, particularly for high-risk groups, thereby guiding PharmaCARE in making informed decisions in drug development.

# Conclusion

This comprehensive analysis of the FDA Adverse Event Reporting System (FAERS) using Neo4j has provided PharmaCARE with critical insights into the patterns and risks associated with cardiovascular-related adverse events. By leveraging the power of graph databases, we have effectively mapped the complex relationships between drugs, reactions, and patient demographics, offering a multi-dimensional view of the data that traditional databases could not easily provide.

Our investigation into the most common outcomes for cases involving myocardial infarction, alongside an in-depth analysis of demographic factors influencing cardiovascular-related drug reactions, has highlighted key areas for PharmaCARE to consider in future drug development strategies. The use of advanced machine learning techniques, such as RandomForest for predicting the likelihood and severity of adverse reactions, has further equipped PharmaCARE with the tools to preemptively address potential safety concerns.

The predictive models developed during this study not only enhance PharmaCARE's ability to ensure patient safety but also facilitate a more targeted approach in drug development. This proactive strategy is particularly crucial in the high-stakes field of cardiovascular diseases, where the margin for error is minimal and the impact on patient health is significant.

In conclusion, our analysis underscores the importance of integrating advanced data analytics and machine learning in the pharmaceutical industry. The insights derived from the FAERS data through Neo4j have set a new standard for PharmaCARE in its commitment to innovation and patient safety, guiding the company towards safer, more effective therapeutic solutions for cardiovascular diseases.

# References

###### [Healthcare Analytics - Load and analyse FDA Adverse Event Reporting System Data with Neo4j](https://github.com/neo4j-graph-examples/healthcare-analytics/blob/main/documentation/healthcare-analytics.adoc)

###### [Machine learning guided association of adverse drug reactions with in vitro target-based pharmacology - PMC](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7379147/)

###### Ali, Usman, and Muhammad Aoun. "Machine Learning and FAERS Data: Revolutionizing Health Care Analytics for Adverse Drug Reaction Prediction." International Journal of Applied Health Care Analytics 8.3 (2023): 1-18.