Prediction of polypharmacy side effects using network analysis and machine learning

-Bisman Singh

I Introduction

When treating diseases such as Cancer, HIV or even some ailments like cough and cold. A lot of time the patients are prescribed multiple drugs. Now the simultaneous usage of drugs is known as polypharmacy. The simultaneous usage of drugs may lead to one or more side effects. The problem is which pairs of drugs may or may not lead to certain side effects. This problem can be viewed as a link prediction problem. Where the nodes are the drugs and the link represents the existence of side effects. In this project , the aim would be to solve this link prediction problem by the help of network analysis and machine learning.

1.1 Dataset

The dataset was taken from Stanford Biomedical Network Dataset Collection (bio-decagon-combo)[1]. Where the dataset represents the combination of two drugs which have certain side effects. Below is a small sample of graph representation of the data.

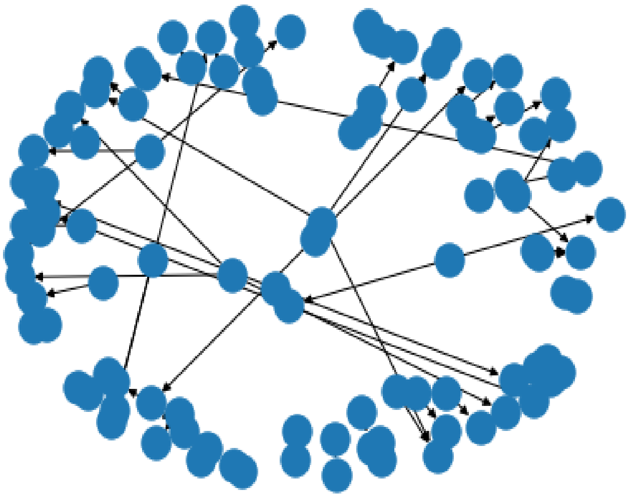


Figure 1 : Graphical representation of the data

The whole dataset consist of 645 drugs and 14,000 unique edges, representing pairs of drugs which lead to some side effect

II Methods

2 Analysis of the Network

In this section we will be analysing various node levels as well as edge level statistics. It is important to analyse such statistics to get a better understanding of the network and to make features of our machine learning model.

2.1 Degree of Nodes

Degree of node is the basic level statistics which represents the number of connections a node has with another node. Usually the notation of degree of the ith node is k_i

$$k_{ii} = \sum_{j=1}^{n} A_{ij}$$

Where A_{ij} is the adjacency matrix of the network[2]

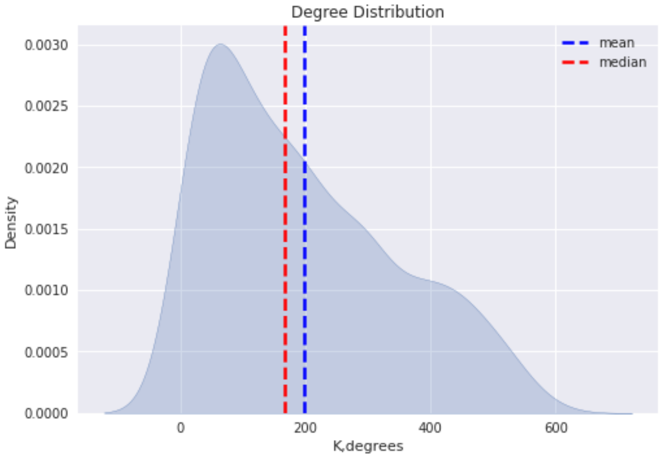


Figure 2: Degree Distribution of the network

The above figure describes the degree distribution of the drug-drug network, the distribution is right skewed. The mean of the distribution is 196 and median is 167.

2.2 Eigenvector Centrality

[1] (n.d.). Decagon is a graph convolutional neural network ... - SNAP: Stanford. Retrieved December 12, 2022, from http://snap.stanford.edu/decagon/

Eigenvector Centrality is an algorithm that calculates centrality of a node based not only just on the neighbours but also on the centrality of the neighbours. Eigenvector centrality is a measure of the influence of nodes in a network. At the end it is then given a relative score provided for all the nodes. Meaning a high-scoring node will be connected to many high-scoring nodes. Mathematically when can use eigenvector notation i.e

$$\mathbf{A}x = \lambda x$$

Where A is the adjacency matrix of graph with λ being the eigenvalue. Where the centrality of value of the ith node is the ith value in vector x[3]

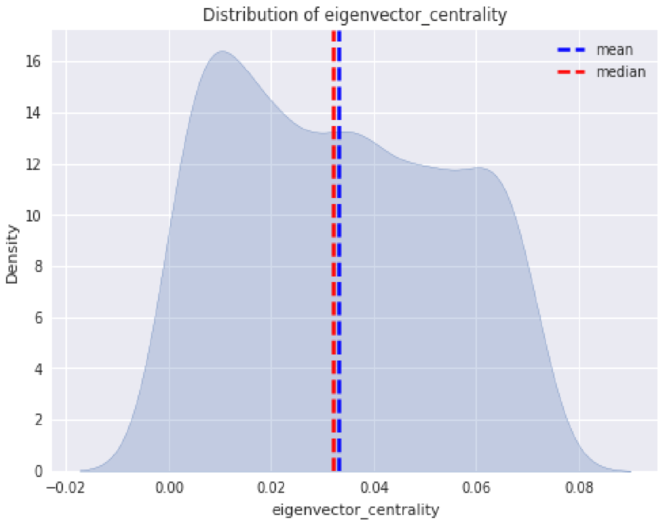


Figure 3 Distribution of Eigenvector Centrality

Here we see that the mean of the distribution is 0.0330 and median is 0.0320.

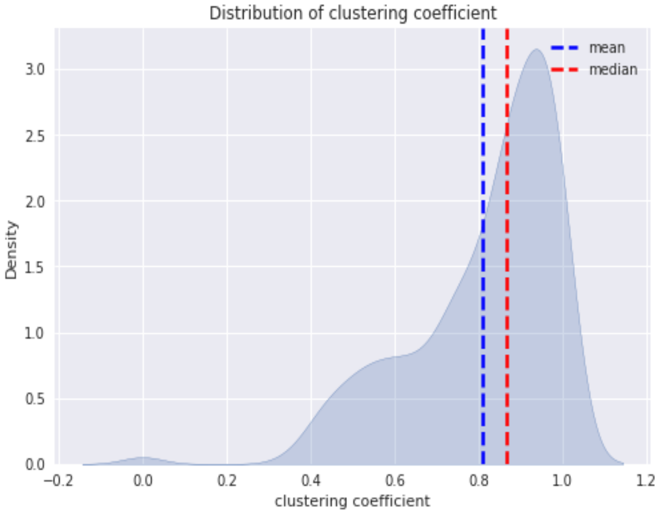
2.3 Clustering Coefficient

Clustering Coefficient is defined as a node level measure, which captures the fraction of neighbours j and k of a node i that are themselves connected by j and k. In other words this algorithm represents how close its neighbours are close to being a clique. Mathematical formulation is below

$$\mathbf{C_i} = \frac{1}{ki(1-ki)}\sum_{j,k} \mathbf{A_{ij}}\ \mathbf{A_{jk}}\mathbf{A_{ki}},$$

$$\mathbf{k_i} = \sum_{j} \mathbf{A_{ij}}$$

Where the A is the adjacency matrix of the graph and k_i is the degree of node i. [4]



Here we see that the mean of the distribution is 0.8114 And median is 0.866

2.4 Jaccard Score

Jaccard score is used across a lot of problems, where the score is representation of heterogeneity and homogeneity of sets. In network analysis Jaccard score represents the fraction of intersection neighbours of two nodes and the union of same nodes. Mathematically

$$\mathbf{J(U,V)} = \frac{|\Gamma(u) \cap \Gamma(v)|}{|\Gamma(u) \cup \Gamma(v)|}$$

Where J(U,V) is Jaccard score of nodes U and V and $\Gamma(v)$ is the neighbours of node V[5]

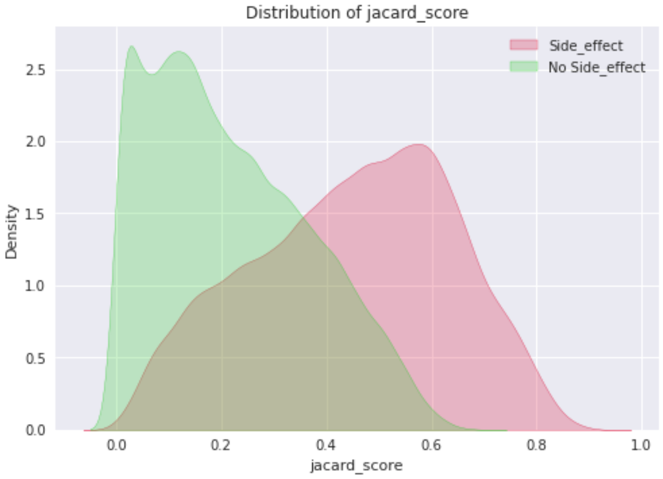


Figure 4 Distribution of Jaccard Score

Here we see that the Jaccard score gives us two distributions where one corresponds to the distribution of the Jaccard score with no side effects and the other with side effects.

2.5 Adamic_adar_index

Adamic adar index is measure which gives the existence of an edge based on numbers of neighbours shared between two nodes

$$A(x,y) = \sum_{u \in N(x) \cap N(y)} \frac{1}{\log |N(u)|}$$

Where N(u) is the neighbours of node u. [6]

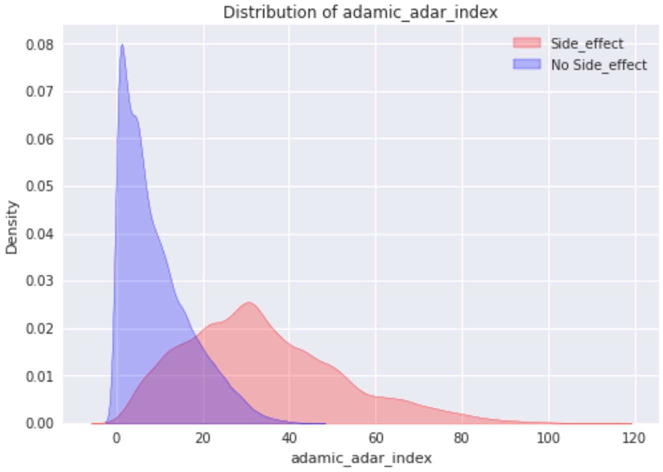


Figure 5 : Distribution of Adamic Adar Index

Here we see a clear difference between the behaviour of adamic adar index in pairs of nodes that have a side effect and those which do not. For example if the adamic score of a pair of nodes is more than 40, there is high probability that the pair will lead to a side effect. As one should remember that adamic adar is a measure which is relative and not to be confused with a raw value which can provide information in isolation. One thing to note higher the score higher is the probability of nodes being close and therefore having an edge.

2.6 Preferential attachment

The Preferential attachment is the measure of product of neighbours of two nodes. This measure is based upon the rule that the more connected a node is the more it is likely to receive new links. Mathematically we can defined the measure as product of neighbours of two nodes

$$|\Gamma(u)||\Gamma(v)|$$

$\Gamma(v)$is the neighbours of node V[5]

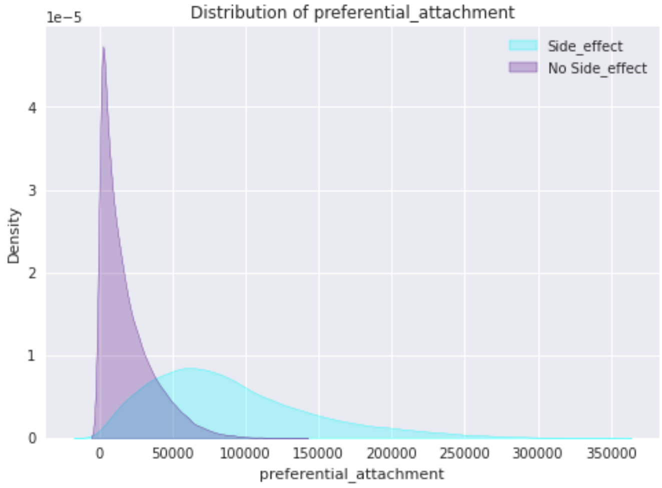


Figure 6 : Distribution of Preferential Attachment

One conclusion that can be drawn there is an existence of side effects when drugs are having side effects with many other drugs. That implies if both drugs in a pair have a high degree meaning that the drugs have side effects with other drugs as well then that particular combination has high probability of having a side effect

2.7 Common Number of Neighbours

Common number of neighbours is basically the set of common neighbours between nodes. Mathematically

$$\Gamma(u) \cap \Gamma(v)$$

$\Gamma(v)$ is the neighbours of node V

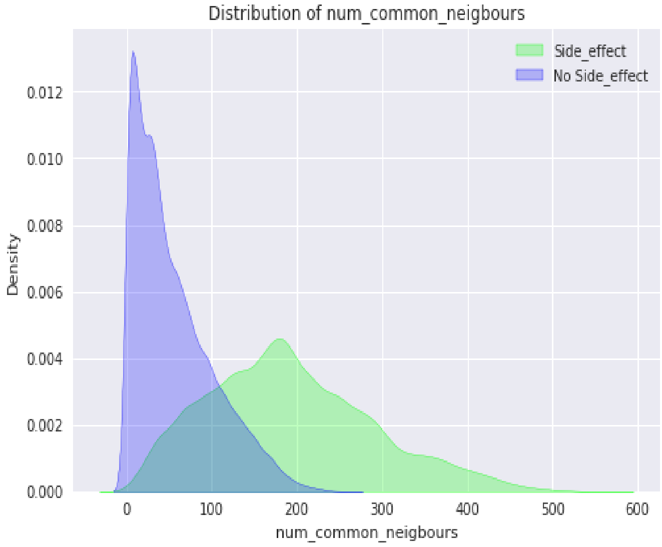


Figure 7 : Distribution of number of common neighbours

Here the observation is that more the common neighbours the higher the probability of having possible side effects. But one drawback of this approach is that it does not consider the degrees of a node. The following approach tries to solve that drawback.

2.8 Cosine similarity

Cosine similarity is a measure which takes the degree of nodes and the common neighbours into account. Mathematically it can be expressed as following

$$\frac{\Gamma(u) \cap \Gamma(v)}{\sqrt{k_u}\sqrt{k_v}}$$

Where k_u is the degree of node u. As one can note cosine similarity is the ratio of common neighbours to the geometric mean of degrees of a given node. Here if the value is 1 that implies that the degrees are equal in both the nodes and hence are only connected to the same nodes. Whereas if the value is 0 that implies there are no common nodes[6]

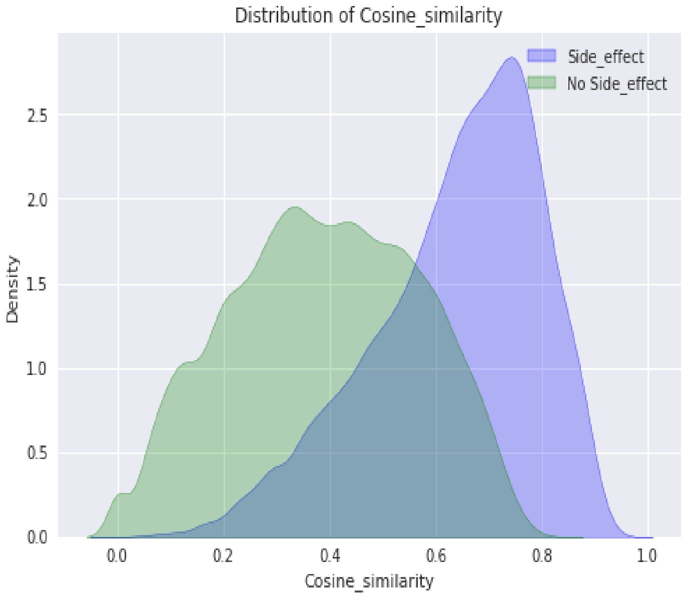


Figure 8 : Distribution of Cosine Similarity

Here the observation the higher the cosine similarity the higher the probability of having the side effect.

3 Using Machine Learning Models

The given task here is to pose the link prediction problem as a supervised classification problem.

The following steps are to be followed in order to solve this problem as a supervised classification problem

1)Feature Generation

2) Dataset split for training and testing

3)Applying Machine Learning Models

4)Choosing appropriate metrics

5) Compare models on the specified metric

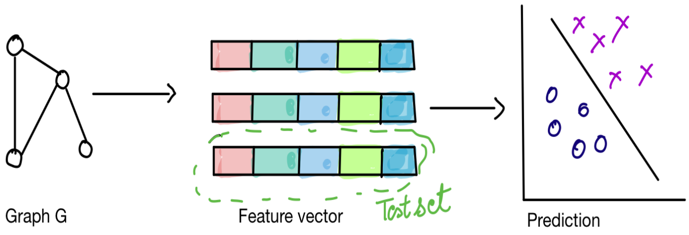


Figure 9: Steps to apply machine learning model

3.1 Feature Generation

In order to provide data to the machine learning model. The above node as well edge level features were stored. Total 13 features were produced(including Drug and Drug 2 features).

The following features were generated:

i) Drug1_degree, Drug2_degree: Degrees of both drugs

ii)Drug1_betweenness_centrality, Drug2_betweenness_centrality : Betweenness Centrality

iii) Drug1_clustering_coeficient, Drug2_clustering_coeficient: Clustering coefficient for the given pair

iv) Drug1_eigenvector_centrality, Drug2_eigenvector_centrality: Eigenvector centrality for the pair of drugs

v) Jacard_score

vi) Adamic_adar_index

vii) Preferential_attachment

viii) Cosine_similarity

ix) Num_common_neigbours

3.2 Dataset Preparation

It is important to prepare a dataset for our machine learning models. So the idea here is to feed the node level features like the degree of nodes of each node a pair at the same time provide with edge level features like jaccard score, academic adar index etc. Therefore some set of features are specified for each pair of nodes i,j . After feature vectors have been specified, a binary classifier such as logistic regression, decision trees etc can be applied. Here the models used are logistic regression and random forest etc.

3.2.1 Negative sampling

In the dataset we are given only the edges which lead to certain side effects. Therefore we are not given the drug pairs which do not lead to any side effects. Now the task is to sample those drug combinations which do not lead to any side effects. First is to find all possible pairs of drugs. So the number of drugs given in the dataset was 645. All possible edges would be 207690, after finding all possible combinations it would be important to remove edges which have a side effect. In order to produce a balanced dataset, the number of pairs with no drug samples was equal to the number of pairs which lead to side effects. This process of sampling edges is also known as negative sampling. The processed data set provided in the github link.

3.2.2 Splitting Dataset into Training and Testing

The dataset is then split into training and testing. 20 percent of the data points (Drug pair along with their features) are randomly selected and the models are trained upon the remaining 80 percent data points.

4 Machine Learning Models

4.1 Logistic Regression

Logistic regression is a classification model, which models the relationship between a set of features(predictor variables) and the categorical feature(target variable). In our problem we have node level and edge level features for every corresponding drug pair (i,j) and the target variable would be the existence of an edge , which also means that the drug (i,j) has a side effect. This mode is derived from the function called sigmoid function.

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

Where the sigmoid function gives a value between 0 and 1. Therefore the logistic regression model gives us a probability of a certain event occurring given the features vectors . The equation is written below

$$Pr(Y_i = 1|X_i) = \frac{exp(\beta_0 + \sum_{i=1}^{n} \beta_i X_i}{1 + exp(\beta_0 + \sum_{i=1}^{n} \beta_i X_i)}$$

Where we see that X_i is the feature vector and β_o is the bias and β_i 's are the regression coefficients.

4.1.2 Performance of the Model

The model gave an accuracy score of 0.8685 and an auc_roc score of 0.9441. Below is the confusion matrix produced from the prediction given by logistic regression

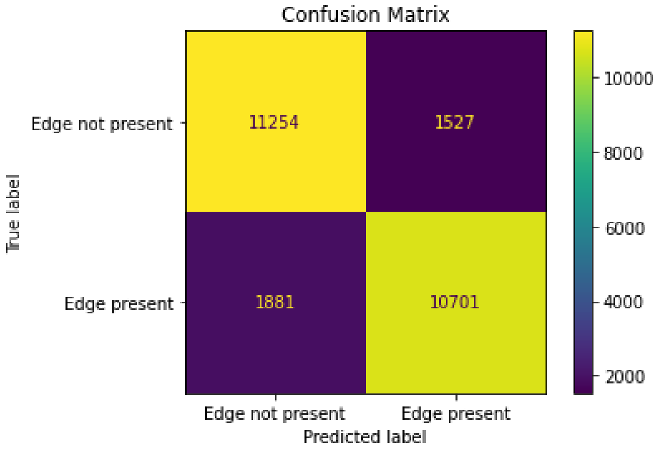


Figure 10 : Confusion matrix for logistic regression

4.2 Decision Tree

Decision Tree is a non parametric supervised learning method used for classification.[7] The decision trees can be thought of as conditional rules. Therefore after the model learns these rules it is able to classify the data into various categories

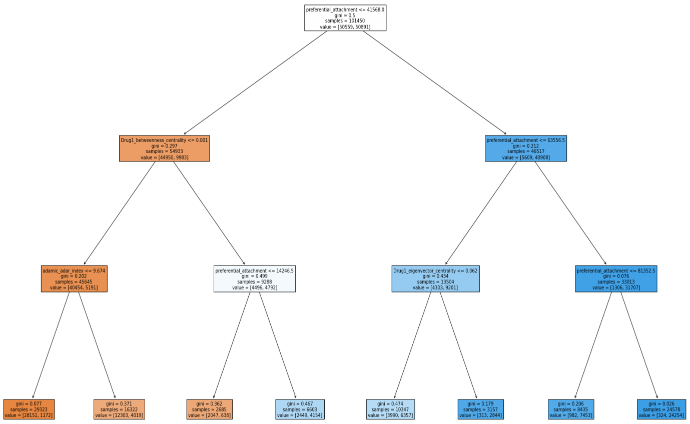


***Figure 11: Tree plot for the decision tree model**

*In order to see the plot clearly , you might have to zoom in

In the figure we see that preferential attachment is the most important feature here as it gains the most information about the labels. Similarly we can see other features. The maximum depth of the decision tree model is 3. That is why there are 3 splits in the plot.

4.2.1 Performance

The model here was able to produce an accuracy score of 0.854 and AUC score of 0.927. Below is the confusion matrix for the mode

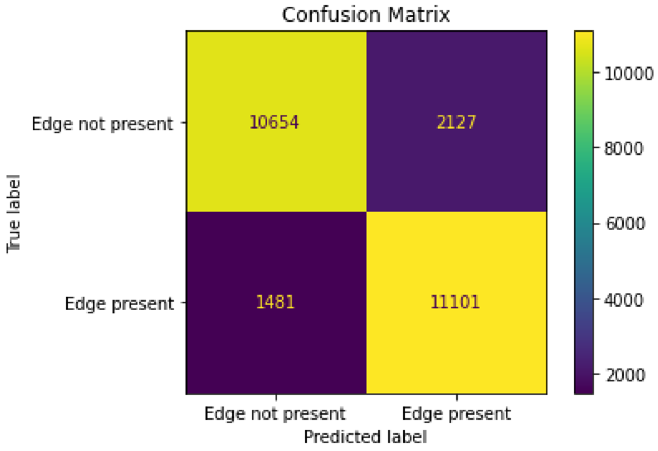


Figure 12 : Confusion Matrix for Decision Tree model

4.3 Random Forest

Random Forest is a machine learning model which is based on the concept of ensemble learning.[8] Ensemble learning concept uses multiple predictors, here the multiple predictors are decision trees. The classification of a data point to class is decided by majority of trees predicting that particular class

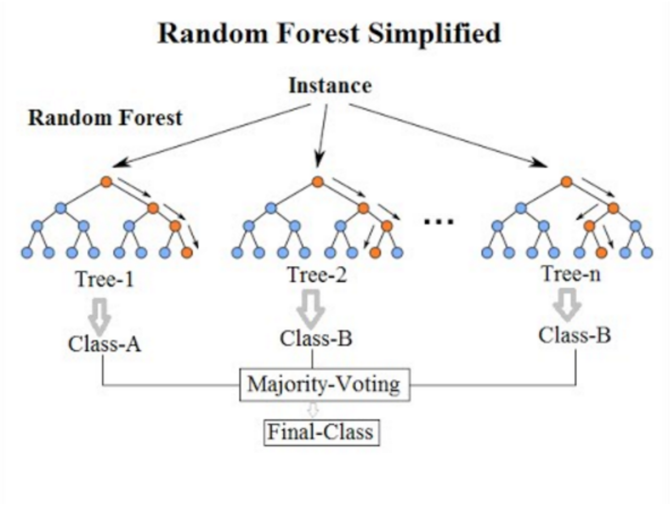


Figure13:https://en.wikipedia.org/wiki/Random_forest

4.3.1 Feature Importance

One benefit of using random forest is that it helps to describe which features are important. The feature importance is based on Gini- importance. Gini importance is an essential concept in random forest. It measures the information gain by splitting the tree on the basis of a variable . Below is the feature importance given the model for our dataset

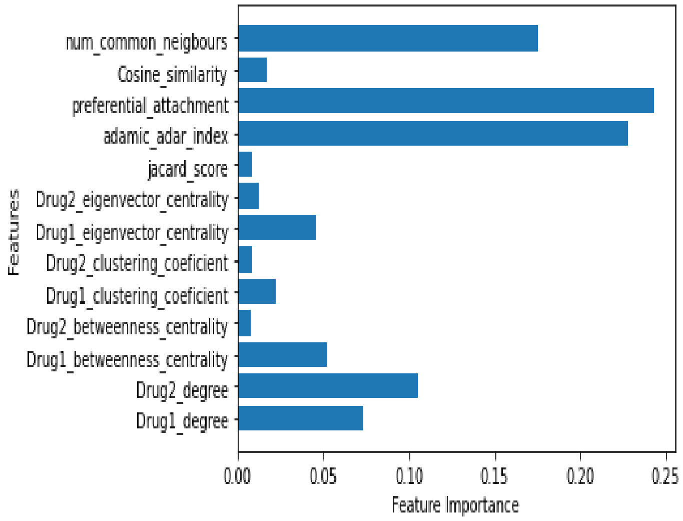


Figure 14 : Feature Importance in Random Forest

Here the preferential attachment, adamic adar index and number of common neighbours are top three important features. This tells us that the existence of an edge(side effect) has a relationship with the number of common neighbours or rather degrees of nodes, as we see the features which are important are a function of common neighbours or degrees of the nodes.

4.3.2 Performance

The random forest produces an accuracy score of 0.852777 and AUC score of 0.933052. Below is the confusion matrix

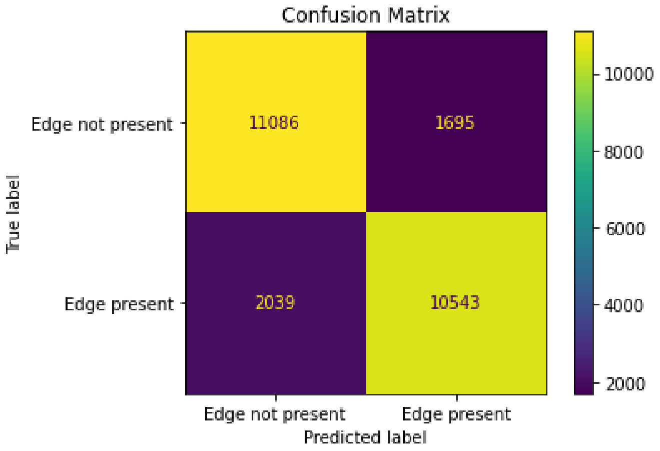


Figure 15 : Confusion matrix for Random Forest

III Results

5 Metrics

5.1 Accuracy

Accuracy score is a metric for quantifying the performance of the model. It is the basic metric where it tells on average how accurately the model is to predict correctly. Mathematically we can define accuracy as

$$\frac{Correct\ prediction}{Total\ number\ of\ predictions}$$

Although there is another way of defining accuracy which well provide better understanding for following metric(AUC)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Where
i) TP = True Positives
ii)TN = True Negatives
iii) FP = False Positives
iv) FN = False Negative

One limitation of accuracy is that it can allow bias to creep in. If we are having an unbalanced dataset where 80 percent of data points have an edge and the other 20 percent does not. So if our model predicts all the 100 percent of data points have an edge. The accuracy score will be 80 percent although our model performed really poorly but the accuracy is still. Therefore in order to overcome this limitation AUC-ROC is used.

5.2 AUC-ROC Score

AUC-ROC score is a graphical way to measure the performance classification model. So the AUC is the measure of the area under the ROC curve. Therefore the higher the score the better the model is performing.

5.2.1 ROC Curve

The ROC curve depicts the rate of true positives with respect to the rate of false positives at different thresholds, therefore highlighting the sensitivity of the classifier model.An ideal classifier will have a ROC where the graph would hit a true positive rate of 100% with zero false positives.[9]

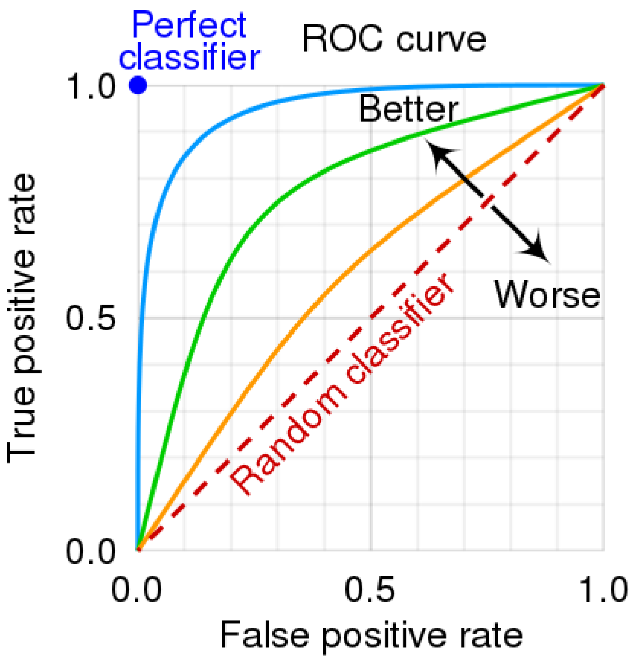


Figure15:https://en.wikipedia.org/wiki/Receiver_operating_characteristic

7. Model Comparisons

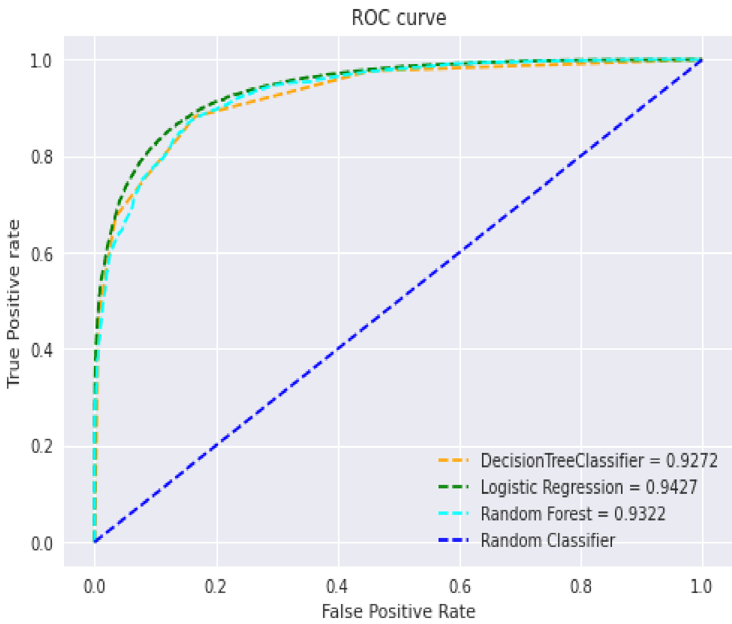


Figure 16 : ROC-AUC curves for all the models

The highest AUC-ROC score is of the logistic Regression Model , although there is not a big difference in AUC-ROC score as well as in accuracy score across the three models. Therefore we can conclude that all the models perform well on our data

Model	Accuracy Score	AUC_ROC score
Logistic Regression	**0.85774**	**0.9272**
Decision Trees	**0.86563**	**0.9427**
Random Forest	**0.85277**	**0.9322**

IV Discussions

After analysing and doing predictive analysis over this dataset. I can conclude the following :

1) Another important learning that was gained from this project was usually see very different behaviours of node and edge level statistics in social networks but in networks like this, the behaviour of such statistics are completely different

2) Edge level features play an important part in the link prediction tasks.

3) Especially those features which are dependent upon the degrees and the number of common neighbours between two nodes.

4) Another important learning that was gained from this project was usually see very different behaviours of node and edge level statistics in social networks but in networks like this, the behaviour of such statistics are completely different

5) Earlier when was doing this project , I was including path features which lead to overfitting and data leakage, therefore when dealing link prediction problem one has to keep in mind about the data leakage

References

1.Clauset, Aaron. 2022. *Lec2: Representing and describing networks*.

2.“eigenvector_centrality — NetworkX 2.8.8 documentation.” n.d. NetworkX. Accessed December 12, 2022. https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.centrality.eigenvector_centrality.html.

3.“jaccard_coefficient — NetworkX 2.8.8 documentation.” n.d. NetworkX. Accessed December 12, 2022. https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.link_prediction.jaccard_coefficient.html.

4. “adamic_adar_index — NetworkX 2.8.8 documentation.” n.d. NetworkX. Accessed December 12, 2022. https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.link_prediction.adamic_adar_index.html.

5.“preferential_attachment — NetworkX 2.8.8 documentation.” n.d. NetworkX. Accessed December 12, 2022. https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.link_prediction.preferential_attachment.html.

6.“Cosine similarity.” n.d. Wikipedia. Accessed December 12, 2022. https://en.wikipedia.org/wiki/Cosine_similarity.

7. “1.10. Decision Trees — scikit-learn 1.2.0 documentation.” n.d. Scikit-learn. Accessed December 12, 2022. https://scikit-learn.org/stable/modules/tree.html.

8. “Random forest.” n.d. Wikipedia. Accessed December 12, 2022. https://en.wikipedia.org/wiki/Random_forest.

9.Dey, Victor. 2021. “Understanding the AUC-ROC Curve in Machine Learning Classification -.” Analytics India Magazine. https://analyticsindiamag.com/understanding-the-auc-roc-curve-in-machine-learning-classification/.

10. Clauset, Aaron. n.d. *Lecture 6:Predicting Missing Links in Networks*.

11.Newman, Mark. 2018. *Networks*. N.p.: OUP Oxford.

12. https://academic.oup.com/bioinformatics/article/34/13/i457/5045770?login=false

Appendix (Code)

Github Link : https://github.com/bis1999/CSCI-5352-Network-Project-Bisman-Singh-