CLASSIFICATION MODEL COMPARISON BETWEEN DECISION TREE AND NAÏVE BAYES IN FRAUD

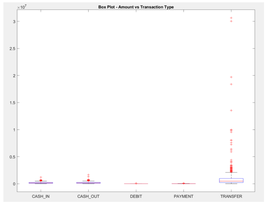
Description and Motivation of the Problem:

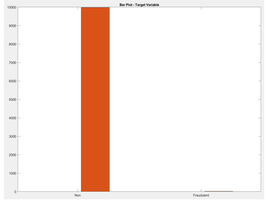
- The aim is to compare the performance of a Decision Tree model verses a Naïve Bayes Model in binary classification problem.
- The problem is to predict if clients' transactions are fraudulent or not based on transaction information.
- Our results will be compared to S Rahmadani paper comparing Decision Tree vs Naïve Bayes [3]

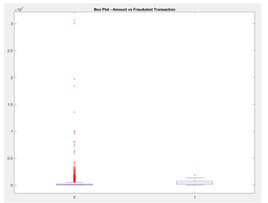
Exploratory Data Analysis:

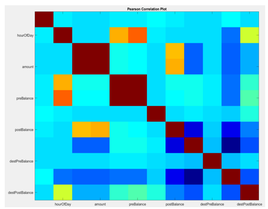
- The dataset is a "Synthetic Financial Datasets for Fraud Detection" from the Kaggle website.
- The data is generated by PaySim, private data is aggregated to generate a synthetic dataset that resembles normal operation of transactions.
- The data contains 6,362,620 transactional records. This would be a huge computational cost, as a result a random sample of 100,000 records will be taken as sub-dataset.
- The original data contains 11 columns including the target column in which 3 columns are categorical and the rest are continuous numerical.
- Non-relevant columns associated with account codes will be removed. Transaction type will be converted to dummy variables for the models to process.
- Box plot "Amount vs Transaction Type" shows that transfers have a large amount value and a wider range compared to the rest. Box Plot "Amount vs Fraudulent" shows fraudulent transactions do have a wider range, larger amounts and only one outlier. Due to the low number of outliers the assumption is it will be easier to learn what is a fraudulent transaction without creating a complex model.
- Looking at the target variable, 1 indicated fraudulent and 0 for non-fraudulent. In the sample data only 15 values are indicated as 1 as seen in the Bar plot. This is a clear case of an unbalanced dataset with class 1 only representing (0.15%). This means the models will have a hard time potentially learning to classify what is fraudulent.
- The Correlation matrix representing Pearson's correlation between the predictors, looking closely there doesn't seem to be any strong correlation between any of the predictors and the target column.

	hour_of_day	amount	oldbalanceOrg	newbalanceOrig	oldbalanceDest	newbalanceDest	isFraud
count	100000.000000	1.000000e+05	1.000000e+05	1.000000e+05	1.000000e+05	1.000000e+05	100000.000000
mean	7.799480	1.797049e+05	8.404981e+05	8.613771e+05	1.103992e+06	1.228195e+06	0.001270
std	4.622563	5.841259e+05	2.911749e+06	2.946836e+06	3.365951e+06	3.620189e+06	0.035615
min	0.000000	3.400000e-01	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000
25%	5.000000	1.338441e+04	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000
50%	8.000000	7.506112e+04	1.457100e+04	0.000000e+00	1.363914e+05	2.196725e+05	0.000000
75%	11.000000	2.082481e+05	1.073766e+05	1.457629e+05	9.536782e+05	1.118084e+06	0.000000
max	24.000000	4.782507e+07	3.382981e+07	3.386997e+07	1.960212e+08	1.960025e+08	1.000000





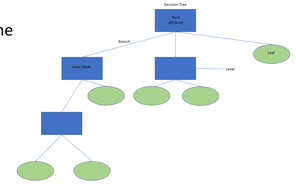




Machine Learning Models

Decision Tree

- This method is known as a predictive model and is quite efficient and reliable in generating classifiers using binary decisions once trained.
- The algorithm uses a top-down method in its learning process, the result is the predictor space is divided into rectangular splits [1].
- It begins by selecting a feature as the root and creates a branch for each value, it then divides the cases into branches and repeats the process until each case on the branch has the same class.
- Each path from the root to the decision leaf can be transformed into a rule by conjoining the tests along the path.

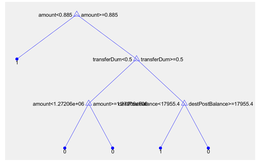


Pros:

- Decision Trees are considered flexible as they can handle a variety of input data such as nominal, numeric and textual.
- Adaptability in handling data with missing values or errors.
- They have a high predictive performance with small computational effort.
- Useful for large datasets.

Cons:

- The Decision Tree's greedy characteristics leads to an over-sensitivity of the training set.
- The instability in a Decision Tree is caused be irrelevant attributes and noise, the whole subtree can change with any minor split near the root of the tree [1].
- Complex trees will try to accommodate noise in the data so it can be designed to fit arbitrary complex boundaries between points, which mean the model is overfitting the data. This can be tackled with pruning but will increase the re-substitution loss.



Naïve Bayes

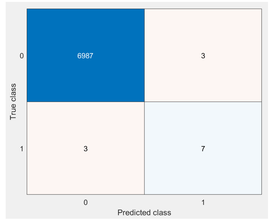
- Naïve Bayes is an example of a simple generative model.
- Before Naïve Bayes uses a decision rule to make predictions, the joint probability of the target and the features are estimated, then the Posterior Probabilities are calculated using Bayes Theorem.
- The classifier will then predict that the target belongs to the class with higher posterior probability, conditioned on it [2].
- Naïve Bayes classifier assumes that there are no dependencies amongst attributes. This assumption is called class conditional independence.

Pros:

- This simplicity of class conditional independence equates to computational efficiency, which makes this technique attractive.
- Since the classifier returns probabilities, applying the results to different tasks are easier in comparison to an arbitrary scale.
- It improves the classification performance by removing the irrelevant features.

Cons:

- The Naive Bayes classifier requires a very large number of records to obtain good results.
- Less accurate in comparison to other classifiers on some datasets.
- The strong assumption of independence class features makes it hard to find relevant data.
- Naïve Bayes is known with an issue called the Zero Conditional Probability Problem. This problem wipes out information but can be corrected using Laplacian correction [3].



Hypothesis Statement:

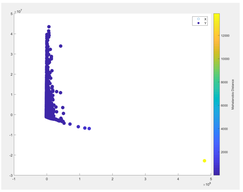
- The expectation is both models to perform well and better than a random prediction with the expectation of the Decision tree to out preform the Naïve Bayes model.
- The Decision tree model is expected to have a lower error rate due to the issue of overfitting the data.
- The model's performances are dependent on altering hyperparameters and feature selection.
- Due to the simplicity of a Decision Tree we will look to improve the model by looking at Random Forest.

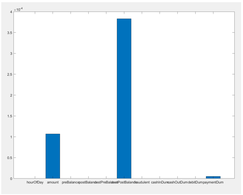
Methodology Description:

- With no further alteration to the data, it will be split using a holdout method at 70% for training and 30% for testing. Each model will be fitted to run on the training data and the re-substitution loss will be calculated to see the performance. The model will be exposed to the testing data and the loss will be calculated and the target predicted. The model will then be evaluated with a re-run using a K-fold cross validation on the training data.
- The predictions will be analysed by calculating true positives, true negatives, false positives and false negatives which will visualised using a confusion matrix. The previous calculations will be used to calculate precision, recall and f1 measures for comparison.
- Compute the posterior probabilities to then plot the ROC curve and calculate AUC figure for comparison.
- Improving the models will then start with Pruning, Predictor Importance, Principal Component Analysis, optimising the hyper-parameters and using Random forest. Outliers will then be removed using the Mahalanobis Distance between the two pca components (first image on the right). All previous steps will be repeated.

Choice of Parameters and Experimental Results:

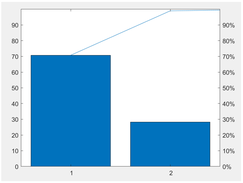
- For the tree model we look at predictor importance to see the relative importance of each predictor. We will eliminate unwanted predictors that had no significance on the Tree decision leaving two predictors (second image on the right).
- Using principal component analysis we reduce the dimensionality by discarding the components beyond the 100% explained variance, leaving only two components (third image on the right).
- For improving the Decision Tree I will look to create an ensemble of trees (random forest) to overcome the its weak learning ability.
- For Naïve Bayes using the setting the distribution to Kernel to not assume normality this helped improve the prior and the predictions.
- In the end after improving the models they were in some instances able to improve on precision but were unable to improve on predicting true positives will lead to an error in calculating the recall value.
- The Naïve Bayes model was able to preform well in terms of predicting true positive with a bad precision value.
- Overall the Decision Tree was the better model as predicted as it was able to predict well as shown in the table of results below with a good precision and recall value even though AUC value suggests the Naïve Bayes was the better model.

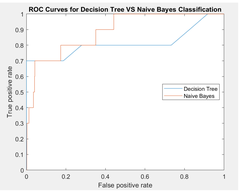




Analysis and Critical Evaluation of Results:

- *Both models have a high bias and low variance.*
- *The optimal hyper-parameters for Naïve Bayes was density set to kernel and width to* 3.7688e+05, this leads to a more complex model. Ensemble Learning was used to improve on the Decision Tree model using a "Bag" method (Random Forest) with a limit of 30 cycles.
- *As predicted the Decision Tree performed better than the Naïve Bayes model with 0.7 for each precision, recall and f1 score. Looking at the DT confusion matrix (bottom left) the model was able to predict 6987 true negative and 7 true positives. The model predicted 3 false positives, but Banks can overcome false positives predictions by customer verification. The model predicted 3 false negatives which means banks would have to refund the loss these loses.*
- *The Naïve Bayes model predicted the same for true positives and false negatives, but its precision was weak with 306 false positives. Banks would be verifying a lot more transactions, could this effect customer satisfaction. Overall, the recall value is very important to this dataset. Even though both models had an equal recall value the DT model had better precision making it the preferred model. This was different from the results in [3] where Naïve Bayes performed better than the Decision Tree.*
- *After optimising the models, accuracy and precision improved but not one prediction was classified as a true positive or a false positive despite feature selection. This is a huge issue of overfitting in an unbalanced dataset. This was different from the results in [3] where the Decision Tree improved more than the Naïve Bayes model.*
- *Both models were extremely fast in training. The Decision Tree was faster in prediction given the logical rules it creates to classify any new data. The Naïve Bayes was much slower in predicting in comparison to all set timings.*
- *The AUC value for Naïve Bayes is slightly greater than the Decision Tree model which indicates it's the preferred model to choose but given the dataset the Decision Tree has a better K-Fold loss, precision and f1 values with faster training and prediction time, its regarded as the better model.*

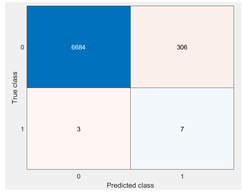




	Method	K-Fold Loss	Precision	Recall	F1	AUC	Train Time	Pred Time
Model A	Decision Tree	0.00085714	0.7	0.7	0.7	0.81151	0.81151	0.018884
Model D	Naïve Bayes	0.044143	0.022364	0.7	0.043344	0.88943	1.0538	8.3618

Lessons Learned and Future Work:

- Any attempt to improve the models resulted in extreme overfitting, all predictions were non-fraudulent. A trade off between improving the model and model complexity is present. The right balance is required.
- With an unbalanced dataset, manipulating the data to have a more balanced spread of either class to improve the models or using an algorithm that is better equipped like anomaly detection.



References:

1. Rokach L and Maimon O 2005 Top-down induction of decision trees classifiers-a survey IEEE Trans. Syst. Man Cybernetics Part C (Appl. Rev.) 35(4) pp 476-87.
2. Sayali D. Jadhav and H. P. Channe. "Comparative Study of K-NN, Naive Bayes and Decision Tree Classification Techniques" IJSR, 2016.
3. S Rahmadani et al 2018 J. Phys.: Conf. Ser. 978 012087.
4. N. B. Amor, S. Benferhat, Z. Elouedi, "Naive Bayes vs Decision Trees in Intrusion Detection Systems," ACM, 2004.
5. Brijain R. Patel and Kushik K.Rana, "A Survey on Decision Tree Algorithm for Classification", International Journal of Engineering Development and Research, 2014.
6. BhaveshPatankar and Dr. Vijay Chavda, "A Comparative Study of Decision Tree, Naive Bayesian and k-nn Classifiers in Data Mining", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 4, Issue 12, December 2014.
7. Breiman L, Friedman J, Stone CJ and Olshen RA 1984 Classification and regression trees (Monterey: CRC press).
8. Rokach L, Maimon O 2008 Data mining with decision trees: theory and applications. (Singapore: World Scientific Pub Co Inc).

By TAHER DARWESH