Research Problem

Telemarketing is an effective strategy of selling goods and services by using phone calls. With the ever-growing competition rates in the business world, tele-sales need innovation. Machine learning can boost this business by identifying potential buyers more precisely. In this project, we will be working with direct telemarketing campaign data of a Portuguese banking institution, retrieved from MCI Machine Learning Repository(link).

This is a binary classification problem, where target variable indicates if term deposit was purchased or not. This target is highly imbalanced, out of 37000+ instances only 3500 observations have purchased term deposit. The overwhelming 'not purchased' class has outnumbered 'purchased' class by 10:1 ratio. Our goal is to identify these potential buyers. Without any modelling, it takes an average of 11 calls (total number of data /numbers of purchased data) to find out 1 term deposit buyers. We want to reduce number of calls needed to find out the buyers, so that cost and resource utilization can be optimized. However, in this process, the model should not misjudge a buyer as non-buyer.

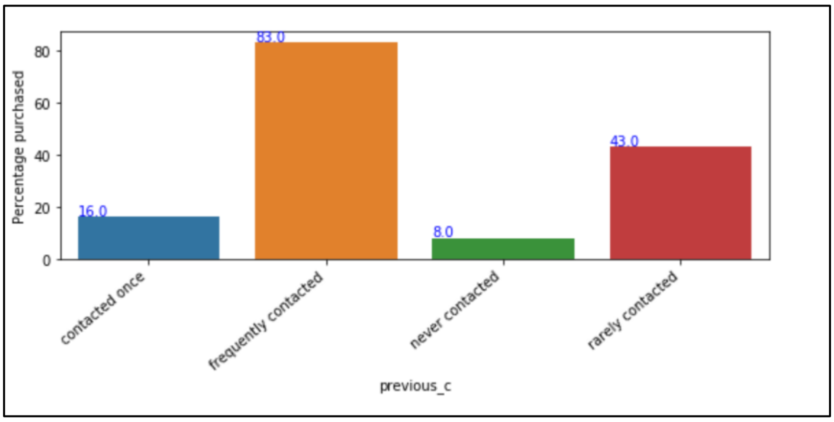
These model's efficiency will be established once they are able to reject below null- hypotheses

- Null hypothesis 1- Developed model does not perform better than baseline model.
- Null hypothesis 2 – In case multiple models are built, they perform equally good on the data.

Evaluation Setup

For a classification problem solving, accuracy score is one of the most important metrics to evaluate model performance. With imbalanced classes, it's easy to get a high accuracy without actually making useful predictions. So, accuracy as an evaluation metrics makes sense only if the class labels are uniformly distributed.

For an imbalanced class problem, precision and recall are more important metrics to measure efficiency of model. As stated in previous section, we don't want to misclassify an actual buyer as non-buyer, so recall is going to be the most significant metric here. In the process of cost and resource optimization, we don't mind placing a few calls where the person receiving the call does not buy the term deposit after talking to the salesperson. From the EDA below we can see that, an unsuccessful call may lead to a successful call in future. So, improving precision is the secondary objective here.



So, instead of F1 score evaluation, our focus will be to improve F-beta score of positive class with more weightage on recall [3].

Below is our evaluation metric.

$$F_\beta = (1+\beta^2)\frac{precision*recall}{(\beta^2*precision)+recall} \quad \beta = 2$$

Approach Description, Result Presentation and Analysis:

We have built two classifiers -random forest and neural network, to solve this problem.

Below are some details of the models we built-

Random Forest-

- The classifier has been instantiated with parameter “class_weight=’balanced’” to provide more weightage to ‘purchased’ class.
- The classifier has been trained on multiple values of each hyper-parameters (e.g. max_depth, max_features,, n_estimators etc.) and GridSearchCV gave us the best set of hyper-params.
- We have used 3-fold cross validation to fight against overfitting.
- Once the best model is obtained, it is fitted on train data to get predicted result on test dataset.

Cross validation and hypermeter tuning tries to immune the model from overfitting.

Neural network-

- We have built a 3-layer neural network where first two activation functions are Relu and the output layer activation function is sigmoid. Sigmoid returns all the predicted value in the range [0,1]. We have used round function on the output to predict the target variable 0(not purchased term deposit) or 1(purchased term deposit).
- To deal with imbalance class problem, we have assigned weightage 10 on each data of purchased class and weight 1 on each data of not purchased class.
- The model aims to reduce binary cross-entropy loss with help of ‘adam’ optimizer.
- The model fits on train dataset and calculates loss on a holdout dataset(validation_data) after each epoch.
- ‘Early stopping’ method is applied to reduce the effect of overfitting. The model stops iterating if validation loss does not decrease in 200 epochs.
- This model compares validation loss with previous epoch’s validation loss and whenever the loss reduces, it saves the model.
  Once we get the best model, we fit this on training data and get prediction on test data.

Dealing with null-hypothesis 1 and 2:

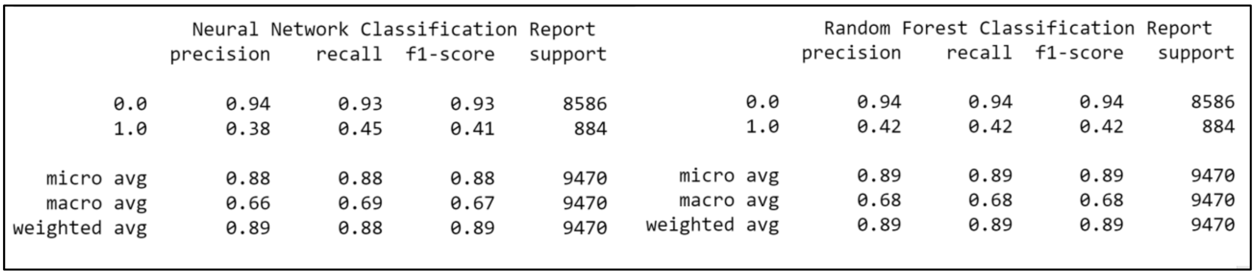
In order to reject both the null-hypotheses we need to show p-value is less than 0.05. First, we have created a baseline model which predicts each output as majority value of target, in our case the baseline model predicts all the data as ‘not purchased’. Next, we created a list (with value 0 and 1) for each of the three models (baseline, random forest and neural network), where 1 indicates the model predicted the output correctly and 0 indicates model misclassified the data point.

Once the lists are prepared, we used chi-squared test and found that both these models work better than baseline model as p-values are 4.36e-14 and 8.5e-06 for neural network and random forest respectively.

We used chi-squared test to compare between random forest and neural network as well and found that p-value is less than 0.05. This rejects the 2nd null hypothesis, which claims both the model works equally well on test data.

Since, the models are not performing same, we will choose the model to be the final one which gives better F-beta score.

Below is classification report of two models on test dataset.

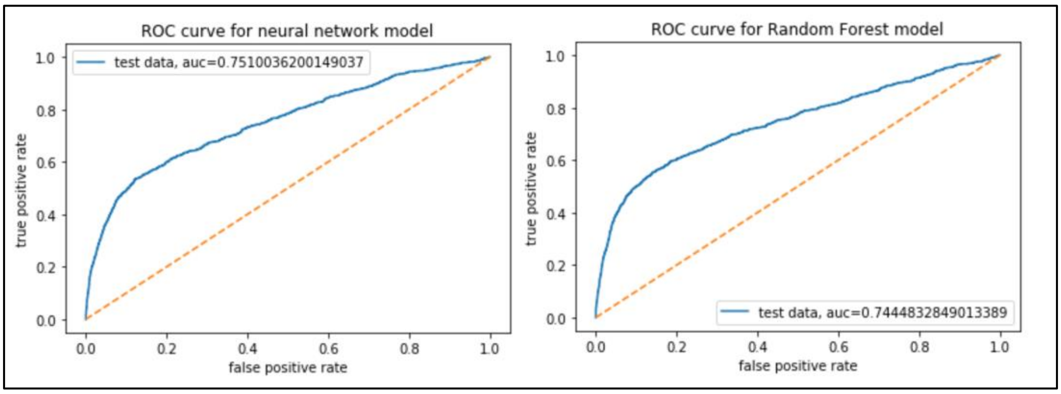


```
                  Neural Network Classification Report                          Random Forest Classification Report
                  precision    recall  f1-score   support                        precision    recall  f1-score   support

         0.0           0.94      0.93      0.93      8586              0.0           0.94      0.94      0.94      8586
         1.0           0.38      0.45      0.41       884              1.0           0.42      0.42      0.42       884

   micro avg           0.88      0.88      0.88      9470        micro avg           0.89      0.89      0.89      9470
   macro avg           0.66      0.69      0.67      9470        macro avg           0.68      0.68      0.68      9470
weighted avg           0.89      0.88      0.89      9470     weighted avg           0.89      0.89      0.89      9470
```

F-beta score of neural network $\frac{5*0.38*0.45}{4*0.38+0.45} = 0.43$, F-beta score of random forest $\frac{5*0.42*0.42}{4*0.42+0.42} = 0.42$.

Neural network provides slightly better F-beta score.

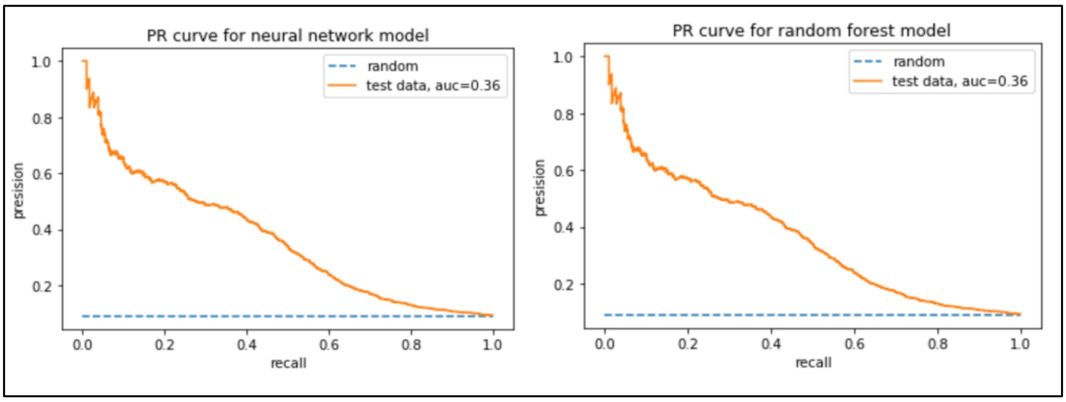
Below is the ROC curve of the two models on test dataset.



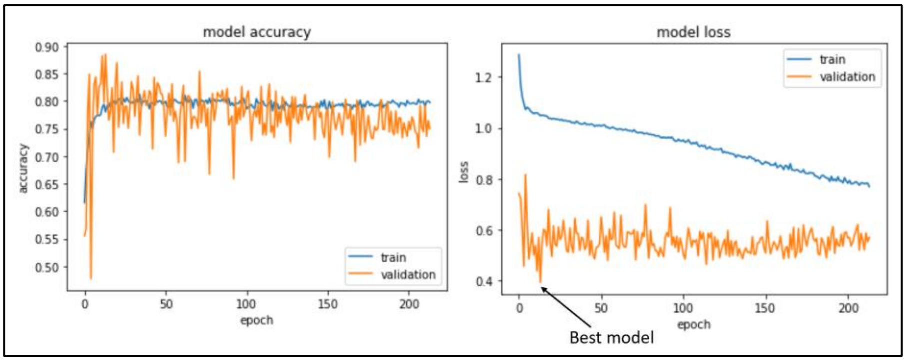
Neural model provides a little better result. But we are not going to consider ROC curve as a metric of model performance for this problem. ROC plot can be misleading when applied to strongly imbalanced datasets [1]. ROC curves appeared to be identical under balanced and imbalanced cases. For imbalanced class problem, precision-recall curve offers better insight [2]. The PRC plot shows the relationship between precision and recall, and its baseline moves with class distribution.

For binary imbalance class problem, if ratio of number of negative points to positive points is 10:1, baseline/random auc of PR curve becomes 0.09 [2]

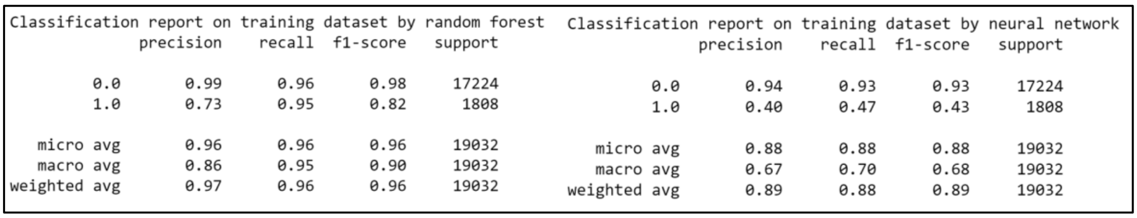
(random = Num of positive data/ (num of positive data+ number of negative data)).



We can see both the models offer similar performance and performs better than baseline model. So, there are very little difference between these two models and whichever model can provide better robust prediction should be the final one. Below loss and accuracy data was captured on holdout dataset while training the neural network.



We can see the neural network model was selected before it reached the point of overfitting. That's the reason this model gives similar result on all the datasets. Whereas, in-spite of using cross validation and hyper-parameter tuning, random forest suffers from overfitting. We can visualize the difference between these two models from the classification report on train data set.



```
Classification report on training dataset by random forest
              precision    recall  f1-score   support

         0.0       0.99      0.96      0.98     17224
         1.0       0.73      0.95      0.82      1808

   micro avg       0.96      0.96      0.96     19032
   macro avg       0.86      0.95      0.90     19032
weighted avg       0.97      0.96      0.96     19032
```

```
Classification report on training dataset by neural network
              precision    recall  f1-score   support

         0.0       0.94      0.93      0.93     17224
         1.0       0.40      0.47      0.43      1808

   micro avg       0.88      0.88      0.88     19032
   macro avg       0.67      0.70      0.68     19032
weighted avg       0.89      0.88      0.89     19032
```

Comparing this report with test dataset report (provided in page 3), we can say neural network provides more robust and reliable result across all the datasets.

So, we are going to choose neural network as our final model for reliability and better F-beta value of positive class.

Conclusion

As stated in the research problem, without implementing any model it took around 11 calls to find out one buyer. Now, after implementing the model, it takes around 2.6 calls(1/precision) to find a potential buyer. However, in this process, model mis-classifies around half(1/recall) of the buyers as non-buyer. As of now, if the predicted value of a data point is below 0.5, we classify them as non-buyer. In case business wants to reduce number of mis-classified buyers, the threshold value can be changed from 0.5 to a lower fraction, so that recall increases further and precision decreases.

We have discarded random forest for less F-beta score and overfitting issue, but, random forest can give us feature importance, which can be very useful to find out more buyers. So, by improving random forest model, we can provide some meaningful insights to the business.

This dataset had a lot of missing values in categorical columns and removing all these missing values would significantly reduce the size of dataset. Also, in real world scenario, we can get information about some people whose some information (for e.g. let's say age is unknown) are unknown to us. My model should be able to predict these people's chances of buying term deposit without that particular information. So, instead of removing these data, they had been replaced by a different category 'others'. These undefined 'others' data obscure the basic underlying pattern of the dataset. If we could have a clean dataset with lesser number of unknown values, the model could work better.

References:-

1. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4349800/
2. https://classeval.wordpress.com/simulation-analysis/roc-and-precision-recall-with-imbalanced-datasets/
3. http://www.marcelonet.com/snippets/machine-learning/evaluation-metrix/f-beta-score