Data Mining Project Report

(MSIS 672 OL)

Group Members:

Abhishek Cholkar

02073868

1) Introduction:

This project utilizes machine learning techniques to develop a predictive model for improving the effectiveness of a software company's direct mailing campaign. The model will be trained on the catelog.csv dataset which contains historical data on past mailing responses and software purchases. The target variable for prediction is the "Purchase" column, representing whether a prospect responded positively to a test mailing by making a purchase. Additional data is provided about each prospect, including attributes and response history. For the purpose of this project, I have used various data mining techniques to mine the catelog.csv data. I will build a decision tree model to predict Purchase and then compared the prediction accuracy with a neural network model and also refine the model using GridSearchCV to predict Purchase.

2) Data exploration and data preparation:

Preparing the data means transforming the data to build better predictive model. This attest for good data quality and involved the following steps:

- Replacing Null values: 'last_update_days_ago' and '1st_update_days_ago' have been replaced by mean values to prepare the data for model building.
- Dropping Null values for Web_Order as we cannot replace a categorical variable by mean or median.
- Dropping columns: Sequence Number. This is because they are irrelevant in the analysis.

3) Data Analysis:

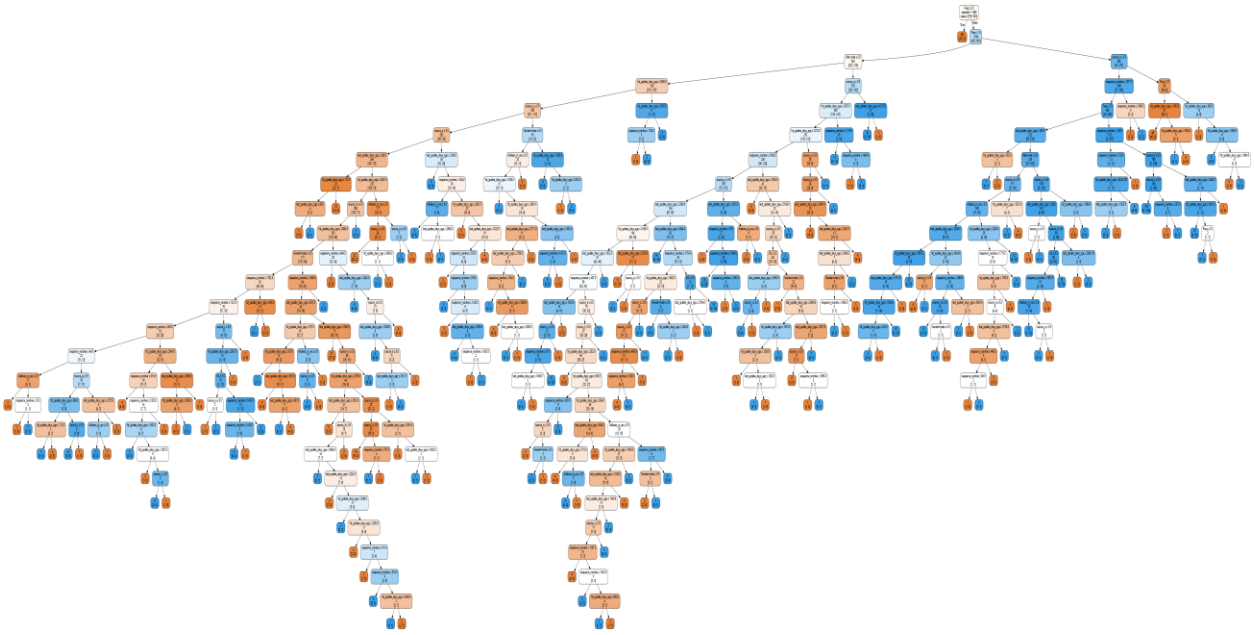
The dataset after the pre-processing were further explored using classification and prediction model to understand the relationships in the data as well as generate predictions from them:

- Classification model using Decision Tree and Neural Network for the Purchase variable. They are used to predict the categorical class label.

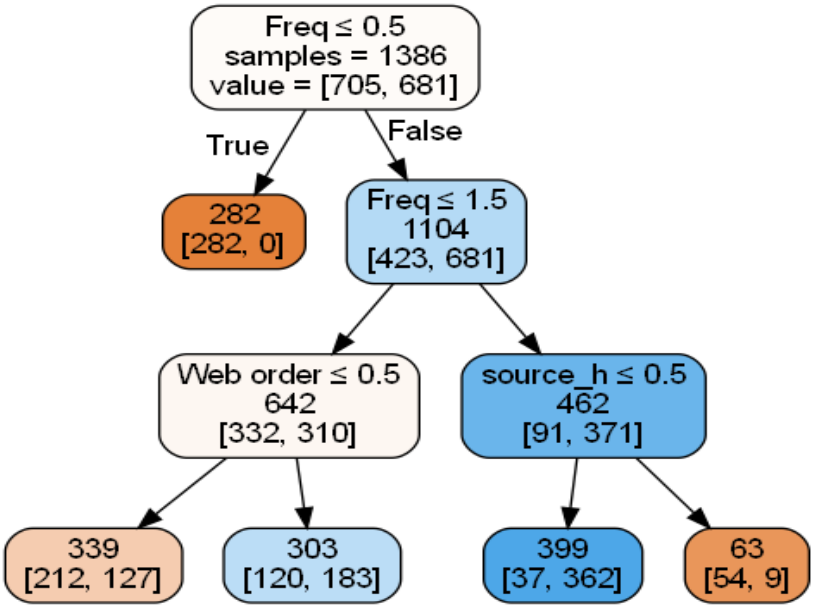
- Both models were separated into training and validation sets and then training a portion of the data so that its predictions can be compared on the validating data thereby measuring the predictive accuracy of the model as well as checking for overfitting.

Purchase Outcome:

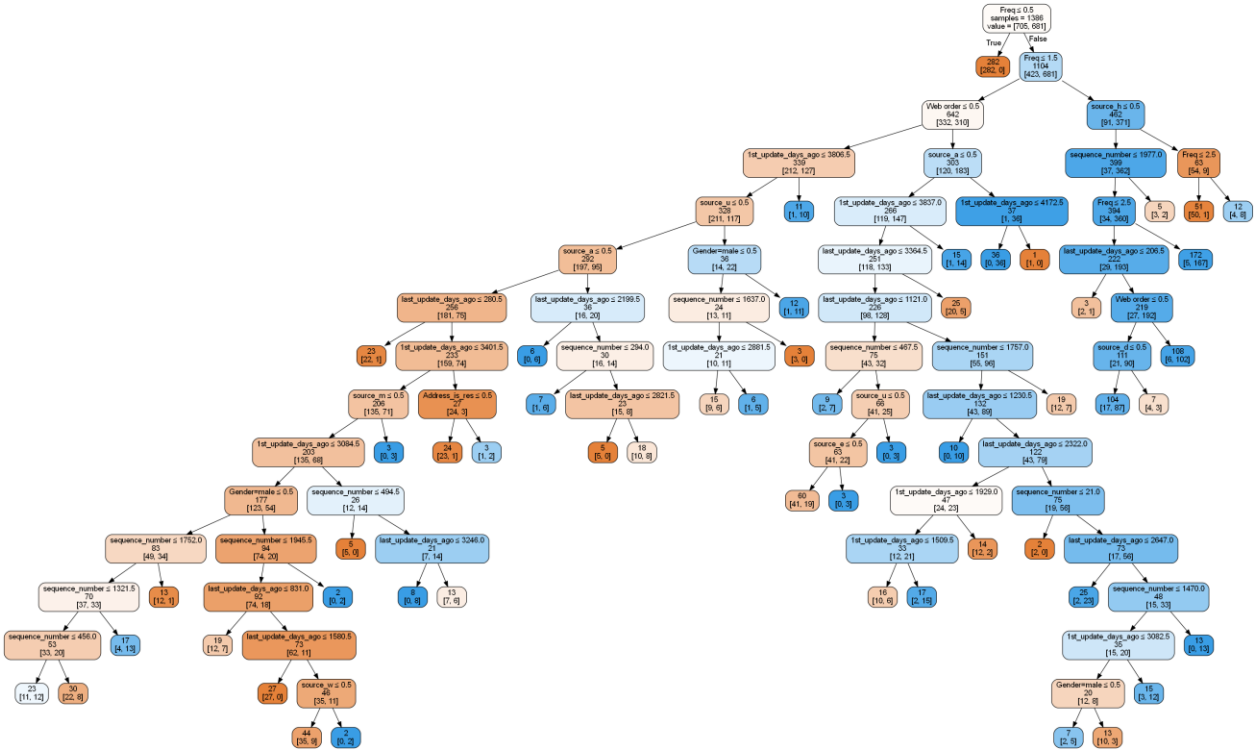
Full Decision Tree:



Small Tree:



Improved Tree after tuning the Hyperparameters:



Neural Networks Outcome:

```
Hidden layer 23 => 4
 Intercepts:
  [-0.22918694  0.04297526 -0.0476927  -0.09787315]
 Weights:
Output layer 4 => 1
 Intercepts:
  [-4.0357421]
 Weights:

In [151]: for weight in weights:
     ...:     print(' ', weight)
     ...: print()
  [6.09287374]
  [-4.10221572]
  [0.6303654]
  [4.60418108]
```

Assess and Compare Both Model's Performance

Performance on the training data vs. on the validation data

Decision Tree:

Model Performance on Training Data. Accuracy = 1.00		
Actual	Prediction	
	0	1
0	705	0
1	0	681

Model Performance on validation Data. Accuracy = 0.7933		
Actual	Prediction	
	0	1
0	227	58
1	65	245

Small Class Tree

Model Performance on Training Data. Accuracy = 0.7886		
Actual	Prediction	
	0	1
0	548	157
1	136	545

Model Performance on validation Data. Accuracy = 0.7664		
Actual	Prediction	
	0	1
0	220	65
1	74	236

Refined Tree:

Model Performance on Training Data. Accuracy = 0.8846		
Actual	Prediction	
	0	1
0	641	64
1	96	585

Model Performance on validation Data. Accuracy = 0.8432		
Actual	Prediction	
	0	1
0	240	45
1	62	248

Neural Networks:

Model Performance on Training Data. Accuracy = 0.7915		
Actual	Prediction	
	0	1
0	539	166
1	123	558

Model Performance on validation Data. Accuracy = 0.7748		
Actual	Prediction	
	0	1
0	220	65
1	69	241

Refined Neural Network:

Model Performance on Training Data. Accuracy = 0.8110		
Actual	Prediction	
	0	1
0	541	164
1	98	583

Model Performance on validation Data. Accuracy = 0.8084		
Actual	Prediction	
	0	1
0	221	64
1	50	260

4.) Overfitting concerns:

While Decision Tree is simple and easy to implement it does have overfitting concerns as the accuracy suggests that it is just memorizing data and there is a clear gap in the validation accuracy. As in the real world, attaining 1.00 score is not possible and suggests overfitting of the Model. Neural Networks model can learn complex non-linear relationships and interactions between variables. Helpful for pattern recognition from large feature space. We can fine tune hyper-parameters and run the Neural Network model. Exhaustive Grid search is a technique used to fine tune the parameters used in the model to improve the model accuracy. It is commonly used for finding the best combination of hyperparameters for a machine learning model. It is widely used to address the stopping point for tree growth especially relying on simple stopping rules, such as maximum tree depth, might not be effective, and a more systematic approach is needed. GridsearchCV() is the function in sci-kit learn which is used to find the best combinations. As mentioned in the text, there are different methods to improve model accuracy and prevent overfitting and tree growth. In exhaustive search there are different parameters like depth, impurity, sample size etc. are used and a 5-fold cross validation technique is used to improve the performance. The splits are taken such that it does not make the sample size too little or too high to prevent underfitting or overfitting.

5) Findings and Conclusion:

The report suggests that the model accuracy derived from the neural network and decision tree classifier methods differed although they all share a number of similarities. Neural network performance accuracy for both the training and validation dataset was 81% and 80% respectively while the Decision Tree performance accuracy measured was 88% and 84% for both training and validation dataset respectively. This suggests that while the decision tree has a higher percentage of accuracy, It has an overfitting issue unlike the Neural network which has a lower accuracy but no issue with overfitting.

Recommendation

To improve the cost-effectiveness of planning a mailing campaign, management should focus on the pointer which plays significant role in affecting the Purchase. Neural networks can be used as the best classification model for analysis.