Banking Campaign Output Prediction

Introduction

The project aims to predict where a bank client wants to subscribe bank term deposit based on the bank's marketing campaigning done through phone calls. The dataset consists of many attributes related to the client information, information on the last contact of the current campaign, social and economic context information, subscription, and other related information. In this project, the required data was preprocessed and with the help of neural networks, a machine learning model was built to predict whether the client would subscribe to the term deposit or not.

Data Processing

Features for the model: Column 'duration' is removed. This attribute highly affects the output target (e.g., if duration=0 then y='no'). Yet, the duration is not known before a call is performed. Also, after the end of the call y is obviously known. All other features are used for training the model.

In this section, numerical and categorical variables were separately preprocessed.

Categorical data:

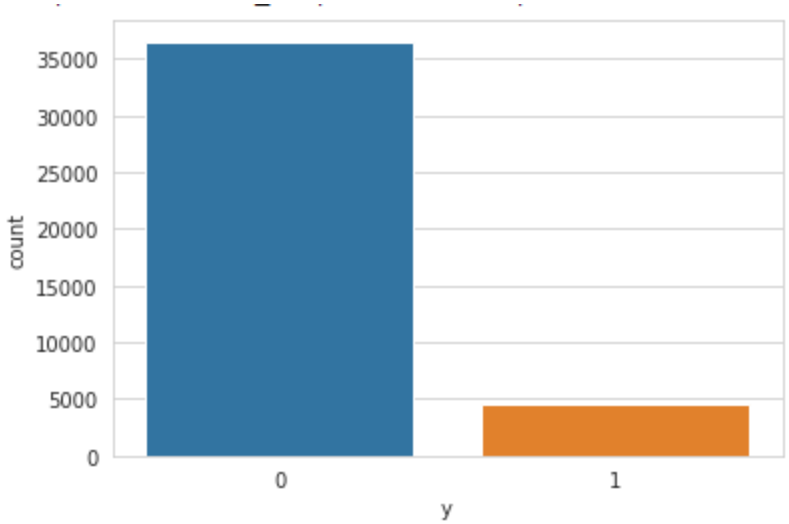
It is observed that there are a lot of unknown variables only in the categorical data. So the unknown values are first imputed by the most frequently occurring values using mode().
After filing all the unknown values, the categorical data is encoded using pd.get_dummies() to use them further as features for the model. This function converts the categorical columns into indicator columns of 0's and 1's.
Further scaling of categorical data is not required as its value is already encoded in the form of 0s and 1s.

Numerical data:

Numerical data is further scaled with StandardScaler(). This standardizes the features by removing the mean and scaling to unit variance

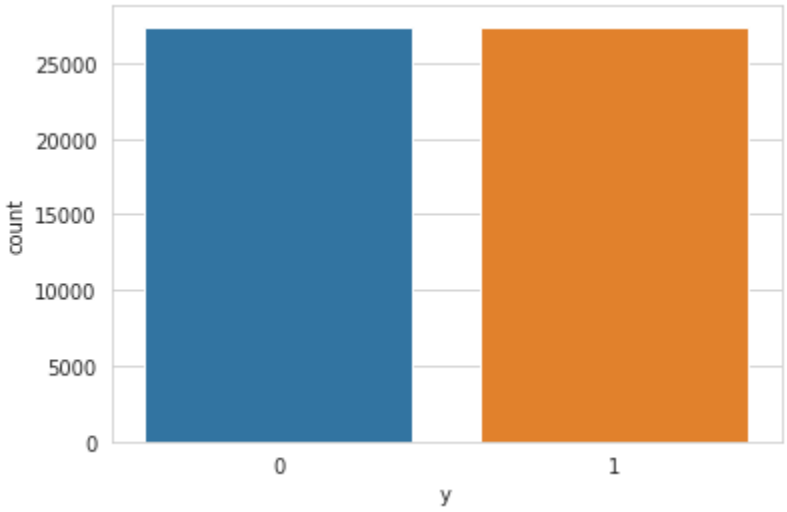
Data Balancing:

Before a model is built and trained over the data, it is important to note the imbalance of the classes in the target variable.



This imbalance of data can result in poor performance of the model on the minority class. One way to solve this poor imbalance is to oversample the minority class by duplicating the examples of the minority class. Though duplicating the data would not provide any new information to the model, it can help the model learn the decision boundary of the classes effectively.

In this project, SMOTE (Synthetic Minority Oversampling Technique) is used for oversampling class 1 in the training dataset. SMOTE works by selecting examples that are close in the feature space, drawing a line between the examples in the feature space and drawing a new sample at a point along that line.



After oversampling

Modelling

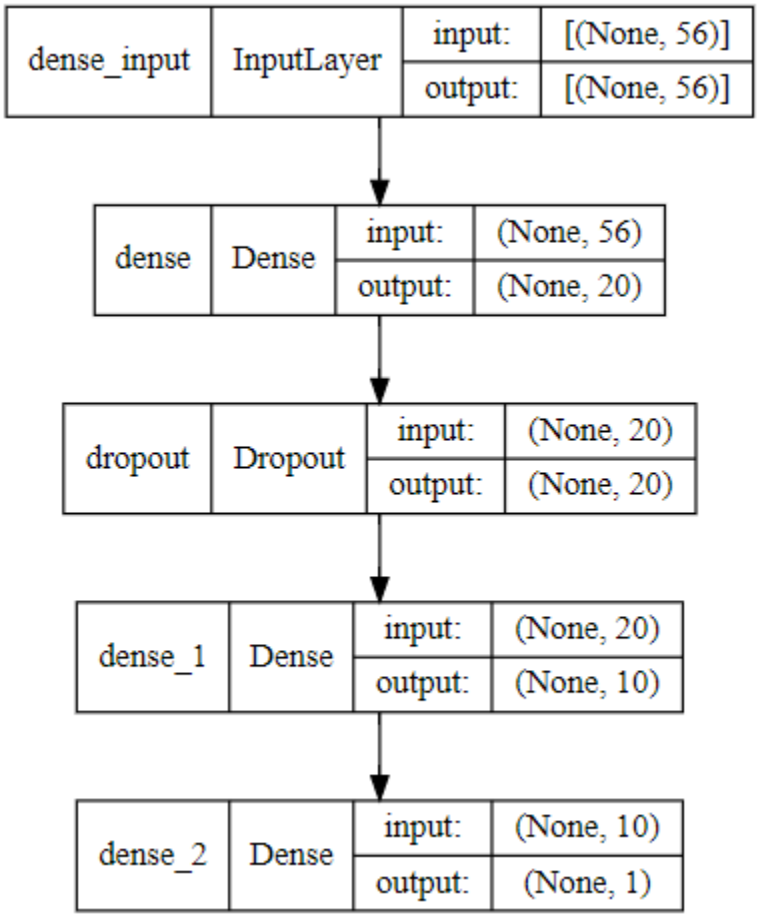
In this project, a fully connected neural network is created using Sequential model with multiple Dense layers.

There are total 5 layers in the sequential model including input and output layers. There are 3 Hidden layer. The input features are passed to the first hidden layer.

The activation functions used for hidden layers are ‘ReLu’ and ‘sigmoid’ for the output layer. Sigmoid predicts the output in the form of probabilities between the range of 0 and 1.

The Architecture of the model:

- The first dense layer(first hidden layer) has an input dimension of 56 (feature columns). The number of nodes is set low to 20 so that the best features of the data are used.
- The next layer Dropout is used because it is an effective regularization method to avoid overfitting and improve generalization. Dropout trains a large number of neural networks with different architectures in parallel and randomly drops out nodes during training.
- The next Dense model has 10 nodes that process the output from the 20 nodes of the previous layer and passes on information to its next layer through its 10 nodes.
- The final output layer takes the information from the 10 nodes of the previous layer and processes them in its 1 node using the sigmoid function.



Performance of the Model
Classification report:

	precision	recall	f1-score	support
0	0.93	0.95	0.94	9139
1	0.50	0.42	0.46	1158
accuracy			0.89	10297
macro avg	0.71	0.68	0.70	10297
weighted avg	0.88	0.89	0.88	10297

- ➢ The model over-all accuracy is 89%
- ➢ Proprtion of positive predictions made by model on class 0 is 93% and class 1 is 50%.
- ➢ The model can predict 95% of class 0 as class 0 and 42% of class 1 as class 1 (recall).

Confusion Matrix:

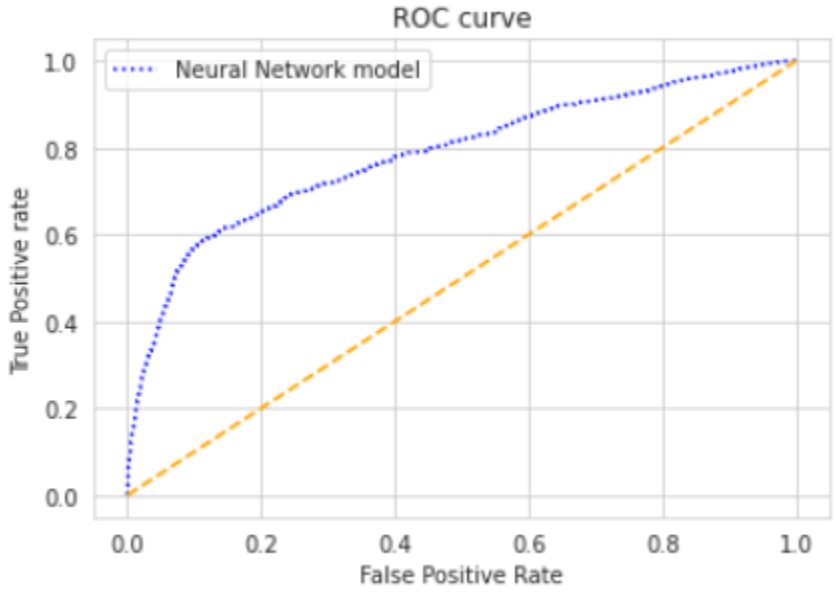
```
Confusion matrix :
[[8644  495]
 [ 669  489]]
```

- ➢ The model can correctly classified ‘class 0’ 8644 times and wrongly predicted ‘class 0’ as ‘class 1’ 669 times.
- ➢ The model can correctly classified ‘class 1’ 489 times and wrongly predicted ‘class 1’ as ‘class 0’ 495 times.

AUC-ROC Curve:

AUC- explains the degree of separatability of the model. How good can the model separate the classes. The auc score of the model is `0.78`.

ROC- probability curve



Scientific Bottlenecks:

- The most difficult problem with the project is to work with the imbalanced data. Imbalance in the classes of target variable can lead to poor performance of the model on the minority class. This problem was solved by using SMOTE to oversample the minority class.
- The precision and recall values of class 1 (minority class) was hard to achieve, as the available minority class data was still not good enough for the model to make separability. This problem was solved to an extent by parameter tuning in the network architecture. The number of nodes in the layers and the number of layers were tuned , Dropout layer was introduced to avoid overfitting.

Conclusion

The Sequential Dense model generated predicted the subscription of the clients well with an over-all accuracy of 89%. But the imbalance in data caused a low precision and recall score in minority. Overall, working on the project was a great experience with hands-on implementation on creating neural network and balancing the data.