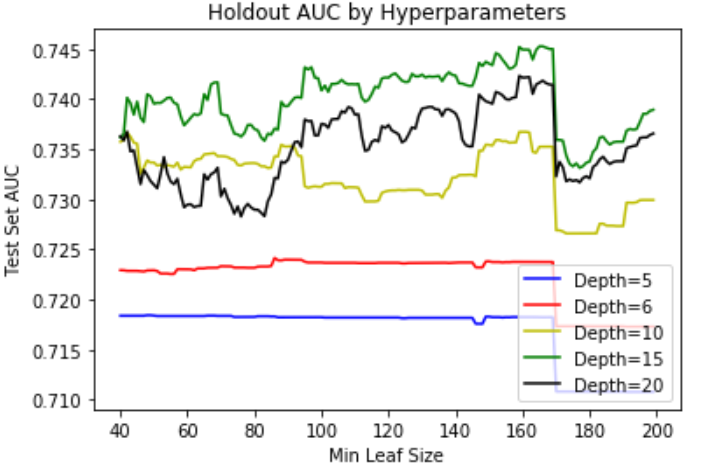
The most important feature is duration which makes sense because more contact duration would potentially lead to higher return value, but the econd feature which is employed which is hard to get during that quarter

Shuffle !!!!!!!!

Since the use of ID3 decision tree in classification doesn’t matter with the value of categorical variable. Feature “pdays” doesn’t necessary to be transformed.

The algorithms that have been implemented are decision tree and random forests. Decision trees often perform well on imbalanced datasets because their hierarchical structure allows them to learn signals from both classes (y = 0 and y = 1 https://elitedatascience.com/imbalanced-classes). The biggest advantage is that it is easy to be interpreted. The decision tree makes explicit all possible alternatives and traces each alternative to its conclusion in a single view, allowing for easy comparison among the various alternatives. The use of separate nodes to denote user defined decisions, uncertainties, and end of process lends further clarity and transparency to the decision-making process. Also, it can perform feature selection and won’t be affected by non-linear relationships between parameters. However, it also has some limitations. It can create over-complex tree structure which is overfitting. It is unstable since small variations in dataset can result a completely different tree.

The procedure of performing decision tree was completed in Python Jupyter notebook. The baseline of decision tree using default parameters has AUC = 0.60062973168304579. To improve AUC, the first start point is to find a reasonable range of parameters in decision tree (minimal size for split, minimal leaf size, maximal depth of tree). The optimal combination of these three parameters results minimum overfitting and lowest error. In this project, AUC is used for all models because it is a good evaluation metrics dealing with unbalanced dataset (https://stats.stackexchange.com/questions/132547/roc-curves-for-unbalanced-datasets). By involving all features in a single decision tree with an initial range of depth and minimum leaf size, figure XXX clearly indicates that the range of depths resulting best AUC is from 10 to 20. The range of leaf size is from 100 to 160.

Next, to see the relationship between minimal size for split and minimal leaf size, depth = 15 is fixed.

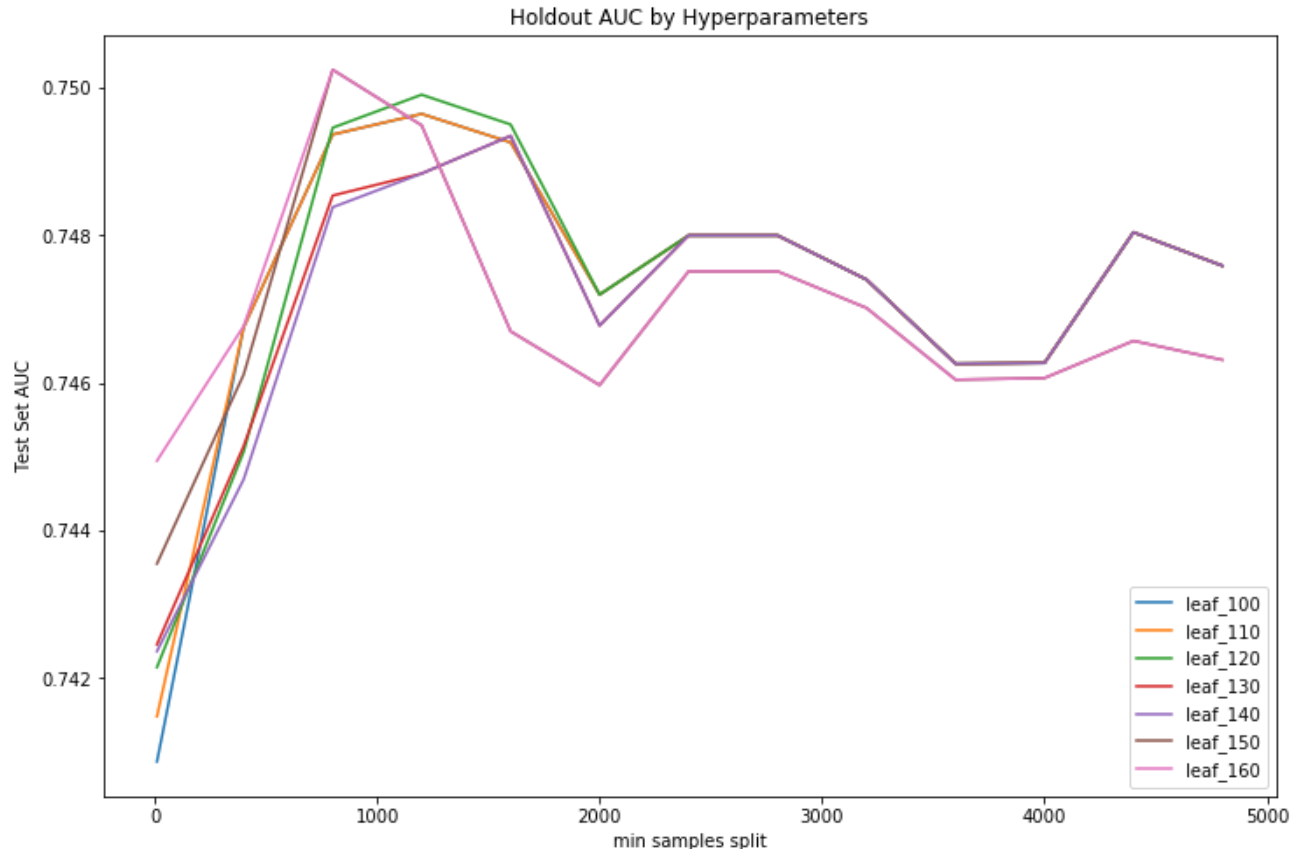
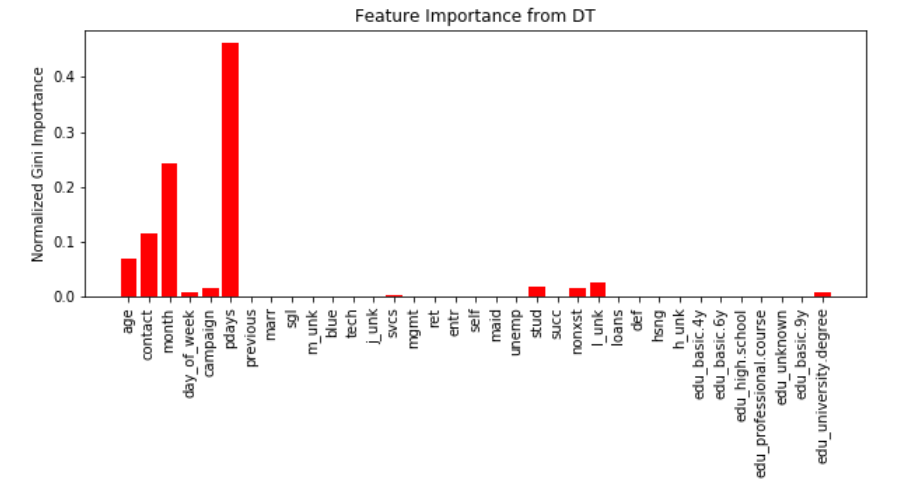


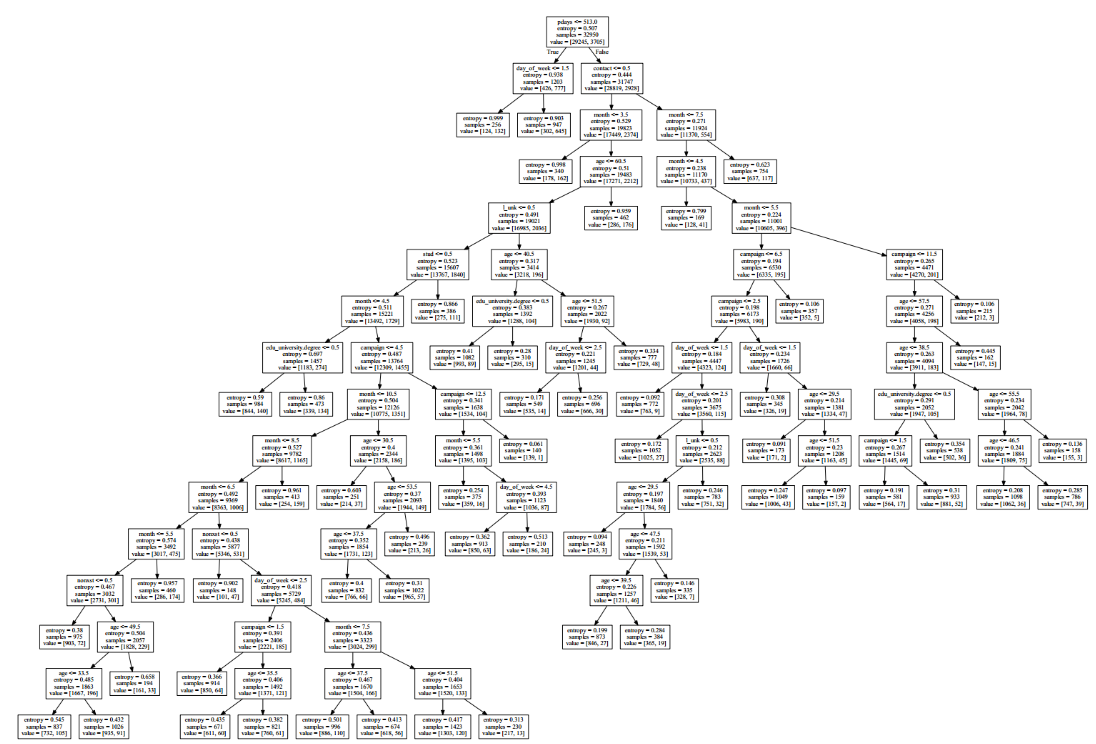
Figure XXX shows that AUCs in the range of splits between 800 to 1200 are close to peak.

After obtaining the ranges of these three parameters, the combination of 5-fold cross validation and function of grid\_search\_cv in Python gave the best combination, which is :

Then, inserting the optimal parameters into the decision tree model results feature importance plot, indicating that variables “pdays”, “month”, “contact”, and “age” are important features. Refitting the model by adding one feature at atime from most important to lowest to see the change of AUC. The best AUC below is obtained by including top 7 important features ('pdays', 'month', 'contact', 'age', 'l\_unk', 'stud' and 'campaign'). Although this AUC is less than 0.7655 from the model with all features included, it is simpler and more explainable.

The refitting procedure stops at 10 features because the rest features are not relatively important and won’t improve AUC a lot if they are included. Also, more features included would make the model complex.

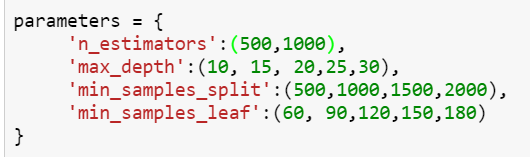
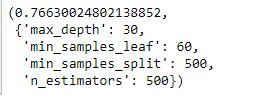
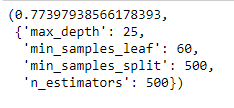
Below is the view of final decision tree model.



In modern applied machine learning, tree ensembles (Random Forests, Gradient Boosted Trees, etc.) almost always outperform singular decision trees. Compared to decision tree that only has a single tree, random forest can develop various trees with the number of k features selected out of the number of p features from the original dataset. Therefore, it doesn’t require to do feature selection and has a more convinced result in feature importance. It can also reduce the variance for unbalanced dataset. Nevertheless, Random forests have been observed to overfit for some datasets with noisy classification/regression tasks (http://rstudio-pubs-static.s3.amazonaws.com/4239\_fcb292ade17648b097a9806fbe026e74.html). Unlike decision trees, random forest models are hard to interpret since the result is a combination of various trees.

Unlike decision tree, there are two additional parameters for random forests needed to be tuned, which are “number of trees” and “number of predictors sampled”. Typically, there are a total of 35 predictors, square root of 35, which is 6 predictors in the case of classification make a good choice (<https://sadanand-singh.github.io/posts/treebasedmodels/>).

The baseline AUC for random forest by using default set in Python is 0. 0.73387246176027576.

By choosing the optimal combination parameters from the dictionary below, Python outputs the result, which is max\_depth =30, min\_samples\_leaf = 60, min\_samples\_split = 500, n\_estimators = 500, AUC is 0.7663, which is higher than the optimal AUC from decision tree. Then, by setting max\_feature ranged from 5 to 9, which means the number of features to consider when looking for the best split is from 5 to 9. (default was 6 resulting from the square root of 35 features). AUC is highest when max\_feature is 9. The relative feature importance graph suggests that the top 9 important features are 'pdays', 'month', 'succ', 'contact', 'age', 'previous', 'nonxst', 'l\_unk' and 'campaign'. This AUC is optimal and the model with 9 features included is not complex but it is difficult to interpret the model itself.