Kaggle Titanic Survival prediction

Step 1 : Data preprocessing

Standard steps

* Create a dataframe on csv
* Df.info() to know the data types , number of columns and name
* Drop columns that are unimportant like name etc
* Object data types are non-numeric . So Quantify all categorical variables ( more details below..Use pd.read\_dummies() to create new df and drop redundant fields from concatenated dataframes (eg. Using columnisation – factorize row values to column header)
* Dealing with missing values using median or interpolate()
* Make sure train and test data are in same dimensions after preprocessing

Type of problem:

Binary Classification problem (ie. survived 1 or not)

Models :

1.Decision Tree from sklearn import tree , tree.DecisionTreeClassifier

Decision trees compute entropy in the information system. If you peform a decision tree on dataset, the variable **importances\_** contains important information on what columns of data has large variances thus contributing to the decision

By specifying maximum depth we can avoid overfitting

2.RandomForest classifier from sklearn.ensemble

Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction. One big advantage of random forest is, that it can be used for both classification and regression problems, which form the majority of current machine learning systems. Random Forest has nearly the same hyperparameters as a decision tree or a bagging classifier. Random Forest prevents overfitting most of the time, by creating random subsets of the features and building smaller trees using these subsets. Afterwards, it combines the subtrees.

Important Hyperparameters:

The parameters in random forest are either used to increase the predictive power of the model or to make the model faster.

1. Increasing the Predictive Power

Firstly, there is the “n\_estimators“ hyperparameter, which is just the number of trees the algorithm builds before taking the maximum voting or taking averages of predictions. In general, a higher number of trees increases the performance and makes the predictions more stable, but it also slows down the computation.

Another important hyperparameter is „max\_features“, which is the maximum number of features Random Forest is allowed to try in an individual tree.

The last important hyper-parameter we will talk about in terms of speed, is „min\_sample\_leaf “. This determines, like its name already says, the minimum number of leafs that are required to split an internal node

2. Increasing the Models Speed

The “n\_jobs“ hyperparameter tells the engine how many processors it is allowed to use. If it has a value of 1, it can only use one processor. A value of “-1” means that there is no limit.

“random\_state“ makes the model’s output replicable. The model will always produce the same results when it has a definite value of random\_state and if it has been given the same parameters and the same training data.

Lastly, there is the “oob\_score“ (also called oob sampling), which is a random forest cross validation method. In this sampling, about one-third of the data is not used to train the model and can be used to evaluate its performance. These samples are called the out of bag samples. It is very similar to the leave-one-out cross-validation method, but almost no additional computational burden goes along with it.

Advantage:

Feature Importance:

Another great quality of the random forest algorithm is that it is very easy to measure the relative importance of each feature on the prediction. Sklearn provides a great tool for this, that measures a features importance by looking at how much the tree nodes, which use that feature, reduce impurity across all trees in the forest.

Limitations :

A more accurate prediction requires more trees, which results in a slower model. In most real-world applications the random forest algorithm is fast enough, but there can certainly be situations where run-time performance is important and other approaches would be preferred.

And of course Random Forest is a predictive modeling tool and not a descriptive tool. That means, if you are looking for a description of the relationships in your data, other approaches would be preferred