Random Forest

December 13, 2019

```
[43]: from sklearn.model_selection import train_test_split, cross_val_score, KFold,␣
       ↪StratifiedKFold
      from sklearn.metrics import roc_auc_score, auc, roc_curve
      from sklearn.ensemble import RandomForestClassifier
      import pandas as pd
      import matplotlib.pyplot as plt
      %matplotlib inline
      from sklearn import metrics
      import numpy as np
```

```
[7]: train_df = pd.read_csv('train.csv')
     test_df = pd.read_csv('test.csv')
```

```
[55]: act_test_df = pd.read_csv('act_test.csv', dtype={'people_id': np.str,␣
       ↪'activity_id': np.str},
                                 parse_dates=['date'])
```

```
[56]: test_id = act_test_df.activity_id
```

```
[9]: X_train = train_df.drop(['outcome'], axis=1)
     Y_train = train_df['outcome']
```

```
[10]: # train, validation set split
```

```
[11]: x_train, x_val, y_train, y_val = train_test_split(X_train, Y_train, test_size =␣
       ↪0.5, random_state=1)
      x_train.shape, x_val.shape, y_train.shape, y_val.shape
```

```
[11]: ((1098645, 59), (1098646, 59), (1098645,), (1098646,))
```

```
[30]: random_forest =RandomForestClassifier(max_features = 10,n_estimators =␣
       ↪100,random_state = 1)
      random_forest.fit(x_train, y_train)
```

```
[30]: RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                             max_depth=None, max_features=10, max_leaf_nodes=None,
                             min_impurity_decrease=0.0, min_impurity_split=None,
                             min_samples_leaf=1, min_samples_split=2,
                             min_weight_fraction_leaf=0.0, n_estimators=100,
                             n_jobs=None, oob_score=False, random_state=1, verbose=0,
                             warm_start=False)
```

[31]:

```
random_forest.score(x_val, y_val)
```

[31]:

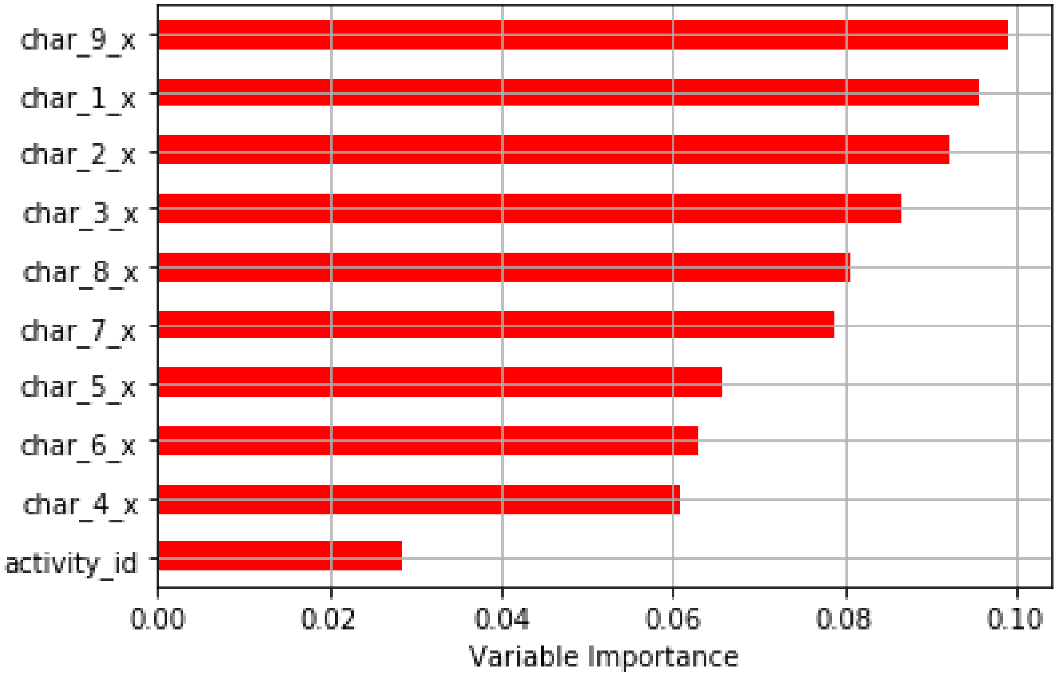
```
0.9906148113223003
```

[32]:

```
Importance = pd.DataFrame({'Importance':random_forest.feature_importances_*100},
                         index = x_val.columns)
```

[33]:

```
Importance.sort_values(by = 'Importance',axis = 0,ascending = True)[:10]\
                      .plot(kind = 'barh',color='r',)
plt.xlabel('Variable Importance')
plt.gca().legend_ = None
plt.grid()
```



[35]:

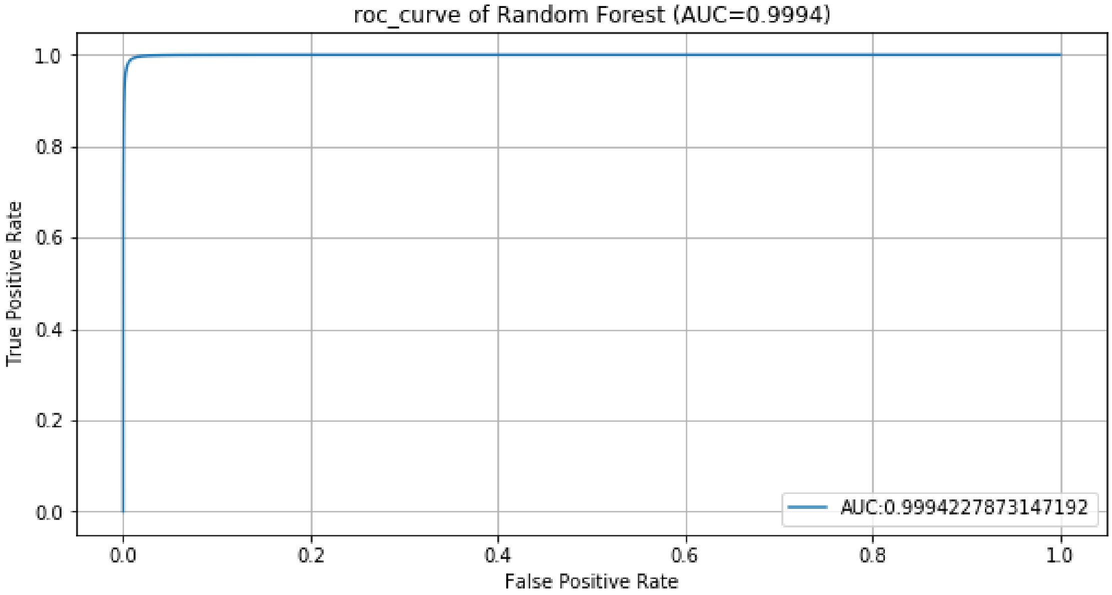
```
rf_predictions = random_forest.predict_proba(x_val)[::,1]
fpr, tpr, thresholds = metrics.roc_curve(y_val,rf_predictions)
rf_roc = pd.DataFrame()
rf_roc['fpr'] = fpr
rf_roc['threshold'] = thresholds
auc = metrics.roc_auc_score(y_val,rf_predictions)
auc
```

[35]:

```
0.9994227873147192
```

[37]:

```
plt.figure(figsize=(10,5))
plt.plot(fpr,tpr,label='AUC:'+str(auc))
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('roc_curve of Random Forest (AUC=%.4f)' %(auc))
plt.legend(loc=4)
plt.grid()
```



[39]:
```
Y_pred_rf= random_forest.predict(test_df)
```

[57]:
```
submission_rf = pd.DataFrame({'activity_id' : test_id, 'outcome': Y_pred_rf})
submission_rf.to_csv('submission_rf.csv', index = False)
```

[41]:
```
# kaggle score of Random Forest: 0.88931
```