KNN

December 13, 2019

[11]:
```
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
from sklearn.neighbors import KNeighborsClassifier
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn import metrics
import numpy as np
```

[2]:
```
train_df = pd.read_csv('train.csv')
test_df = pd.read_csv('test.csv')
```

[12]:
```
act_test_df = pd.read_csv('act_test.csv', dtype={'people_id': np.str,␣
↪'activity_id': np.str},
                          parse_dates=['date'])
```

[13]:
```
test_id = act_test_df.activity_id
```

[4]:
```
X_train = train_df.drop(['outcome'], axis=1)
Y_train = train_df['outcome']
```

[5]:
```
# train, validation set split
```

[6]:
```
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
```

[6]:
```
((1098645, 59), (1098646, 59), (1098645,), (1098646,))
```

[7]:
```
knn = KNeighborsClassifier(n_neighbors = 3)
knn.fit(x_train, y_train)
```

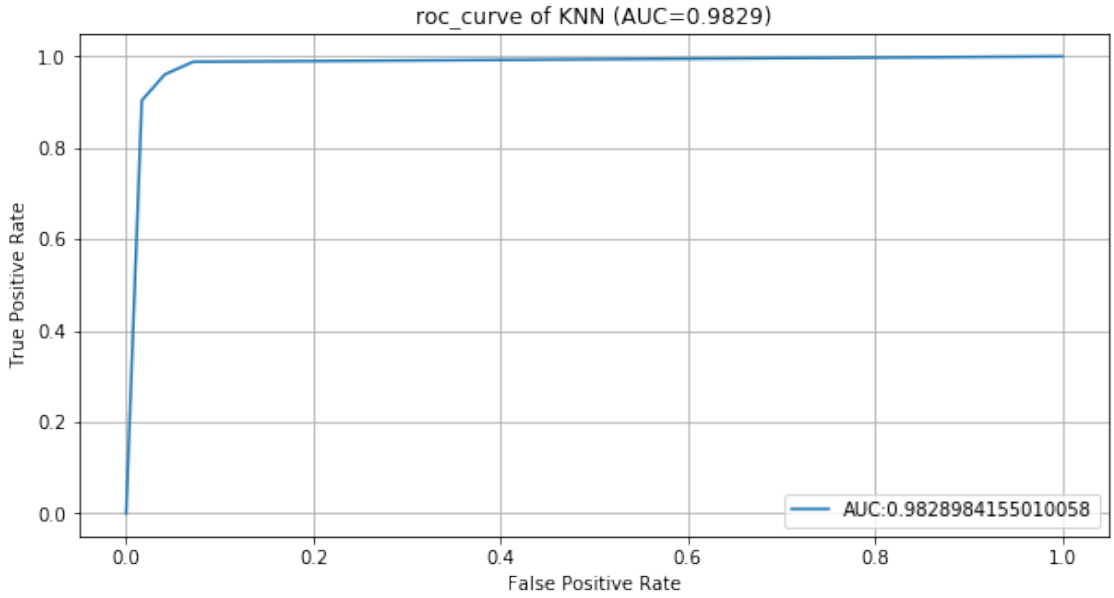
[7]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=3, p=2,
                     weights='uniform')
```

[17]:
```
knn_predictions = knn.predict_proba(x_val)[::,1]
fpr, tpr, thresholds = metrics.roc_curve(y_val,knn_predictions)
knn_roc = pd.DataFrame()
knn_roc ['fpr'] = fpr
```

```
knn_roc ['threshold'] = thresholds
auc = metrics.roc_auc_score(y_val,knn_predictions)
auc
```

[17]:
```
0.9828984155010058
```

[18]:
```
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 KNN (AUC=%.4f)' %(auc))
plt.legend(loc=4)
plt.grid()
```



[19]:
```
Y_pred_knn= knn.predict(test_df)
```

[20]:
```
submission_knn = pd.DataFrame({'activity_id' : test_id, 'outcome': Y_pred_knn})
submission_knn.to_csv('submission_knn.csv', index = False)
# kaggle score of KNN:0.83523
```