Ben Roth Week 5 - Jupyter Notebook

February 26, 2020

1 Assignment 5

1. Choose a regression dataset (bikeshare is allowed), perform a test/train split, and build a regression model (just like in assingnment 3), and calculate the
   - Training Error (MSE, MAE)
   - Testing Error (MSE, MAE)
2. Choose a classification dataset (not the adult.data set, The UCI repository has many datasets as well as Kaggle), perform test/train split and create a classification model (your choice but DecisionTree is fine). Calculate
   - Accuracy
   - Confusion Matrix
   - Classifcation Report
3. (Bonus) See if you can improve the classification model's performance with any tricks you can think of (modify features, remove features, polynomial features)

In [1]:
```
#load in relevant packages
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import (accuracy_score,
                             classification_report,
                             confusion_matrix,
                             auc,
                             roc_curve,
                             mean_squared_error,
                             mean_absolute_error
                            )
```

1.0.1 Question 1

In [2]:
```
#import data
congress = pd.read_csv("../data/congress.csv")
```

```
#dropna
congress = congress.dropna()

congress.head(5)
```

Out[2]:

	ChildPoverty	MedianIncome	Obama2008	GOP2party2010	GOPwin2010	WhitePct	Ideology
0	0.27	39597.0	0.39	1.000	1.0	0.68	0.460
1	0.29	37289.0	0.36	0.511	1.0	0.67	0.443
2	0.25	38079.0	0.43	0.595	1.0	0.65	0.426
3	0.26	35719.0	0.23	1.000	1.0	0.90	0.467
4	0.19	43832.0	0.38	0.580	1.0	0.78	0.797

In [3]:
```
#child poverty is rate in that district, median income is income in the district, obam
#voted for obama in that district, GOP2party2010 is % that voted for GOP congressional
#district that is white, Ideology is scale from -1 to 1, with most liberal to most con
X = congress[['ChildPoverty', 'MedianIncome', 'Obama2008', 'WhitePct', 'Ideology']]
y = congress['GOP2party2010']


#train/test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = .33)
```

In [4]:
```
#load model
linreg = LinearRegression()
#fitmodel
linreg.fit(X_train, y_train)
#predictions, training data
y_train_pred = linreg.predict(X_train)
#predictions, test data
y_test_pred = linreg.predict(X_test)

#training error measurements
train_mse = mean_squared_error(y_true = y_train, y_pred = y_train_pred)
train_mae = mean_absolute_error(y_true = y_train, y_pred = y_train_pred)

#test error measurements
test_mse = mean_squared_error(y_true = y_test, y_pred = y_test_pred)
test_mae = mean_absolute_error(y_true = y_test, y_pred = y_test_pred)

#print results
```

```
        print('The training MSE is: ', np.round(train_mse, 5))
        print('The training MAE is: ', np.round(train_mae, 5))
        print('The test MSE is:     ', np.round(test_mse, 5))
        print('The test MAE is:     ', np.round(test_mae, 5))
```

```
The training MSE is:  0.00763
The training MAE is:  0.05896
The test MSE is:      0.00906
The test MAE is:      0.06453
```

1.0.2 Question 2

In [5]:
```
#from kaggle.com
hr_data = pd.read_csv('../data/WA_Fn-UseC_-HR-Employee-Attrition.csv')

hr_data.head()
```

Out[5]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	...	RelationshipSatisfaction	StandardHours	StockOptionLevel	TotalWorkingYears	TrainingTimesLastYear	WorkLifeBalance	YearsAtCompany	YearsInCurrentRole	YearsSinceLastPromotion	YearsWithCurrManager
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	...	1	80	0	8	0	1	6	4	0	5
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	...	4	80	1	10	3	3	10	7	1	7
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	...	2	80	0	7	3	3	0	0	0	0
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	...	3	80	0	8	3	3	8	7	3	0
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	...	4	80	1	6	3	3	2	2	2	2

```
1                   7                   1                   7
2                   0                   0                   0
3                   7                   3                   0
4                   2                   2                   2

[5 rows x 35 columns]
```

In [6]:
```
#label encode string columns of interest
str_col = ['BusinessTravel', 'EducationField', 'Gender', 'JobRole', 'Attrition']
hr_data[str_col] = hr_data[str_col].apply(LabelEncoder().fit_transform)

#X matrix
X = hr_data[['Age', 'BusinessTravel', 'DailyRate', 'DistanceFromHome', 'Education', 'Ed
            'RelationshipSatisfaction', 'StandardHours', 'StockOptionLevel', 'TotalWorl
            'JobLevel', 'JobRole']]

#y array
y = hr_data['Attrition']

#train-test-split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = .33, random_state
```

In [7]:
```
#load in random forrest
rfmodel = RandomForestClassifier(random_state = 4321)

#train random forest
rfmodel.fit(X_train, y_train)

#predict test data
predictions = rfmodel.predict(X_test)

#accuracy of model
accuracy = accuracy_score(y_test, predictions)

#confusion matrix
conf_matrix = confusion_matrix(y_test, predictions)

#classification report
class_report = classification_report(y_test, predictions)

#print results
print('The accuracy score of the model is: ', round(accuracy, 4), '\n')

print('The confusion matrix of the model is:\n',
      conf_matrix, '\n')

print('The classification report is:\n', class_report)
```

```
The accuracy score of the model is:  0.8374
```

```
The confusion matrix of the model is:
 [[397  10]
 [ 69  10]]

The classification report is:
              precision    recall  f1-score   support

           0       0.85      0.98      0.91       407
           1       0.50      0.13      0.20        79

   micro avg       0.84      0.84      0.84       486
   macro avg       0.68      0.55      0.56       486
weighted avg       0.79      0.84      0.79       486



/Users/Broth/anaconda3/lib/python3.7/site-packages/sklearn/ensemble/forest.py:246: FutureWarnin
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
```

1.0.3 Question 3

In [8]:
```
#try to add additional variables to improve accuracy of the model
str_col = ['BusinessTravel', 'EducationField', 'Gender', 'JobRole', 'Attrition', 'Depa
          'OverTime']

hr_data[str_col] = hr_data[str_col].apply(LabelEncoder().fit_transform)


#X matrix
X2 = hr_data[['BusinessTravel', 'HourlyRate', 'DistanceFromHome', 'Education', 'Educati
             'StandardHours',  'TotalWorkingYears', 'Gender','JobLevel', 'JobRole', 'Ye
             'YearsSinceLastPromotion', 'WorkLifeBalance', 'MaritalStatus', 'OverTime'
            'Department', 'EnvironmentSatisfaction', 'StockOptionLevel', 'WorkLifeBala

#y array
y = hr_data['Attrition']

X_train2, X_test2, y_train2, y_test2 = train_test_split(X2, y, test_size = .33, random_
```

In [9]:
```
#load in random forrest
rfmodel2 = RandomForestClassifier(random_state = 1234)

#train random forest
rfmodel2.fit(X_train2, y_train2)

#predict test data
```

```
        predictions2 = rfmodel2.predict(X_test2)

        #accuracy of model
        accuracy2 = accuracy_score(y_test2, predictions2)

        #confusion matrix
        conf_matrix2 = confusion_matrix(y_test2, predictions2)

        #classification report
        class_report2 = classification_report(y_test2, predictions2)

        #print results
        print('The accuracy score of the model is: ', round(accuracy2, 4), '\n')

        print('The confusion matrix of the model is:\n',
              conf_matrix2, '\n')

        print('The classification report is:\n', class_report2)
```

```
The accuracy score of the model is:  0.8395

The confusion matrix of the model is:
 [[398   9]
 [ 69  10]]

The classification report is:
              precision    recall  f1-score   support

           0       0.85      0.98      0.91       407
           1       0.53      0.13      0.20        79

   micro avg       0.84      0.84      0.84       486
   macro avg       0.69      0.55      0.56       486
weighted avg       0.80      0.84      0.80       486



/Users/Broth/anaconda3/lib/python3.7/site-packages/sklearn/ensemble/forest.py:246: FutureWarnin
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
```

```
In [10]: print('The model improves by ', round(accuracy2 - accuracy, 5), ' from the first itera
```

```
The model improves by  0.00206  from the first iteration to the second iteration.
```