Quiz 2 – Feature Selection

Feature selection methods in Python

Tareq Haboukh

Instructions

Download the dataset (Dataset.csv)

Work in Python

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from sklearn.feature_selection import VarianceThreshold
from itertools import compress
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, accuracy_score, mean_squared_error, confusion_matrix

Dataset = pd.read_csv("Dataset.csv")
df = pd.DataFrame(Dataset)
```

In [3]:
```
def Logistic_Regression():

    y = df.loc[:, df.columns == 'gnd']
    x = df.loc[:, (df.columns != 'gnd') & (df.columns !='ID')]

    x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3)

    lr_model = LogisticRegression(solver='lbfgs', max_iter=1000)
    lr_model.fit(x_train, y_train.values.ravel())

    y_pred = lr_model.predict(x_test)

    accuracy = accuracy_score(y_test, y_pred)
    print("Accuracy of the logistic regression model on test set: %3f" %accuracy)
```

In [21]:
```
def Logistic_Regression_Variance_filter(VT):
    selector = VarianceThreshold(threshold=VT)
    selected_features = selector.fit_transform(df)
    selector.get_params()

    features_selected_VarianceThreshold = list(compress(df.columns, selector.get_support()))

    df_new = df.filter(features_selected_VarianceThreshold)
    y = df_new.loc[:, df_new.columns == 'gnd']
    x = df_new.loc[:, (df_new.columns != 'gnd') & (df_new.columns !='ID')]

    x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3)

    lr_model = LogisticRegression(solver='lbfgs', max_iter=1000)
    lr_model.fit(x_train, y_train.values.ravel())

    y_pred = lr_model.predict(x_test)

    # Model Accuracy
    accuracy = accuracy_score(y_test, y_pred)
    return(VT,accuracy)
```

In [5]:
```
# Low Variance Filter
def low_variance_filter(threshold):
    variance = df.var()
    features = df.columns
    selectedVariables = []
    for i in range(0, len(df.columns)):
        if variance[i] >= threshold:
            selectedVariables.append(features[i])
    return selectedVariables
```

In [6]:
```
df.rename(columns = {'Unnamed: 0' : 'ID'}, inplace = True)
```

1. Look at the shape of the dataset and print out the number of features. (gnd is the target variable)

In [7]:
```
df.shape
```

Out[7]:
```
(800, 258)
```

In [8]:
```
features_count = df.columns.str.contains("fea.").sum()
print("The dataset has {}".format(features_count) + " features")
```

```
The dataset has 256 features
```

2. Apply the Logistic Regression model on the given dataset and get the accuracy of the model.

In [9]:
```
Logistic_Regression()
```

```
Accuracy of the logistic regression model on test set: 0.991667
```

3. Remove low variance features set the threshold = 0.1, 0.2, 0.3, and 0.4

In [10]:
```
# Threshold = 0.1
Threshold_1 = low_variance_filter(0.1)

# Threshold = 0.2
Threshold_2 = low_variance_filter(0.2)

# Threshold = 0.3
Threshold_3 = low_variance_filter(0.3)

# Threshold = 0.4
Threshold_4 = low_variance_filter(0.4)
```

4. Print out the new datasets with selected features.

In [11]:
```
# Dataset with 0.1 Variance Threshold
df[Threshold_1].head(2)
```

Out[11]:

	ID	fea.5	fea.6	fea.7	fea.8	fea.9	fea.10	fea.11	fea.19	fea.20	...	fea.244	fea.245	fea.246	fea.247	f
0	1	-0.774192	-0.275283	0.034194	-0.399883	-0.815319	-0.955680	-0.989120	-0.987214	-0.844745	...	-0.183571	0.170864	0.365778	0.414430	0.4
1	2	-0.901515	-0.545999	0.030940	0.411866	0.390567	0.008434	-0.502208	-0.997706	-0.952953	...	-0.979200	-0.858993	-0.527557	-0.043773	0.3

2 rows × 209 columns

In [12]:
```
# Dataset with 0.2 Variance Threshold
df[Threshold_2].head(2)
```

Out[12]:

	ID	fea.6	fea.7	fea.20	fea.21	fea.22	fea.23	fea.24	fea.25	fea.26	...	fea.235	fea.236	fea.237	fea.245	fea.
0	1	-0.275283	0.034194	-0.844745	-0.336824	0.331032	0.504431	0.130807	-0.319797	-0.623297	...	0.522361	0.366751	-0.082735	0.170864	0.365
1	2	-0.545999	0.030940	-0.952953	-0.659959	0.035736	0.653449	0.839105	0.688151	0.494709	...	0.013770	-0.672177	-0.956167	-0.858993	-0.527

2 rows × 182 columns

In [13]:
```
# Dataset with 0.3 Variance Threshold
df[Threshold_3].head(2)
```

Out[13]:

	ID	fea.21	fea.22	fea.26	fea.27	fea.37	fea.38	fea.43	fea.44	fea.52	...	fea.220	fea.221	fea.228	fea.229	fe
0	1	-0.336824	0.331032	-0.623297	-0.835472	0.235636	0.540066	-0.274300	-0.644889	-0.081776	...	0.505857	0.495046	0.339861	0.368924	0.3
1	2	-0.659959	0.035736	0.494709	0.140118	-0.264202	0.538892	0.384775	-0.081463	-0.657173	...	-0.221164	-0.808547	-0.866280	-0.422491	0.1

2 rows × 133 columns

In [14]:
```
# Dataset with 0.4 Variance Threshold
df[Threshold_4].head(2)
```

Out[14]:

	ID	fea.43	fea.53	fea.54	fea.60	fea.69	fea.70	fea.72	fea.76	fea.85	...	fea.203	fea.204	fea.205	fea.212	fea.2
0	1	-0.274300	0.537992	0.294486	0.015796	0.511609	-0.166338	-0.920920	0.186746	0.241829	...	-0.582717	-0.143460	0.405317	-0.050328	-0.4303
1	2	0.384775	0.166035	0.792276	0.293105	0.484311	0.875532	-0.293676	0.413152	0.618632	...	0.796826	0.345453	-0.523125	-0.599433	0.1868

2 rows × 52 columns

5. Apply the Logistic Regression model on each of these datasets and get the accuracy scores.

In [59]:
```
yaxis = []
xaxis = []

variance_list = [0.1, 0.2, 0.3, 0.4]

for i in variance_list:
    x,y = Logistic_Regression_Variance_filter(i)
    xaxis.append(x)
    yaxis.append(y)

for i in range(len(xaxis)):
    print("Variance: {} Model Accuracy: {}".format(xaxis[i],yaxis[i].round(3)))
```

```
Variance: 0.1 Model Accuracy: 0.988
Variance: 0.2 Model Accuracy: 0.988
Variance: 0.3 Model Accuracy: 0.979
Variance: 0.4 Model Accuracy: 0.958
```

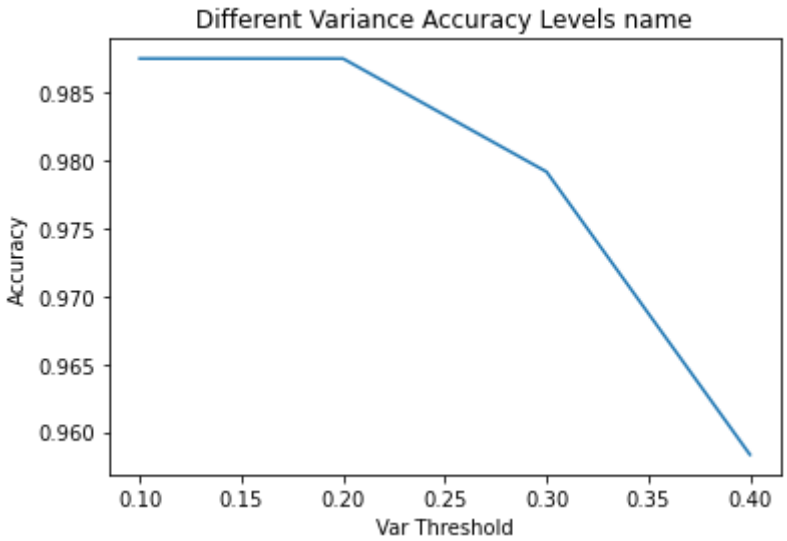
6. Compare the accuracy scores and define does the feature selection improve the model or not.

The differences between accuracies are small but it is worth mentioning that by increasing the variance threshold the accuracy tends to go lower usually at 0.4.

7. Show the of accuracy scores on a plot (line chart).

In [60]:
```
line_x = np.array(xaxis)
line_y = np.array(yaxis)


plt.plot(line_x,line_y)
plt.title('Different Variance Accuracy Levels name')
plt.xlabel('Var Threshold')
plt.ylabel('Accuracy')
plt.show()
```



Submission Instructions:

Please pay attention to the following tips.

- This quiz is an individual exercise.
- Please submit your work before the due date, March 6th, by the end of the day.