Objective:

1. Fitting KNN model to Treasury Data Set
2. Predicting with diffrent value of K
3. Determine the best Model by measuring performance

In [1]:
```
#Import Libraries
import csv
import numpy as np
from numpy.random import randn
import pandas as pd
from pandas import Series, DataFrame
import matplotlib.pyplot as plt
import os
import math
import re
pd.options.display.max_columns=40
#Import KNeighborsClassifier from sklearn.neighbors
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
```

In [2]:
```
### Load
path_hw2="C:\\Users\\fbaharkoush\\IE 598 Machine Learning\\Homework\\HW 2\\"
df_ts=pd.read_csv(path_hw2+"Treasury Squeeze test - DS1.csv").drop(["rowindex","contract"],axis=1)
```

In [3]:
```
### Missing Values
df_ts.isnull().sum().sum()==0
```

Out[3]:
```
True
```

In [4]:
```
X=df_ts.drop("squeeze",axis=1).values
y=df_ts["squeeze"].values
```

In [5]:
```
### Test and Train
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = .2, random_state=42, stratify=y)
```

Fit KNN for different values of K (Neighbours)

In [6]:
```
list_of_models_score=[]
list_of_y_pred=[]
list_of_neighbors=[]
for i in range(1,22):
    ### Create the Model
    knn = KNeighborsClassifier(n_neighbors=i)
    ### Fit values to the model
    knn.fit(X_train,y_train)
    ### Prediction
    y_pred=knn.predict(X_test)
    y_pred=list(y_pred)
    ### Store the prediction of each model
    list_of_y_pred.append(y_pred)
    #### Evaluate the score of each model
    knn_model_score=knn.score(X_test,y_test)
    ### Store the Scores of each model
    list_of_models_score.append(knn_model_score)
    ### Store Neghbour Number
    list_of_neighbors.append(i)
```

In [7]:
```
### Store the Performance of all models fitted
df_performance=pd.DataFrame({"Neighbours":list_of_neighbors,
             "Model Accuracy%":list_of_models_score})
df_performance["Model Accuracy%"]=df_performance["Model Accuracy%"]*100
df_performance.sort_values("Model Accuracy%",ascending=False).head(10)
```

Out[7]:

	Neighbours	Model Accuracy%
19	20	65.555556
20	21	63.333333
15	16	62.222222
4	5	61.111111
16	17	61.111111
5	6	60.555556
14	15	60.000000
18	19	60.000000
17	18	60.000000
8	9	60.000000

In [8]:
```
### Load the Predictions
df_prediction=pd.DataFrame(list_of_y_pred).T
### Add the neighbors number to the columns
df_prediction.columns=["K"+str(c) for c in list_of_neighbors]
### Merge Prediction with Y_test
df_prediction=pd.merge(df_prediction.reset_index(),
         pd.DataFrame({"Y Test":y_test}).reset_index(),
         left_on="index",right_on="index",how="left").drop('index',axis=1)
df_prediction.head()
```

Out[8]:

	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10	K11	K12	K13	K14	K15	K16	K17	K18	K19	K20	K21	Y Test
0	True	False	True	False	False	False	True	False	True	True	True	True	True	True	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
2	True	False	False	False	True	False	False	False	True	True	True	True	True	True	False	False	False	False	False	True	False	False
3	True	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	True
4	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False

In [9]:
```
best_k=df_performance.sort_values("Model Accuracy%",ascending=False)["Neighbours"].values[0]
best_k_accuracy=round(df_performance.sort_values("Model Accuracy%",ascending=False)["Model Accuracy%"].values[0],2)
```

In [10]:
```
print("The most accurate model is the one with " + str(best_k) + " neighbours with " + str(best_k_accuracy)+"% accuracy")
```

```
The most accurate model is the one with 20 neighbours with 65.56% accuracy
```

In [11]:
```
print("My name is Farbod Baharkoush")
print("My NetID is: fbahar2")
print("I hereby certify that I have read the University policy on Academic Integrity and that I am not in violation.")
```

```
My name is Farbod Baharkoush
My NetID is: fbahar2
I hereby certify that I have read the University policy on Academic Integrity and that I am not in violation.
```

In [ ]:

In [ ]: