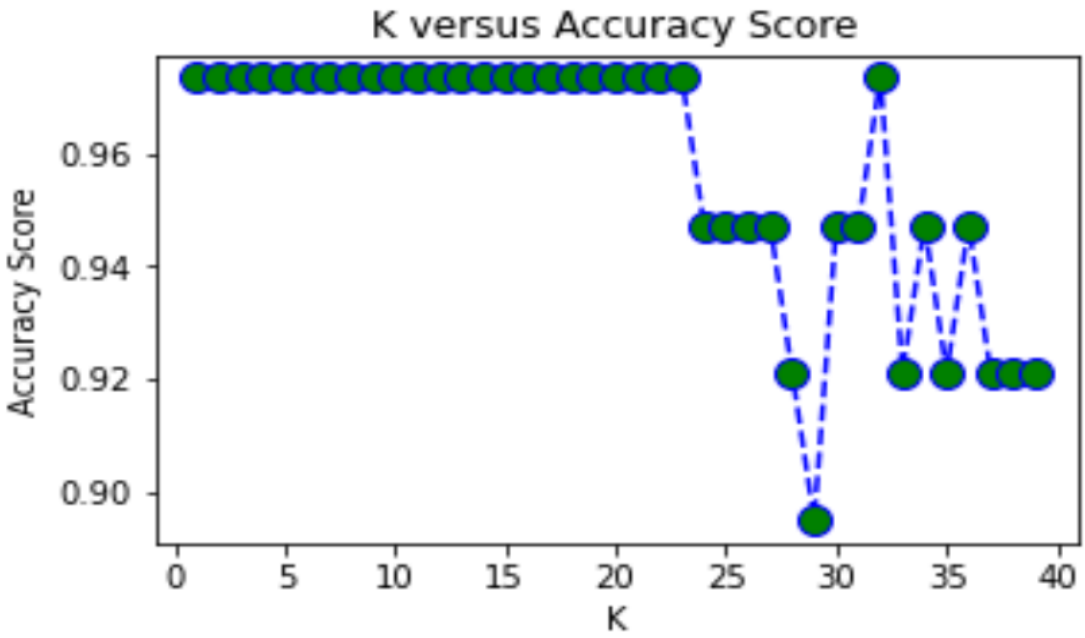
Quiz-1 – Econ 425

Anshika Sharma, UCLA ID(305488635)



The aim of this quiz is to examine how the parameter K from the K-Nearest Neighbor Method model changes as we increase the parameter. The training and test dataset split wasn't changed and we proceed with changing the parameter K in the KNeighborsClassifier model in Python. We can attempt to improve the accuracy of our results by modifying the number of neighbors using the following method. We first iterate through 40 neighbor values, represent a KNeighborsClassifier object with that number of neighbors. We can then fit the training data to this KNN model using knn.fit(), get the predictions, and append the mean value between the predictions, pred_i and the correct values, y_test. Where pred_i and y_test match up in the array, a true value is returned. The higher value for the accuracy score will correspond to a better performing model. These results can be plotted using the range of i values on the x-axis, versus the accuracy score on the y-axis. As the value of K increases till approximately first 23 values of k, the prediction accuracy is high, 0.97 which suggests the good performance of model. After 23, the accuracy rate starts declining and reaches it's lowest value at 29. At roughly k = 33 it again reaches 0.97 and then starts oscillating. As we increase the value of K, our predictions become more stable due to majority voting / averaging, and thus, more likely to make more accurate predictions (up to a certain point). Eventually, we begin to witness an increasing number of errors. It is at this point we know we have pushed the value of K too far. However, one should keep in mind that we did not change the test and training dataset split and it could also depend on how large the actual dataset is. With a smaller dataset, the model performs at the level we achieved above, but for much larger datasets, the accuracy depending on the value of K can drop at a faster rate.

Appendix - Codes

In [1]:
```
from sklearn.datasets import load_iris
iris_dataset = load_iris()
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    iris_dataset['data'], iris_dataset['target'], random_state=0)
```

In [3]:
```
accuracy_score = []
from sklearn.neighbors import KNeighborsClassifier
for i in range(1, 40):
    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(X_train, y_train)
    pred_i = knn.predict(X_test)
    accuracy_score.append(np.mean(pred_i == y_test))
plt.figure(figsize=(5, 3))
plt.plot(range(1, 40), accuracy_score, color='blue', linestyle='--',
         markersize=10, markerfacecolor='green', marker='o')
plt.title('K versus Accuracy Score')
plt.xlabel('K')
plt.ylabel('Accuracy Score')
```

Out[3]:
```
Text(0, 0.5, 'Accuracy Score')
```