CS 191 K-Nearest Neighbors (KNN) - Haberman's Survival Data Set

Submitted by
Femo Bayani and Mikaela Ramos

INTRODUCTION

K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is an algorithm that stores all available cases and classifies new cases based on a similarity measure, such as a distance function. KNN is used in statistical estimation and pattern recognition [1].

IMPLEMENTATION

Data Set

The data set was retrieved from Kaggle [2]. Additionally, a row indicating the column names were prepended to the csv file, and the data was split as such:

```
# Load dataset
df = pd.read_csv(os.path.join("dataset", "haberman.csv"))
X = df[["age", "op_year", "axil_nodes"]].values
y = df["surv_status"].values

# Split dataset into train and test data
X_train, X_test, y_train, y_test = train_test_split(X,
                                                    y,
                                                    test_size=0.2,
                                                    random_state=191,
                                                    stratify=y)
```

K-Nearest Neighbors

The KNN was tested from 1 number of neighbors to 200, with 10-fold cross validation score (still accuracy) as the metric for determining performance:

```
k_values = range(1, 200 + 1)
cv_scores = []

for k in k_values:
   # Create KNN classifier
   knn = KNeighborsClassifier(n_neighbors=k)
```

```
    # Fit the classifier to the data
    knn.fit(X_train, y_train)

    # Perform cross validation
    scores = cross_val_score(knn, X_train, y_train, cv=10, scoring='accuracy')

    # Add to cv_scores
    cv_scores.append(scores.mean())
```

Afterwards, the best K value is determined:

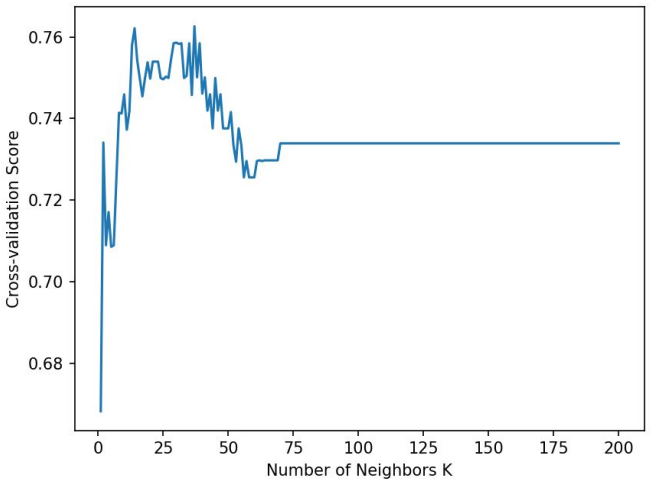
```
# Find best K
best_cv_score = max(cv_scores)
best_k = k_values[cv_scores.index(best_cv_score)]
print(f"Optimal K: {best_k} with {best_cv_score}")
```

Then, a graph is plotted, with the x values being the K value, and the y value being the corresponding cross-validation score.

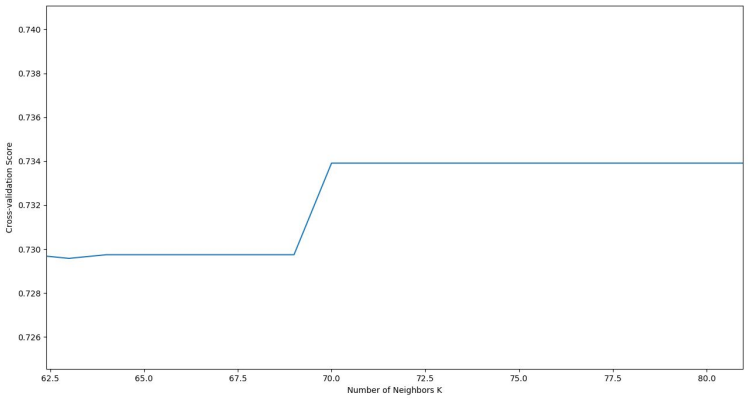
```
# Plot graph
plt.plot(k_values, cv_scores)
plt.xlabel("Number of Neighbors K")
plt.ylabel("Cross-validation Score")
plt.savefig("graph.png", dpi=150)
plt.show()
```

RESULTS

The graph comparing K-value versus the accuracy metric is shown below:



The score peaks at K = 37 with 76.26% accuracy, and goes stagnant from K = 70 onwards.



CONCLUSION

We can see that the performance of the model is initially low, but has an upward trend until it reaches a peak, then has a downward trend, until it stagnates.

REFERENCES

[1] Sayad, S. (2019). K Nearest Neighbors - Classification. Retrieved from https://www.saedsayad.com/k_nearest_neighbors.htm

[2] Lim, T-S. (1999). Haberman's Survival Data Set. Retrieved from https://www.kaggle.com/gilsousa/habermans-survival-data-set.