Antoine Gaton

Professor Rida Moustafa

CS356: Foundations of Big Data Analytics

7 July 2024

***Unit 4 - Individual Project***

* **Introduction**
  + Predictive analytics plays a crucial role in various fields by enabling organizations to forecast future events based on historical data. One effective method for such predictions is the K-Nearest Neighbors (KNN) algorithm. This document explains how the provided script leverages KNN to predict new records, specifically using a dataset of COVID-19 daily counts of cases, hospitalizations, and deaths.
* **Overview of the Script**
  + The script demonstrates the implementation of the KNN algorithm in Python. It involves the following key steps:
    - *Loading the Dataset:* The dataset contains daily counts of COVID-19 cases, hospitalizations, and deaths.
    - *Data Preprocessing:* Date features are extracted, and class imbalances are handled using SMOTE.
    - *Model Training:* The KNN model is trained using the preprocessed data.
    - *Model Evaluation:* The model's performance is evaluated using accuracy, a classification report, and a confusion matrix.
* **Steps for Predicting New Records**
  + *Data Preparation*
    - To predict new records, the new data must be prepared in the same way as the training data. This includes converting date columns to DateTime objects, extracting numerical features from dates, and ensuring that the feature set matches the training set.
      * Example Code for Data Preparation

| import pandas as pd  *# Load the dataset* data = pd.read\_csv(r"C:\Users\antoi\OneDrive\Documents\Coding\CTU\Foundation\_of\_Big\_Data\_Analytics\COVID-19\_Daily\_Counts\_of\_Cases\_\_Hospitalizations\_\_and\_Deaths.csv")  *# Convert date column to datetime* data['date\_of\_interest'] = pd.to\_datetime(data['date\_of\_interest'], format='%m/%d/%Y')  *# Extract numerical features from the date column* data['year'] = data['date\_of\_interest'].dt.year data['month'] = data['date\_of\_interest'].dt.month data['day'] = data['date\_of\_interest'].dt.day  *# Drop the original date column* data.drop(columns=['date\_of\_interest'], inplace=True) |
| --- |

* *Load and Process New Data*
  + Once the new data is prepared and the trained model is loaded, you can make predictions. The model will output the predicted class for each new record.

| new\_data = pd.read\_csv('path\_to\_new\_data.csv') new\_data['date\_of\_interest'] = pd.to\_datetime(new\_data['date\_of\_interest'], format='%m/%d/%Y') new\_data['year'] = new\_data['date\_of\_interest'].dt.year new\_data['month'] = new\_data['date\_of\_interest'].dt.month new\_data['day'] = new\_data['date\_of\_interest'].dt.day new\_data.drop(columns=['date\_of\_interest'], inplace=True) |
| --- |

* *Predict the Class for New Records*
  + Use the trained KNN model to predict the class (e.g., low, medium, high, very high) of new records.

| *# Make predictions on the test set* y\_pred = knn.predict(X\_test) |
| --- |

* *Output the Predictions*
  + The predictions can be saved or used for further analysis.

| *# Evaluate the model* accuracy = accuracy\_score(y\_test, y\_pred) print(f'Accuracy: {accuracy:.2f}') print('\n---------------------------------------------\n')  *# Detailed classification report with zero\_division parameter* report = classification\_report(y\_test, y\_pred, zero\_division=1) print('Classification Report:') print(report) print('---------------------------------------------\n')  *# Confusion matrix* conf\_matrix = confusion\_matrix(y\_test, y\_pred) print('Confusion Matrix:') print(conf\_matrix) print('\n---------------------------------------------\n') |
| --- |

* **Model Evaluation Result**
  + The output of the script provided the following results:
    - *Class Distribution*
      * This output shows the distribution of cases in the dataset, categorized into four classes: low (0), medium (1), high (2), and very high (3). The majority of cases fall into the "very high" (731) and "medium" (541) categories.
    - *Model Accuracy*
      * The model achieved an accuracy of 92%, indicating that it correctly predicted the class of new records 92% of the time.
    - *Classification Report*
      * The classification report provides detailed metrics for each class, including precision, recall, and F1-score. Class 0 (low) has lower precision and recall due to the small number of samples, while classes 1 (medium), 2 (high), and 3 (very high) show high precision and recall values, contributing to the overall high accuracy of the model.
    - *Confusion Matrix*
      * The confusion matrix shows the number of true and predicted values for each class. For example, the model correctly predicted 161 instances of class 1 and misclassified 5 instances of class 1 as other classes. Similarly, it correctly predicted 196 instances of class 3 and misclassified 13 instances of class 3 as other classes.

| --------------------------------------------- CASE\_COUNT 3 731 1 541 2 284 0 29 Name: count, dtype: int64 --------------------------------------------- Accuracy: 0.92 --------------------------------------------- Classification Report:  precision recall f1-score support   0 0.29 0.67 0.40 3  1 0.96 0.93 0.94 174  2 0.79 0.89 0.84 90  3 0.98 0.94 0.96 209   accuracy 0.92 476  macro avg 0.75 0.85 0.78 476 weighted avg 0.93 0.92 0.93 476 --------------------------------------------- Confusion Matrix: [[ 2 1 0 0]  [ 5 161 8 0]  [ 0 6 80 4]  [ 0 0 13 196]] --------------------------------------------- |
| --- |

* **Benefits of Scoring New Records**
  + *Real-Time Decision Making*: Predicting new records enables timely decisions based on forecasted data.
  + *Resource Allocation*: Helps in planning and allocating resources effectively, especially in healthcare settings.
  + *Trend Analysis*: Identifies emerging trends and potential outbreaks in advance.
* **Conclusion**
  + The predictions generated by the KNN model provide valuable estimates of expected case counts for new records, which can help decision-makers anticipate and respond to potential trends in COVID-19 cases. By effectively preparing new data, using the trained model to make predictions, and interpreting the results, you can gain actionable insights. This process of scoring new records not only aids in accurate forecasting but also supports informed decision-making. The KNN model, once trained and validated, stands as a robust tool for predictive analytics, enhancing the ability to manage and respond to emerging patterns in the data.