Final Project

INFO 4330 S50

Group I

Ali Manghat

James Tariga

Project Topic:

For this Project we decided to search for a data set that had data on a topic we found interesting and implement some of the important concepts we have covered in this course such as supervised learning which includes machine learning models such as linear regression, K-Nearest Neighbors, and logistical regression. We also wanted to test the topic of unsupervised learning which contains the topic of clustering. We found a data set that provides a detailed overview of the recent pandemic and an overview of covid-19 statistics for various different regions around the world. After finding this dataset we performed many of the machine learning techniques we have learned in this class.

Linear Regression:

Step 1: Importing the dataset in the variable df.

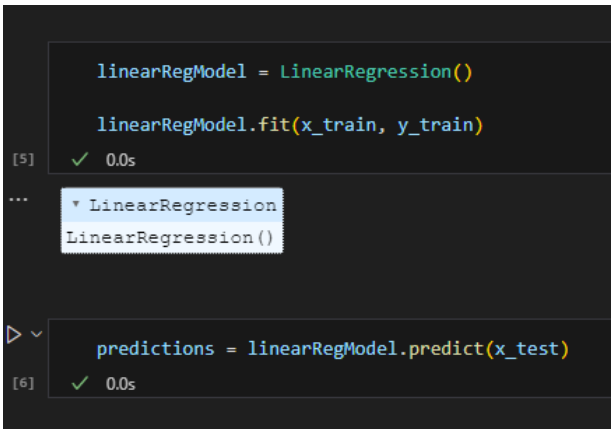
Step 2: We decided to use ‘Confirmed’, ‘Active’, ‘Recovered / 100 Cases’ as the independent variables, and ‘Death / 100 cases’ as the dependent variable.

Step 3: We split the data set for the first 131 rows for training and the remaining for testing. There are a total of 188 rows.

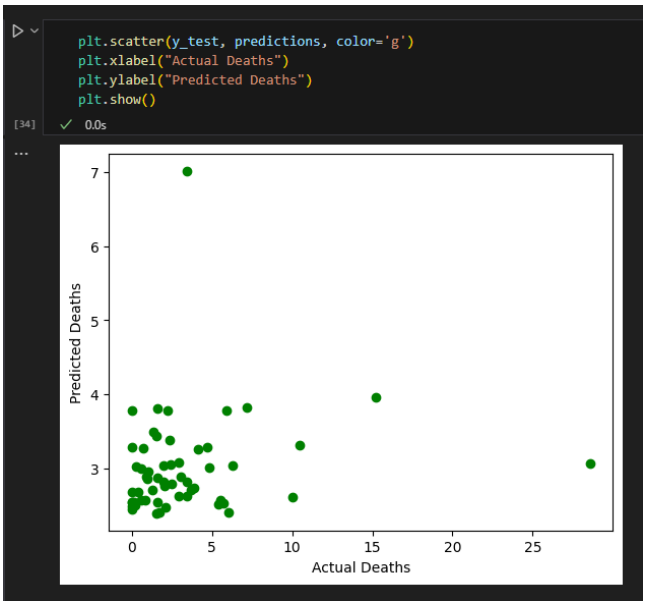
Step 4: We created a linear regression model.

Step 5: We trained the model using the training data.

Step 6: We make predictions on the model using the testing data.

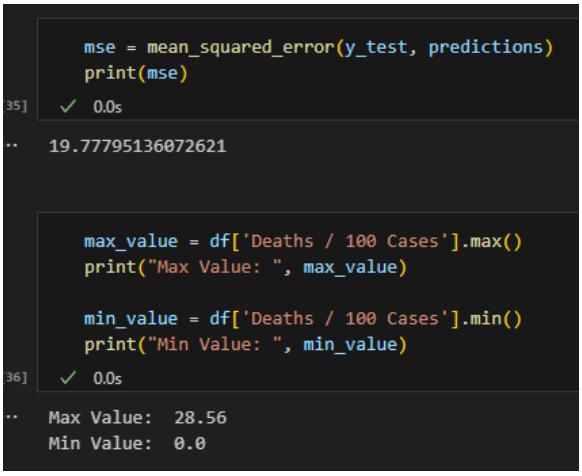


Step 7: Creating a scatter plot to show the actual and predicted values of the deaths / 100 cases.



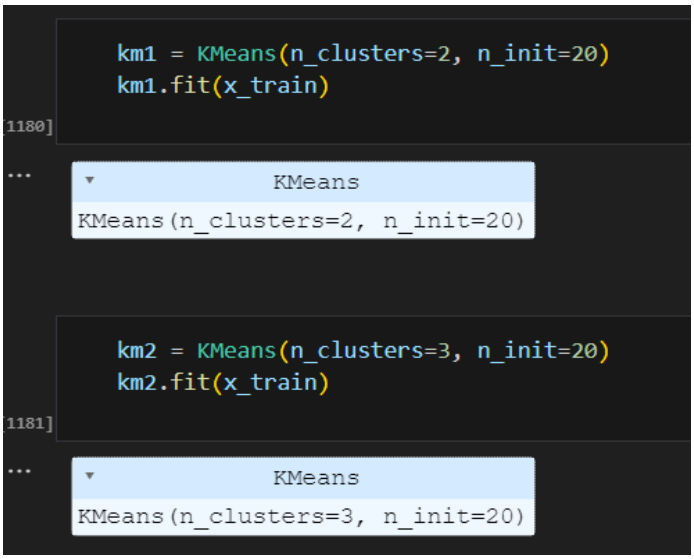
Step 8: Calculating MSE. The MSE is 19.78 which is relatively low. This means that the model is pretty accurate. The predicted values are like the actual values. We can see the max value for

the dependent variable ‘Deaths / 100 cases’ is 28.56 and the lowest is 0. This further proves that the MSE is relatively accurate.



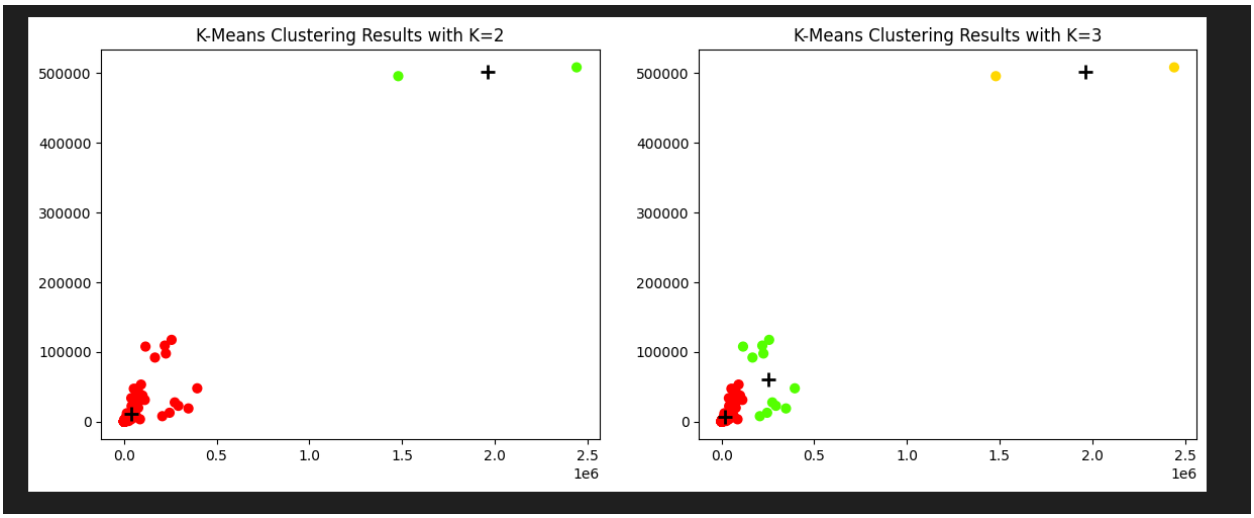
Clustering:

Step 1: We decided to make two clustering models. The first with 2 clusters, and the second with 3 clusters. W



Step 2: Plotting the clustered data. Here we can see the two models plotted. The graph on the left represents the first model which is grouping the data into two clusters and on the right the model is

grouping the data in three clusters. We can see covid-19 cases/deaths grouped by different regions. First world countries have a much lower number of deaths, the number increases as the countries become less developed. The centroids have also been plotted to show the 'mean' for each of the clusters.



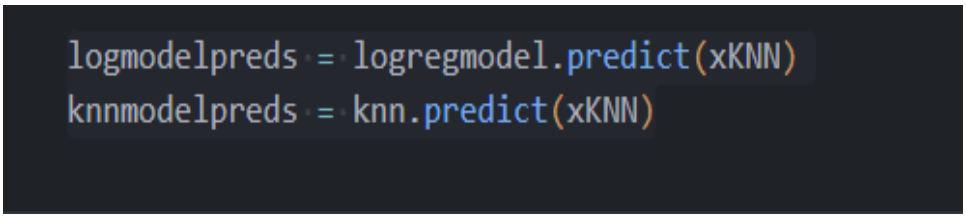
Comparison:

Both Linear regression and clustering have their use cases. Choosing which model to use comes down to what you are trying to predict. In our scenario linear regression makes sense because we are trying to map the features confirmed cases, recovered cases, and active cases to the dependent variable deaths. With clustering we are able to see what regions had the highest death rates but it does not provide us with a way to predict the deaths based on the features we are using. Therefore, linear regression is more appropriate for this specific scenario but clustering is also useful if you want to break down the dataset into clusters to visualize similarities in the data.

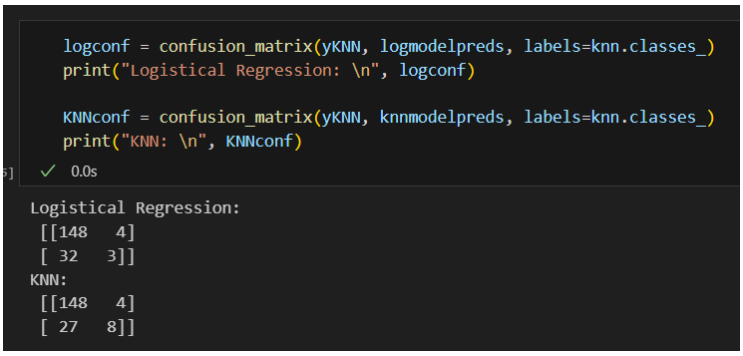
KNN and Logistic Regression:

Step 1: Initialized a feature called xKNN and a target called yKNN to train the model. Then the models are trained using the fit method on the provided training data.

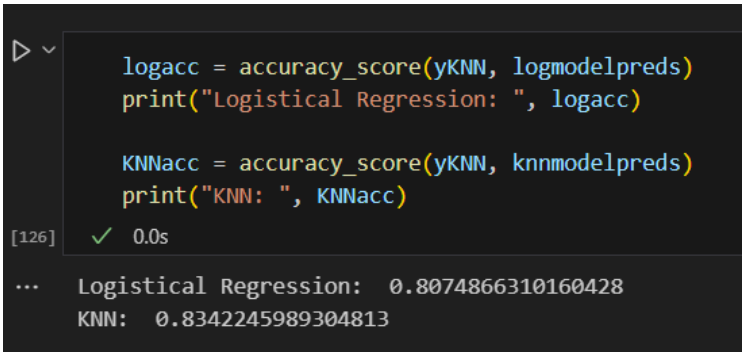
Step 2: We then make predictions on the same training data using the trained models.



Step 3: Confusion matrices are computed for both models using the true labels (yKNN) and predicted labels (logmodelpreds, knnmodelpreds).

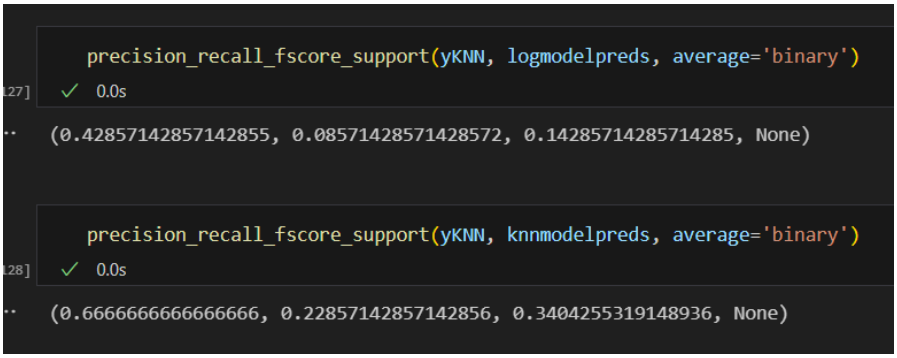


Step 4: The accuracy scores are calculated for both models using the true labels and predicted labels.



Step 5: Precision, recall, and F1-score are computed using the precision_recall_fscore_support function for both models, considering a binary classification scenario.

This essentially trains two classifiers, makes predictions, evaluates their performance using confusion matrices, accuracy scores, and precision-recall-f1 metrics. It provides a comprehensive overview of how well each model is performing on the given dataset.



Comparison:

Based off the output below, we can conclude that the Logistic Regression model correctly predicts the majority class ('not Americas' in this case) well but struggled with the minority class ('Americas').

True Negative: 148

False Positive: 4

False Negative: 32

True Positive: 3

KNN:

On the other hand, the KNN model shows a slight improvement over Logistic Regression in predicting the minority class, with more true positives and fewer false negatives.

True Negative: 148

False Positive: 4

False Negative: 27

True Positive: 8

These values indicate the overall proportion of correct predictions made by each model. KNN performs slightly better in this regard.

Logistical Regression: 0.8074866310160428

KNN: 0.8342245989304813

Both models are more inclined towards predicting the majority class correctly while struggling with the minority class. We believe the reason as to why this occurs is because our dataset is imbalanced, and this is a common issue with imbalanced datasets. To improve model performance, techniques like resampling the dataset, trying different classification algorithms, or using different feature selection methods might alter the result. Additionally, adjusting the decision threshold and employing methods to address class imbalance can be considered to improve the recall without significantly compromising precision.

Citation of data set:

Devakumar, K.P. (n.d.). Corona Virus Report. Kaggle. Retrieved from

https://www.kaggle.com/datasets/imdevskp/corona-virus-report