Avinash Chouhan
CISC 484 Final Project

Traditional Model (RandomForest Regression)

For the traditional machine learning model, I utilized the RandomForest regression model to predict the weather data of the next hour given the weather conditions of the last 6 hours of data. The features that I used for the model are the temperature, humidity and pressure. With the help of the given features, the RandomForest model is able to recognize patterns in the given data and predict the most likely outcome, giving us the prediction of the weather. To implement the model I first had to scale the features and create lagged features of the last six hours. The model then evaluated these features giving us the MSE values. The dataset was split into a standard 80-20 split for training and testing, respectively. The MSE values from the model can be seen below. The first list depicts the predicted values for temperature, humidity and pressure, the MSE and MSE per feature are shown after that. Here the MSE values are quite low showing that the model actually predicted a similar point compared to the given data points. This would mean that the random forest regression model did a good job at predicting the next hour’s weather given the previous 6 hours of data.

```
[[  20.5623    99.99    1010.04181]
 [  20.5659    99.989   1010.04178]
 [  20.5621    99.989   1010.03513]
 ...
 [  22.2563    96.247   1010.12035]
 [  22.2466    96.217   1010.08297]
 [  22.2526    96.172   1010.08959]]
Mean Squared Error (Overall): 3.1001970484341137
Mean Squared Error (Per Feature): [6.60698082 0.90853945 1.78507087]
```

LSTM Model

The LSTM model was overall more challenging due to its complexity. The deep learning model learns on 12 hours of data to predict the weather conditions for the next 3 hours. I started with creating sequential data from the given file, including the three features, temperature, humidity and pressure. Upon research found that the LSTM model is well suited for forecasting tasks due to its neural network which maintains memory of previous inputs. Due to the complex nature of the algorithm it is also very capable in handling complex predictions, like the one we will make in this case. However this also does make the algorithm more intensive, requiring longer run time and more resources. I processed the data by scaling the features and then creating sequences based on the timeline. There were multiple layers involved in the network, with a few dropout layers as well to prevent overfitting. I chose to use the Adam optimiser to train the model as I have used it in our previous projects. After processing the dataset was split into the standard 80-20 split for training and testing, respectively. Calculating the model's MSE we can see quite a large number. I believe this is an issue with scaling values, causing the model to get confused and not be able to merge to a correct value. It could also be caused by overfitting of the data. I did not have time to diagnose the issue but I believe its one of the two reasons listed above.

```
Mean Squared Error (Overall): 343271.1843156395
Mean Squared Error (Per Feature): [6.54461495e+02 7.68292429e+03 1.02147617e+06]
```