Machine Learning (CSC8635)

Datasets: 1. Weather Dataset
2. Twitter Dataset

Student Name: Akash Mohandoss
Student ID: 220488118

1. Introduction:

This report gives a brief analysis of two different datasets. Firstly, the **Twitter dataset** contains the columns 'tweet ids' – hash unique id for the tweets, 'text' - column containing text comments given by the people with respect to the Twitter handles, and the 'airline sentiment' column which tells us the type of sentiment review. Secondly, the **Weather dataset** provided 8 columns namely 'Date and time', 'Internal sensor temperature1', 'Outside temperature', 'CPU Temperature', 'Internal sensor temperature2', 'Air Pressure', and 'Humidity'.

The key aspect discussed include Shallow and Deep Classifier models that have been implemented to predict the sentiment of the text comments provided in the Twitter data 'text' column. As a result, based on the evaluation metrics the best model to predict the sentiment of the text is being derived. Furthermore, weather data is pre-processed, multiple classifier models have been implemented for each class label, and future values are predicted for the same.

2. Exploratory data analysis:

Exploratory Data Analysis (EDA) is the procedure of methodically analysing and condensing a dataset to learn more about its attributes and qualities. EDA is frequently the initial stage of data analysis and can provide guidance for the creation of more formal statistical models and hypothesis testing.

Twitter Data EDA-

Exploratory Data Analysis is carried out for the provided Twitter data which involves gathering basic insights from the given data to obtain an overall overview of the data.

Columns	Non-Null	Count	Dtype
Tweet_id	11858	non-null	Int64
text	11858	non-null	object
airline_sentiment	11858	non-null	object

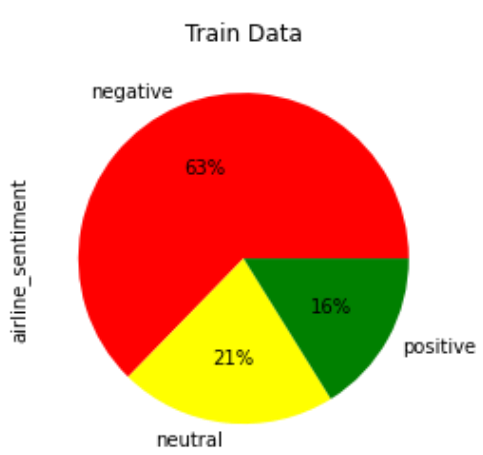
Train data

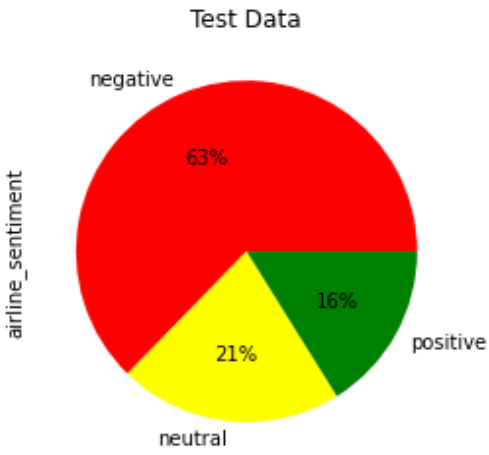
Columns	Non-Null	Count	Dtype
Tweet_id	1464	non-null	Int64
text	1464	non-null	object
airline_sentiment	1464	non-null	object

Test data

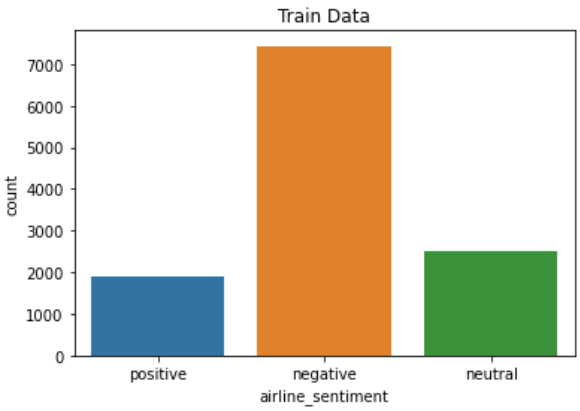
Columns	Non-Null	Count	Dtype
Tweet_id	1318	non-null	Int64
text	1318	non-null	object
airline_sentiment	1318	non-null	object

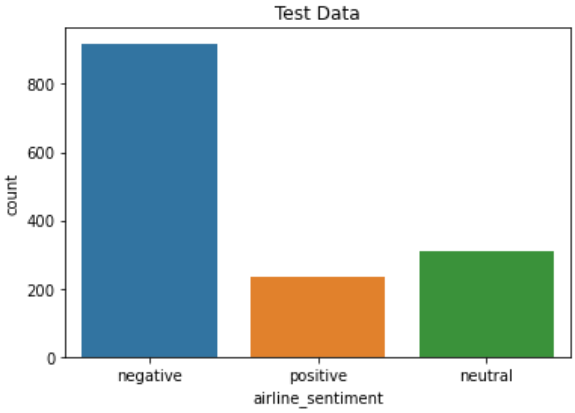
Validation data





The above plots show the percent of data in each sentiment category in a pie chart and the below bar chart depicts the count of each sentiment for the given train data and test data respectively.





It has been shown that the data provided to us contains a significant percentage of negative tweets than other sentiment-type tweets. Both Train data and test data have a higher ratio of negative sentiment data and thus will have an impact on the classifier accuracies.

Airline sentiment	Text
Negative	7434
Neutral	2510
Positive	1914

Train data

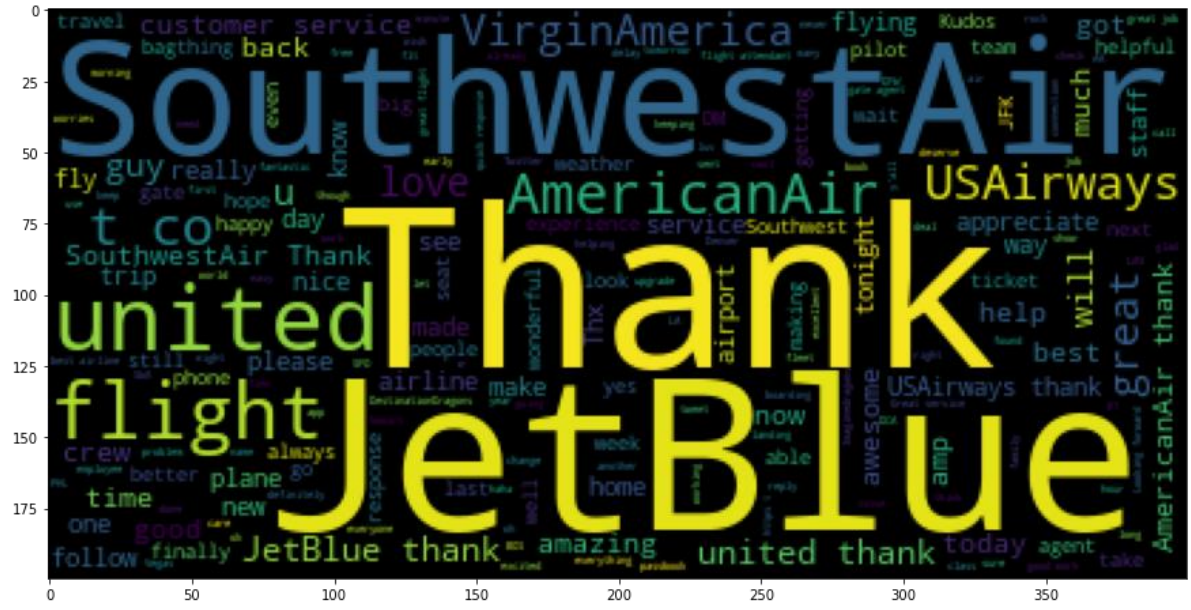
The above table shows us the count of data on each sentiment category as depicted earlier by the bar charts and the pie chart in the above segment. From the table above of the train data we can see that there are 7434 Negative tweet data, 2510 Neutral tweet data, and 1914 Positive tweet data.

Further, the below table for test data shows us that the count of Negative tweet data is 918, Neutral tweet data is 310 and Positive tweet data is 236.

Airline sentiment	Text
Negative	918
Neutral	310
Positive	236

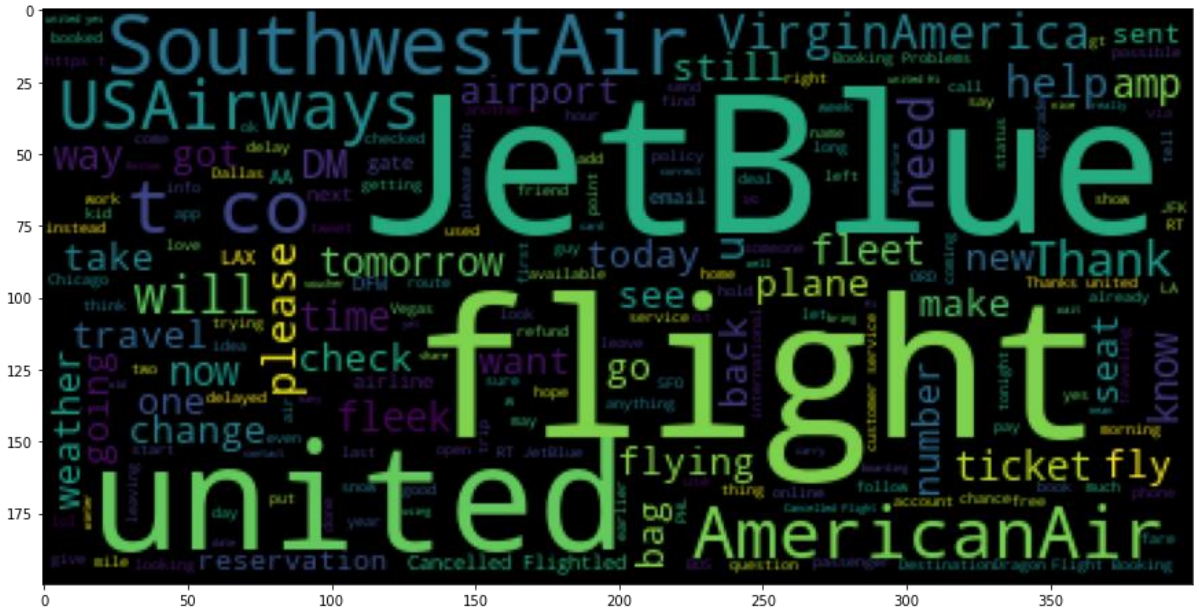
Test data

WordCloud for the Positive tweets:



WordCloud for the Negative tweets:

WordCloud for the Neutral tweets:



It is apparent that from the above wordcloud images for the train data sentiment individually, we can say that positive sentiment tweets are more used words such as Thank, JetBlue, Southwest, and Air. For Negative sentiment tweets, the more often used words are USAirways, United, and Flight. Neutral tweets contain words such as JetBlue, Flight, and United. The words identified are more used in each sentiment and Wordcloud identifies the words that are more often used and shows it in enormous size in the word cloud compared to other words which appear lesser in number.

Weather Dataset-

The weather dataset contains data collected by a Raspberry Pi computer at home in Newcastle. It consists of 12 weather features collected over approximately 12 months. The features available are shown in the below table.

S no.	Feature
1	Date and time [Linux Format]
2	First Sensor Internal Temperature (C)
3	Outside Temperature (C)
4	CPU Temperature (C)
5	Count
6	Second Sensor Internal Temperature (C)
7	Air Pressure (mmHg)
8	Humidity (percentage)

Weather data features

Numerical summaries of the Weather time series data:

	1st internal sensor temperature	Outside Temperature	CPU Temperature	Count
Count	545434.000	545434.00	545434.00	545434.00
Mean	21.177	11.189	33.285	1.000
Std	3.289	6.855	2.903	0.002
Min	11.300	-3.100	25.200	1.000
25%	18.800	6.300	31.476	1.000
50%	21.100	10.200	33.628	1.000
75%	23.200	14.900	34.704	1.000
Max	36.200	49.000	47.780	3.000

	2nd internal sensor temperature	Air Pressure	Humidity
Count	545434.000	545434.00	545434.00
Mean	23.401	1001.626	36.735
Std	3.142	11.94	7.410
Min	14.520	958.660	19.850
25%	21.210	994.280	30.950
50%	23.460	1002.530	35.710
75%	25.270	1009.640	41.780
Max	38.510	1031.780	58.710

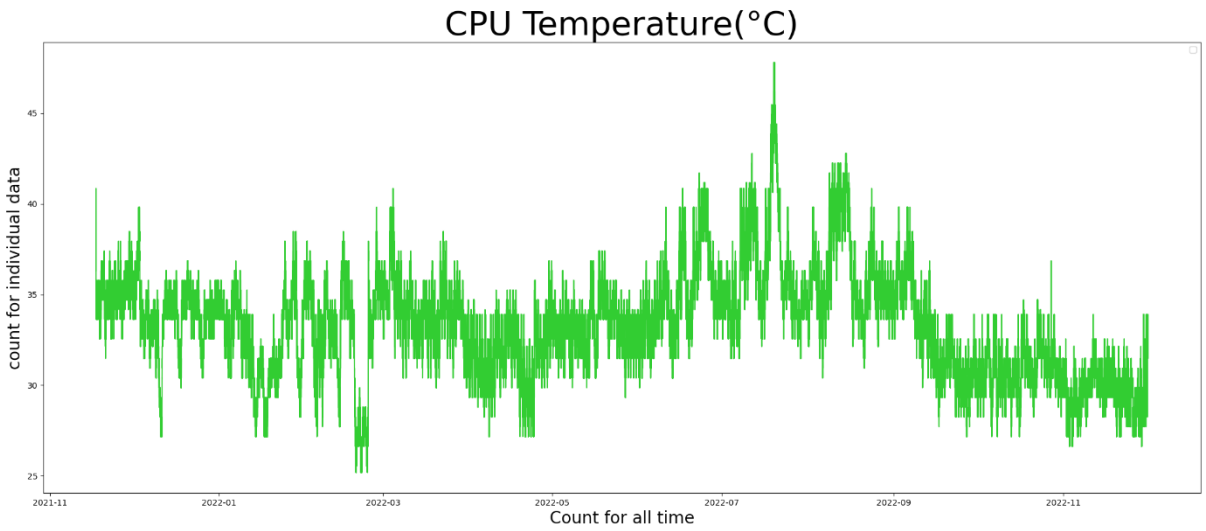
Weather features description

From the numerical analysis of the weather dataset shown in the above tables, we can infer that the total data count is 545434 rows. The second internal temperature has a mean temperature higher than the first internal sensor temperature. Outside temperature collected in the given time has an average mean of 11.189 and a maximum value of 49 degree. The highest CPU temperature that can be recognized in the given data was 47.780 degree Celsius. The mean Air Pressure and Humidity were identified to be 1001.626 and 36.735, respectively.

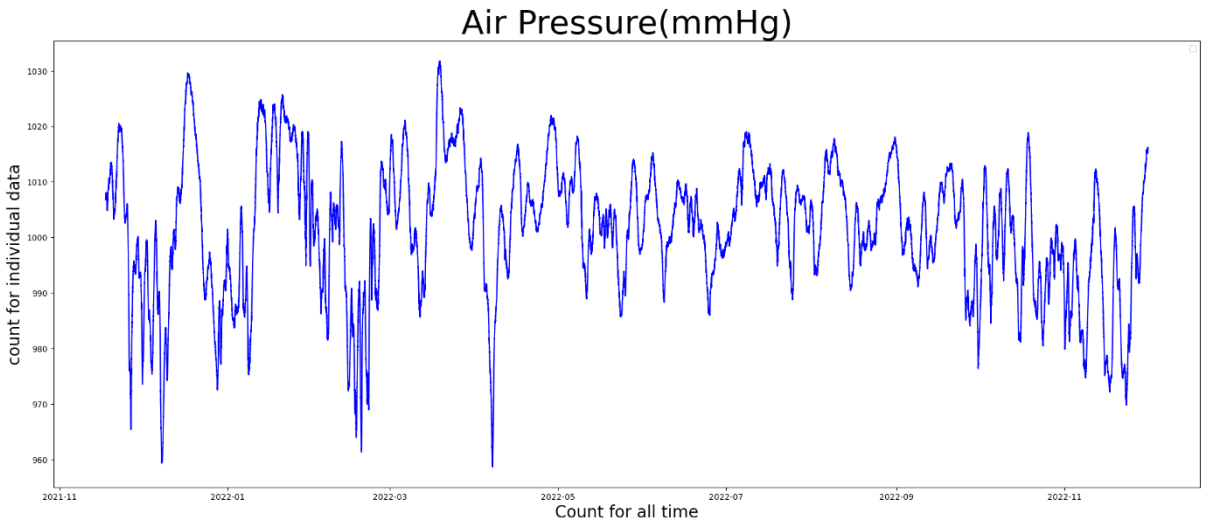
Graphical Analysis of Weather time series Dataset:

The Graphical summaries of the data provide us with an overall view of the information that the data is conveying to us. This helps us understand how the time and various weather features are varying from the initial to the final time recorded.

Line plot for CPU Temperature:

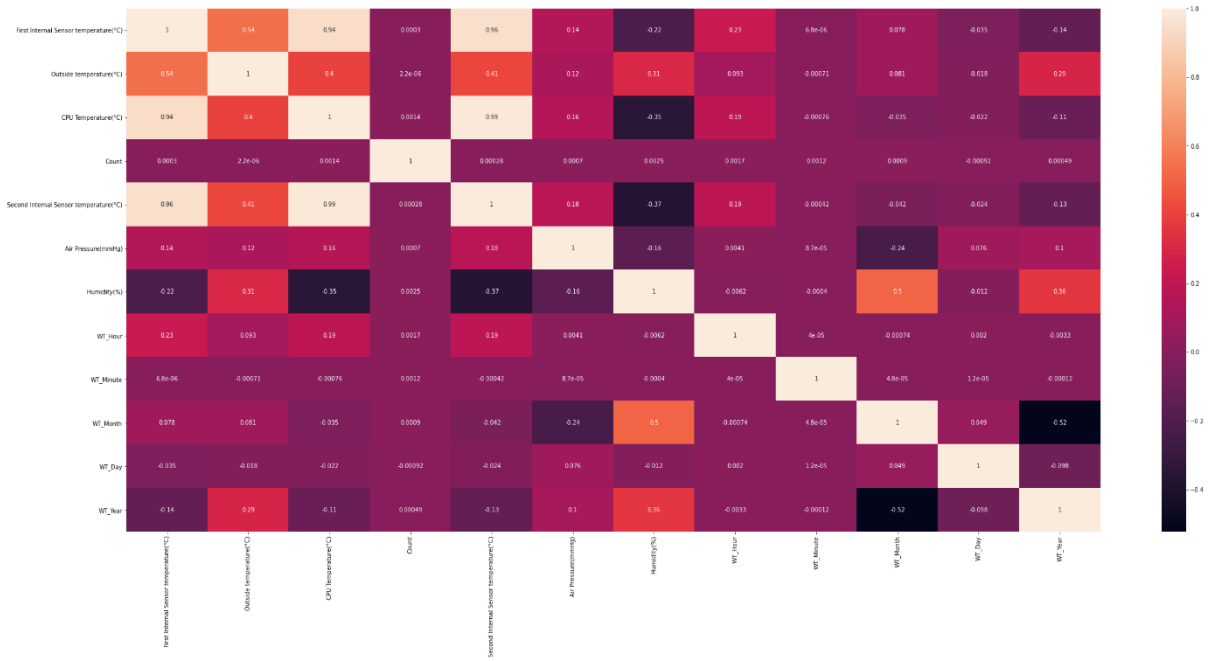


Line plot for Air Pressure:



The line plots show us the varying CPU temperature(celsius) and Air pressure(mmHg) recorded while the time when the data was captured.

A correlation heatmap plot is plotted against all weather data features to find the correlation between features. Further, we can infer that the First sensor temperature is highly correlated with the CPU Temperature and the Second sensor temperature.



Correlation plot of weather data

3. Data Pre-Processing:

The Twitter dataset is pre-processed through a few steps as listed below,

- Removing Twitter Handles
- Removing the punctuations from the data
- Removing URL and website links in the data
- Tokenization
- Removing stopping words
- Stemming and Lemmatization
- Vectorizing the data

To elaborate, the first three steps remove the Twitter handles, punctuations, and URLs from the data and prepare the data for the next stage of pre-processing. Tokenization splits the returned text into chunks of token and saves it as a list. Further, stopping words are removed from the list since it does not add additional information to the Twitter data and to get more accuracy. Stemming and lemmatization are normalization techniques used to normalize the text data and the last is the vectorization stage comes into play where we need to convert the text data into vector values to apply them for the shallow and deep classifier models and predict the accuracy.

On the other hand, the weather dataset is loaded, and column names are added to the given data frame. Further, the given timestamp column is sliced into individual Months, Years, Days, hours, and Minutes. Finally, the given timestamped has been made the index of the data frame instead of being a feature and once the column names are added, the features available in the data are listed below,

- First Internal Sensor temperature(°C)
- Outside temperature(°C)
- CPU Temperature (°C)
- Count
- Second Internal Sensor temperature(°C)
- Air Pressure (mmHg)
- Humidity (%)
- WT_Hour
- WT_Minute
- WT_Month
- WT_Day
- WT_Year

To begin with the Analysis part, we first segregate the given weather data into Train and Test Data. In this Analysis, I have sliced the data into two halves. The data after October namely November and December data in between time periods 2021 and 2022 are taken as Test Data and the monthly data of October and months before are considered as Train data.

4. Applying Machine learning and deep learning models/algorithms for the pre-processed data.

Based on the ideas of machine learning, a machine learning algorithm is a mathematical method that finds patterns and trends in the dataset that has been passed to the respective model. some of the popular models are the RandomForest classifier, Support Vector Machine model, Logistic Regression model, Naïve Bayes model, Decision tree model, and so on.

In addition, Deep learning is a variant of machine learning which follows a neural network with multiple layers to predict the accuracy of the models. Machine learning on the other hand is neural architecture with a single layer when compared to the deep learning models.

The Shallow classifier models that have been used for predicting the accuracy of the data are,

- Naïve Bayes model
- Linear SVM (Support Vector Machines) Model
- Random Forest Classifier model
- XGB Classifier model
- Ensembled model approach

As discussed in the above paragraph deep learning models contains multiple layers to predict the accuracy of the given Training and Test data. These layers are hidden, and the embedding weights are randomly initialized acting just like another layer in training the data. For Twitter sentiment analysis we have used the GloVe algorithm to generate an embedding layer-alike lookup table where the words act as the keys and the vectors act as the values for the keys.

The Deep classifier models that have been used for predicting the accuracy of the data are,

- Convolutional Neural Network (CNN) Model
- Bidirectional Long Short Term Memory (BiLSTM)model

Comparison Table of all the Machine learning and Deep learning models

Comparison of all algorithm results:

Model	**F1_Score**
SVM Algorithm	0.7835
Naive Bayes Algorithm	0.6954
Random Forest Algorithm	0.7698
Ensemble Learning Model	0.7739
CNN Model	0.627
Bidirectional LSTM Algorithm	0.627

Best Model:

Model	**F1_Measure**
SVM Algorithm	0.7835

F1_score is a measure of a model's accuracy that balances precision and recall. The F1-score is calculated as the harmonic mean of precision and recall, where the best value is 1 and the worst value is 0.

F1_score= 2 * (Precision * Recall) / (Precision + Recall)

The above two tables illustrate the F1-score of 6 models of which two are deep learning models and the rest are machine learning models. The graph shows us that the SVM algorithm has an F1_score of 0.7835, Naïve Bayes [0.6954], Random Forest [0.7698], Ensemble Learning [0.7739], CNN model [0.62], and Bidirectional LSTM Algorithm [0.627]. Thus, the F1_measure for the SVM algorithm is maximum when compared to other algorithms for the sentiment analysis of the data given.

The Supervised model or regression models used in this project to predict the accuracy of the weather data are,

- RandomForestRegressor algorithm
- GradientBoostingRegressor Algorithm
- KNN Regression Algorithm
- RNN Algorithm
- SNN Algorithm

For weather time series data, the dependent variable is usually a measure of a phenomenon over time and the independent variables may include other time-based covariates.

MLPRegressor algorithm is a supervised machine learning algorithm using a neural network with multiple layers. It is one of the models used to forecast the time series data. The reason the MLP regressor is an excellent choice for forecasting is that time series data often have non-linearity patterns in the data.

This model can easily model non-linear relationships between the input and the outputs. It is very flexible and can handle missing data and high-dimensional data which might be encountered during forecasting. Thus, the MLP regressor was a desirable choice to perform the forecasting to predict the future values for the weather data.

Comparison Table of MSE Score of all regression models for CPU temperature:

Model	**MSE Score**
RandomForestRegressor Algorithm	0.003
GradientBoostingRegressor Algorithm	2.036
KNN Regression Algorithm	0.099
RNN Algorithm	7.247
SNN Algorithm	42.008

According to the above table, the MSE score for the SNN algorithm is extremely high followed by the RNN algorithm and GradientBoostingRegressor algorithm. The efficiency for the RandomForest and KNN models is high because it has lower MSE to predict future values based on the given data.

Comparison Table of RMSE Score of all regression models for CPU temperature:

Model	**RMSE Score**
RandomForestRegressor Algorithm	0.131
GradientBoostingRegressor Algorithm	0.362
KNN Regression Algorithm	0.303
RNN Algorithm	0.977
SNN Algorithm	31.46

An RMSE is the squared root value of the Mean Squared root (MSE) and is the commonly used metric for evaluating the performance of regression models. Based on the RMSE value already in consideration from the below table, we can infer that RandomForestRegressor Algorithm has a low RMSE score when compared to all other models. Thus, we can finalize that RandomForestRegressor is the most efficient among the ones to which the model has been fitted with the weather data for the specific feature.

Conclusion:

To conclude, Exploratory Data Analysis and Data pre-processing have been carried out in a systematic way throughout the dataset individually to achieve good accuracy. In general, the dataset's helped me understand a lot of concepts about how neural networks work and how neural network models carry out their processes. All the regression models used for the weather dataset showed a higher number value for MSE and RMSE which had to be small for a model to be efficient. RandomForest regressor is the chosen one that yields a good throughput. Furthermore, in the Twitter data sentiment analysis, all the models showed us good accuracy and the decision was taken based on the evaluation metric 'F1-score'.

Finally, the SVM model was the chosen model to predict perfect sentiments from the Twitter dataset among the others which showed less accuracy.

References link:

- https://www.kaggle.com/code/prashant111/complete-guide-on-time-series-analysis-in-python
- https://towardsdatascience.com/time-series-analysis-in-python-an-introduction-70d5a5b1d52a
- https://www.kaggle.com/code/paoloripamonti/twitter-sentiment-analysis
- https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/