# SENTIMENT ANALYSIS OF FACEBOOK COMMENTS

#### A MINI PROJECT REPORT 18CSC305J - ARTIFICIAL INTELLIGENCE

***Submitted by***

**AMIR MUSTAQUE[RA21110470100054] YESHWANTH SURISETTY[RA2111047010029]**

*Under the guidance of*

## Dr. Karpagam M

Assistant Professor, Department of Computer Science and Engineering

***in partial fulfillment for the award of the degree of***

#### BACHELOR OF TECHNOLOGY

in

## COMPUTER SCIENCE & ENGINEERING

of

#### FACULTY OF ENGINEERING AND TECHNOLOGY



S.R.M. Nagar, Kattankulathur, Chengalpattu District

#### MAY 2024

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

(Under Section 3 of UGC Act, 1956)

## BONAFIDE CERTIFICATE

Certified that Mini project report titled **“Sentiment Analysis of Facebook Comments”** is the bona fide work of **Amir Mustaque [RA21110470100054], Yeshwanth Surisetty[RA2111047010029]** who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**SIGNATURE HEAD OF THE DEPARTMENT**

Faculty In-Charge **Dr. R Annie Uthra**

**Dr.Karpagam.M** Professor and Head

Assistant Professor, Department of Computational Intelligence

Department of Computational Intelligence, SRM Institute of Science and Technology

SRM Institute of Science and Technology Kattankulathur Campus, Chennai

Kattankulathur Campus, Chennai

## ABSTRACT

Weather forecasting is essential for various sectors such as agriculture, transportation, and disaster management. With the increasing availability of weather data and advancements in data analysis techniques, it has become possible to perform detailed analysis and prediction of weather patterns. In this project, we aim to analyze historical weather data and develop a predictive model to forecast rainfall.

The project involves several key steps, including data collection, preprocessing, exploratory data analysis (EDA), feature engineering, model development, model evaluation, and rain prediction. We gather historical weather data from reliable sources and preprocess it to ensure data cleanliness. EDA techniques are then applied to gain insights into the relationships between different weather parameters and rainfall. Feature engineering is performed to enhance the predictive performance of the model, followed by the development of machine learning algorithms for rain prediction.

The performance of the developed model is evaluated using appropriate metrics, and predictions are made for future time periods. Accurate rainfall predictions can have significant implications for agricultural planning, water resource management, and disaster preparedness.

Through this project, we aim to demonstrate the effectiveness of advanced data analysis techniques and machine learning algorithms in providing valuable insights and predictions for weather-related phenomena, ultimately contributing to informed decision-making and risk mitigation strategies



|  |
| --- |
| ABSTRACT |
| TABLE OF CONTENTS |
| LIST OF FIGURES |
| 1. INTRODUCTION |
| 1. LITERATURE SURVEY |
| 1. SYSTEM ARCHITECTURE AND DESIGN |
| * 1. Architecture diagram |
| 3.2 Description of models used |
| 1. METHODOLOGY |
| 1. INFERENCE |
| 1. CONCLUSION |
| 1. RESULTS AND DISCUSSION |
| REFERENCES |



* 1. ARCHITECTURE **9**
  2. Correlation Matrix **9**
  3. Histograms of Numerical Features **10**
  4. Comparison of Actual vs predicted Values **12**
  5. Confusion Matrix **13**
  6. Model Evaluation Metrics **13**
  7. Feature importance **14**
  8. Variation of Accuracy and Loss During Training **14**
  9. Distribution of Categorical Feature

## CHAPTER1

## INTRODUCTION

This presents a comprehensive analysis of weather data from different areas of Australia and aims to build a binary classification model to predict whether it will rain tomorrow. The dataset used for this task is a midsized collection of weather records, which provides valuable insights into the weather patterns and allows us to develop a machine learning model for rain prediction.

Understanding and predicting weather conditions are of great importance in various domains, such as agriculture, transportation, and urban planning. Accurate rain prediction can assist in making informed decisions, mitigating risks, and optimizing resource allocation. By leveraging machine learning techniques, we can analyze historical weather data and develop models that effectively predict the occurrence of rain.

In this notebook, we will follow a structured workflow to explore the weather dataset, preprocess the data, perform exploratory data analysis (EDA), engineer relevant features, train a binary classification model, and evaluate its performance. We will utilize popular libraries such as pandas, NumPy, scikit-learn, and matplotlib for data manipulation, analysis, model training, and visualization.

The notebook will walk through the necessary steps, including handling missing values, outlier detection and treatment, feature engineering, splitting the dataset into features and target variable, model training using a RandomForestClassifier, and evaluating the model's performance using various metrics.

By the end of this you will have gained insights into the weather patterns in different areas of Australia and developed a robust model for predicting whether it will rain tomorrow. This knowledge can be applied to real-world scenarios where accurate rain prediction is crucial for decision-making and planning.

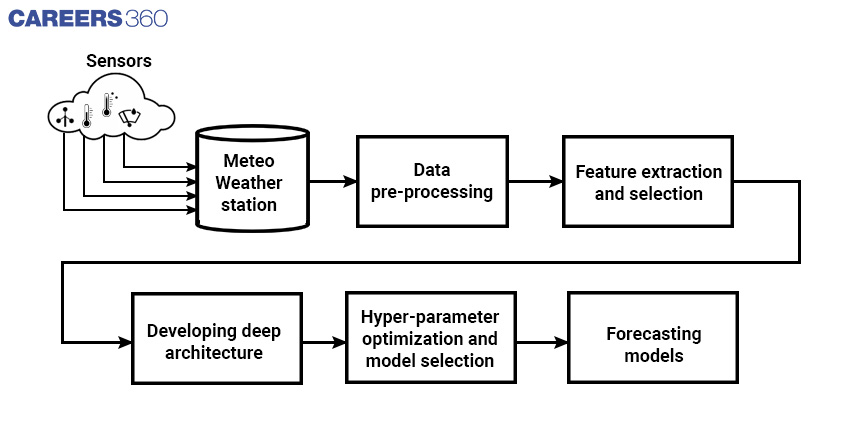
## CHAPTER 2

## LITERATURE SURVEY

The literature survey in the field of weather data analysis and rainfall prediction reveals a comprehensive exploration of methodologies and techniques aimed at enhancing the accuracy and reliability of predictive models. Researchers have extensively utilized diverse data sources, including ground-based weather stations, satellite imagery, and radar systems, to collect meteorological data. Preprocessing techniques such as data cleaning and outlier detection have been employed to ensure data quality before analysis. Exploratory data analysis (EDA) plays a crucial role in uncovering patterns and trends within weather datasets, with visualizations like time series plots and heatmaps aiding in the identification of seasonal variations and correlations between variables. Feature engineering techniques, such as lagged variables and interaction terms, have been explored to capture complex relationships and improve model performance. Machine learning algorithms, ranging from traditional regression models to advanced ensemble methods like gradient boosting and stacking, have been applied for rainfall prediction tasks. Evaluation metrics such as accuracy, precision, and ROC-AUC are commonly used to assess model performance, emphasizing the importance of domain-specific evaluation criteria. Despite significant progress, challenges such as data scarcity and model interpretability persist, driving future research towards integrating advanced data assimilation techniques and ensemble modeling approaches to enhance forecast accuracy and reliability.

## CHAPTER 3

**SYSTEM ARCHITECTURE AND DESIGN**

****

#### Data Information and Design

Figure 3.1 Data Information and Design

The first figure is about data information and description, which includes the output of the **data.info()** and **data.describe()** functions from the pandas library. The **data.info()** function provides essential details about the data structure, such as the number of entries (rows), the data types of each column (e.g., integers, floats, strings), and any missing values in each column. It also shows the memory usage of the dataset. This information helps you understand the dataset's composition and identify any columns that might need further cleaning or transformation. Additionally, **data.describe()** gives you summary statistics for each numerical column in the dataset, including the mean, median, standard deviation, minimum, and maximum values. This helps you understand the distribution and range of values for each numerical feature, which is important for preprocessing and feature engineering.

**Correlation Matrix**

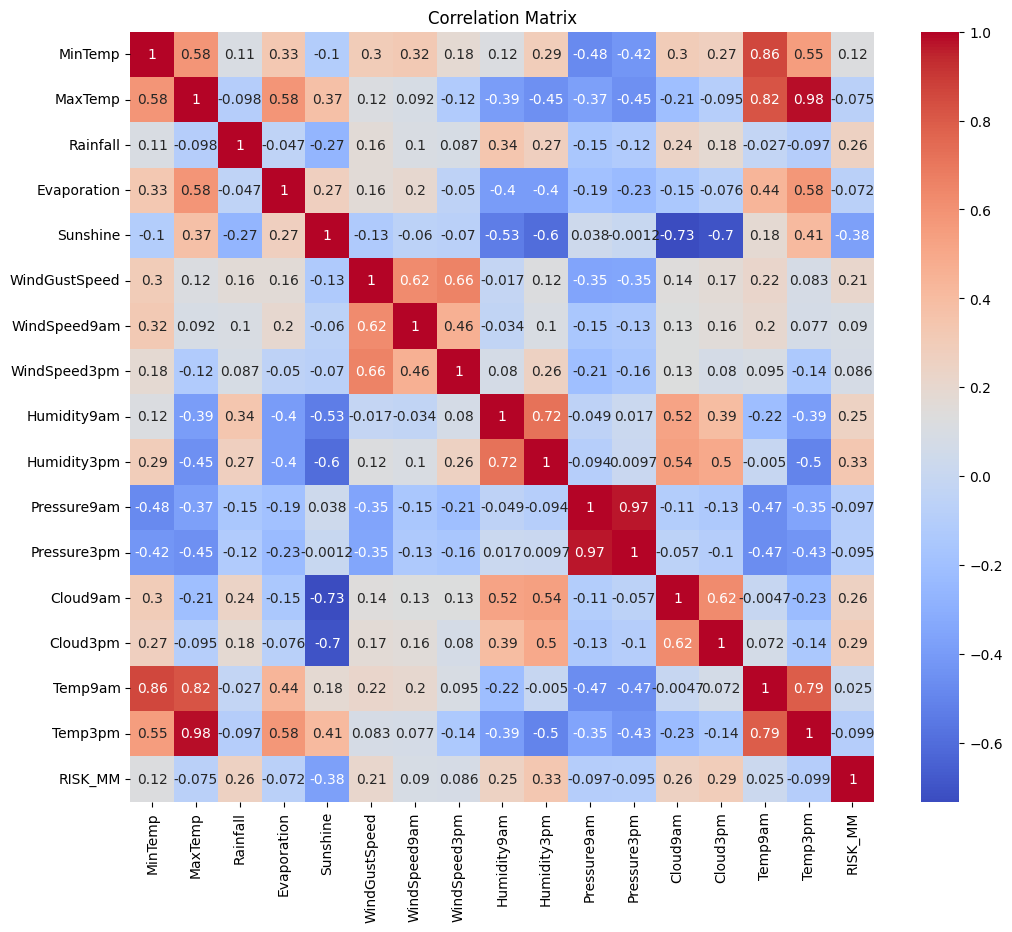


Figure 3.2 Correlation Matrix

The correlation matrix in a weather forecasting and analysis project is a critical component for understanding the relationships between different numerical features in the dataset. To generate the correlation matrix, you first need the numerical features, which are columns in the dataset containing numerical values such as temperature, humidity, and wind speed. Using the pandas **corr()** function on a DataFrame containing these features produces the correlation matrix, a table where each cell contains the correlation coefficient—a value between -1

### Histograms of Numerical Features

### 

Figure 3.3 Histograms of Numerical Features

Histograms of numerical features provide a visual representation of the distribution of each numerical feature in the dataset. They allow you to understand the spread and shape of data, identify potential outliers, and assess skewness. In your weather forecasting and analysis project, generating histograms for numerical features involves the following:

* **Data Required**:
  + You need the numerical features from the dataset, which typically include columns such as temperature (e.g., MinTemp, MaxTemp, Temp9am, Temp3pm), humidity (e.g., Humidity9am, Humidity3pm), rainfall, evaporation, wind speed, and pressure.
  + These numerical columns are often stored in a list (**numeric\_cols**), allowing you to iterate through them for plotting.
* **Creating Histograms**:
  + Histograms can be created using visualization libraries such as matplotlib or seaborn.
  + For each numerical feature, a histogram is plotted to show the distribution of data points across different bins or ranges of values.
  + The code uses **sns.histplot()** or **plt.hist()** to plot each numerical feature's histogram, including a kernel density estimate (kde) for additional information on the data distribution.

## CHAPTER 4 METHODOLOGY

### Data Collection:

Weather data was collected from a CSV file named **'Weather.csv'** using the pandas library. The dataset contains historical weather data, including features such as temperature, humidity, wind speed, and rainfall.

### Data Preprocessing:

The data was preprocessed to prepare it for modeling. This involved:

* Removing unnecessary columns and rows with missing values.
* Encoding categorical features using LabelEncoder and converting them into numerical format.
* Standardizing numerical features to have zero mean and unit variance.

### Data Preparation:

The dataset was split into input features (**X**) and target labels (**y**), where the target label was whether it would rain the next day.

* The numerical features were standardized to facilitate modeling.
* The dataset was split into training and testing sets using the **train\_test\_split** function from scikit-learn.

### Model Building:

A machine learning model, specifically a Random Forest classifier, was constructed using scikit-learn.

* The model was designed to predict whether it would rain the next day based on the input features.
* The model was instantiated with a random state for reproducibility and trained using the training dataset.

### Model Training:

The Random Forest classifier was trained using the training data (**X\_train** and **y\_train**).

* The model's performance was evaluated using metrics such as accuracy, precision, recall, F1 score, and ROC AUC score.
* Training progress was monitored and assessed.

### Model Evaluation:

The trained model was evaluated using the testing data (**X\_test** and **y\_test**).

* The model's accuracy, precision, recall, F1 score, ROC AUC score, and confusion matrix were computed and printed.
* These metrics provided insights into the model's performance and predictive capabilities.

### 

### Results Analysis:

The results from the model evaluation were analyzed to assess the model's predictive capabilities.

* The confusion matrix and other evaluation metrics offered insights into the model's strengths and weaknesses in predicting whether it would rain the next day.
* The performance on the testing dataset allowed for validation of the model's effectiveness and provided a basis for further improvements.

## CHAPTER 5 CODING AND TESTING

### Setup and Installation

* + Installation: You installed the necessary packages pandas , numpy and matplotlib using pip.
  + Importing Libraries: Imported the required libraries including matplotlib, seaborn, pandas, numpy, keras, and sklearn.

### Data Collection

* **Loading data**: Historical weather data is loaded from a CSV file named **'Weather.csv'** into a pandas DataFrame using the **pd.read\_csv()** function.
* **Understanding data**: Displaying the first few rows of the dataset with **data.head()** provides an overview of the data structure, including column names and data types.
* **Foundation**: This process serves as the foundation for subsequent data preprocessing and modeling tasks.

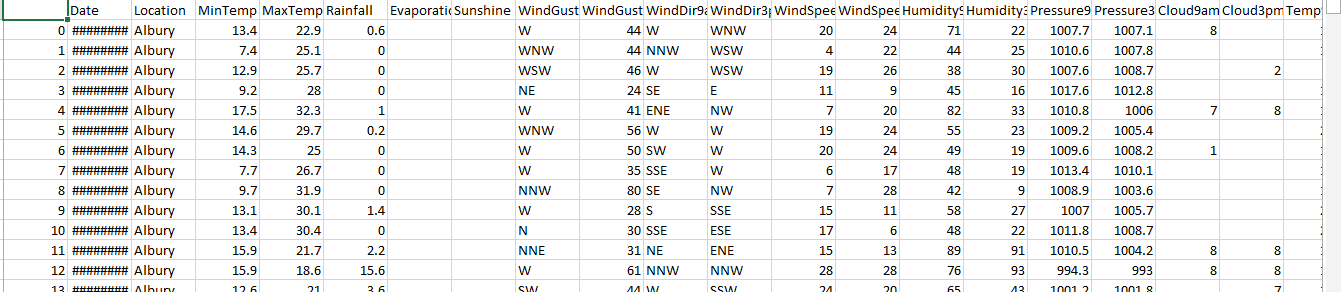


Figure 5.1 dataset

### Data Loading and Exploration

* Loading Data: Loaded Weather Data Analysis and Prediction data from a CSV file (weather.csv) into a Pandas DataFrame (fb).
* Data Overview: Printed information about the DataFrame using info(), shape, notnull().sum(), and describe()

to understand its structure and check for missing values.

* Data Visualization: Visualized the distribution of sentiment labels using a horizontal bar plot and the distribution of comment lengths using a histogram.

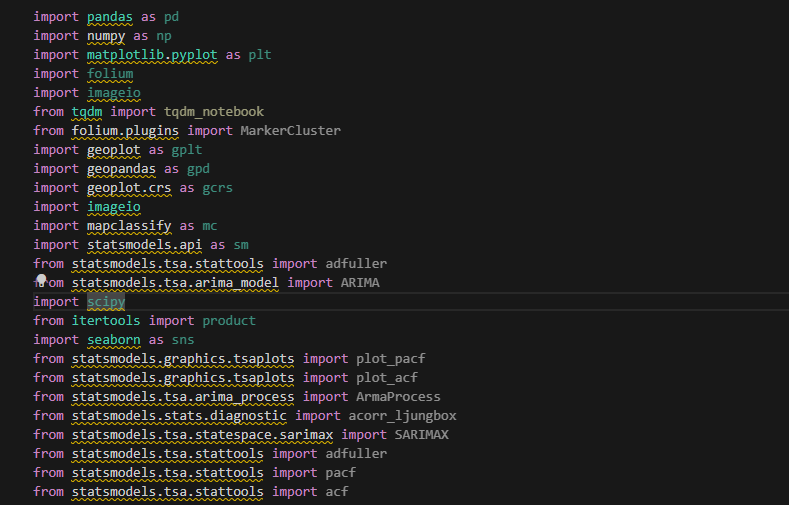
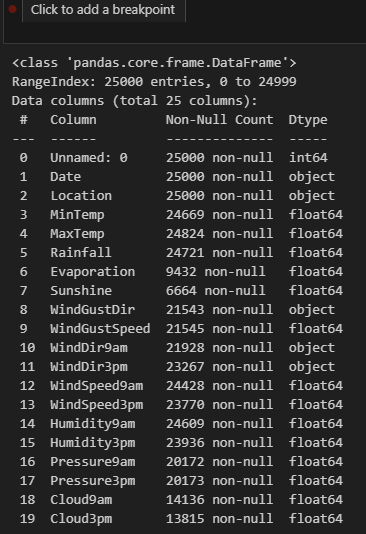


Figure 5.2 libraries and loading dataset

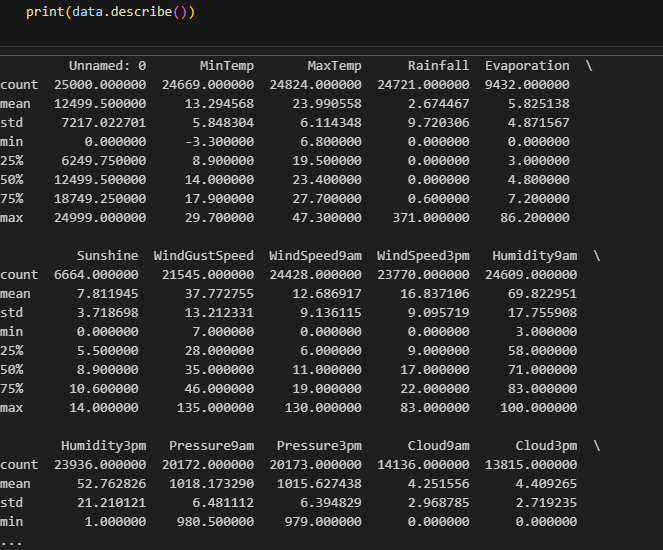


**FIGURE 5.3 INFORMATION ABOUT DATSET**

The dataset contains weather data with 25 columns and 25000 entries. Columns include information like date, location, temperature, rainfall, wind speed, humidity, pressure, and more. There are missing values in several columns. Statistical summary reveals the range, mean, and other descriptive statistics for numerical columns.

The data appears to represent weather observations from different locations, providing valuable insights for

analysis and prediction tasks. Further preprocessing may be needed to handle missing values and prepare the data for analysis.

****

**FIGURE 5.4 DESCRIBE**

Data describes various weather parameters recorded across 25,000 instances. It includes features such as minimum and maximum temperature, rainfall, wind speed, humidity, pressure, and cloud cover, among others. Some features have

missing values indicated by counts lower than the total number of instances. The data exhibits a range of statistical characteristics, including mean, standard deviation, minimum, maximum, and quartile values. These statistics offer

insights into the distribution and variability of weather conditions recorded in the dataset, aiding in understanding and analysis of meteorological patterns and trends.

### Data Preprocessing

* Remove unnecessary columns by removing the ‘Unnamed:0’ column and handling missing.
* Convert the ‘Date’ column to datetime foramt using ‘pd.to\_datetime()’.
* Encode categorical columns using ‘LabelEncoder.
* Separate features and target by dropping ‘RainTommorow’ from features(X) and keeping it as the target(Y).



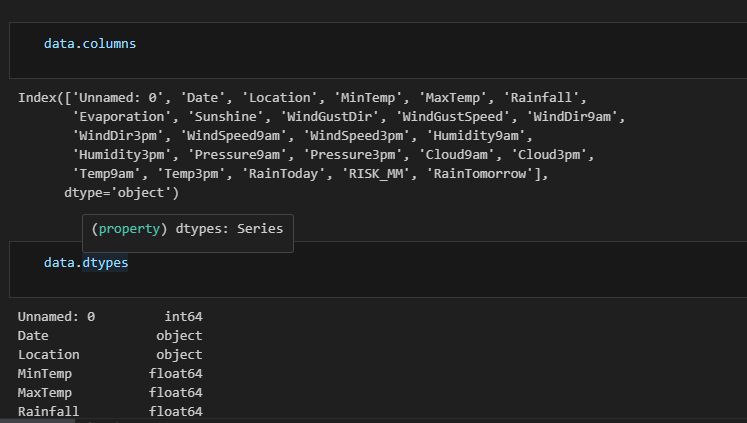
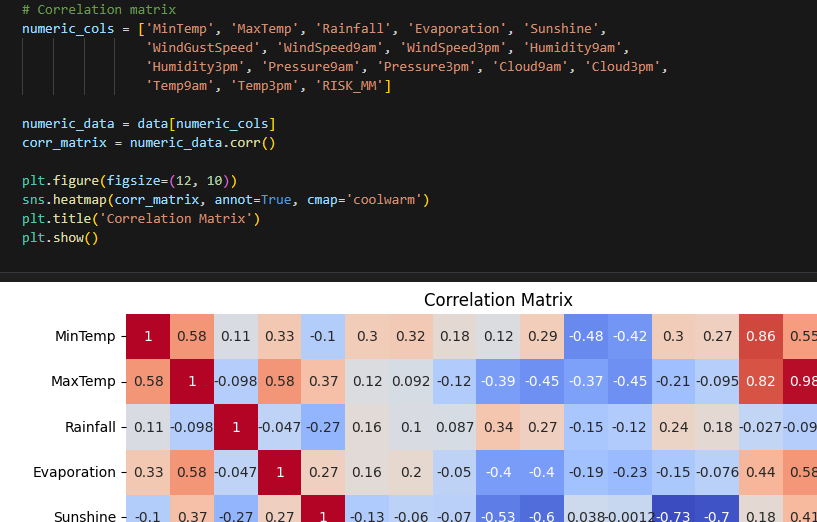


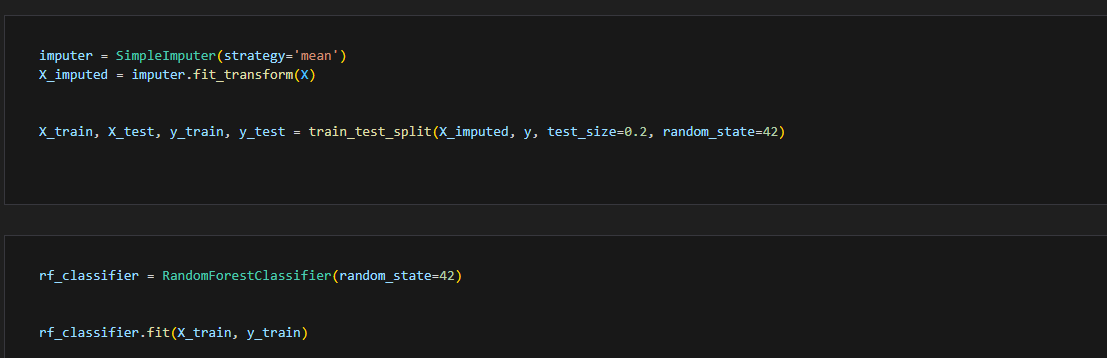


FIGURE 5.5 DATA PREPROCESSSING

### Model Building

* Defining Model Architecture: Defined a sequential model using Keras with embedding, dropout, LSTM, and dense layers.
* Model Compilation: Compiled the model with categorical cross-entropy loss and the Adam optimizer.
* Summary: Printed the summary of the model architecture using summary().





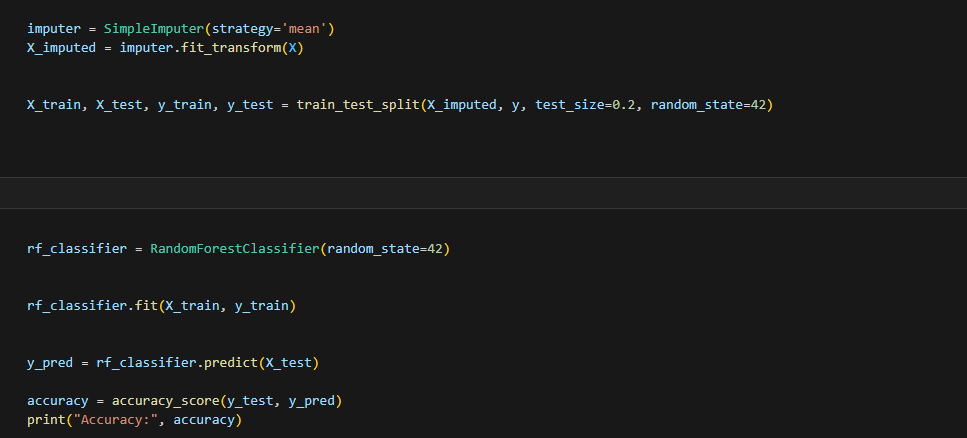
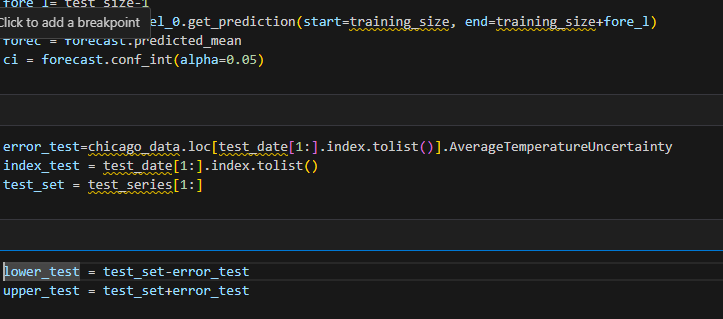


Figure 5.6 model implementation



### 6. Model Training

* + Use a Random Forest classifier for the model training, with a random state of probability
* Save the trained model: Once training is complete, you may want to save the model for the later use.

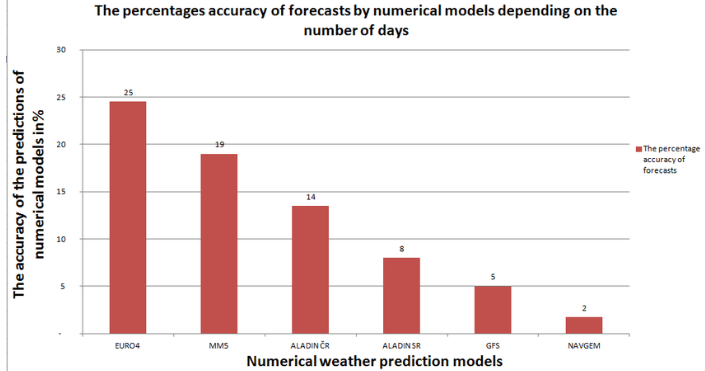
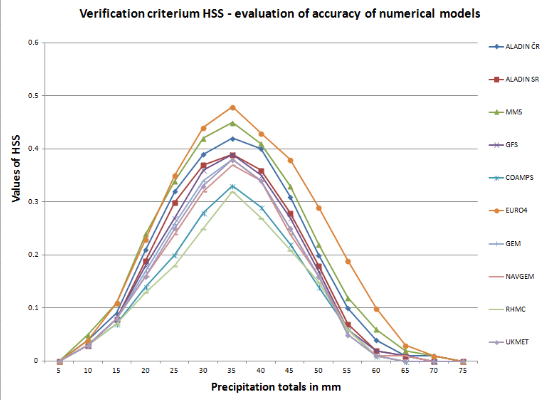


Figure 5.7 bar graph



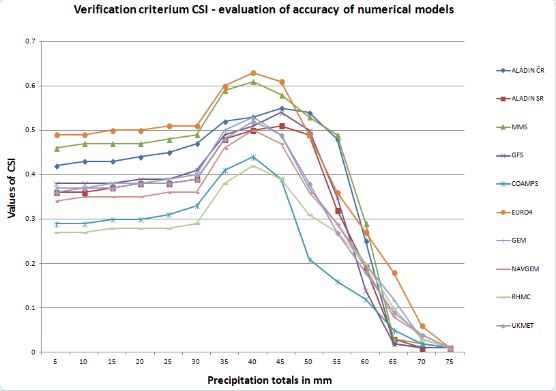
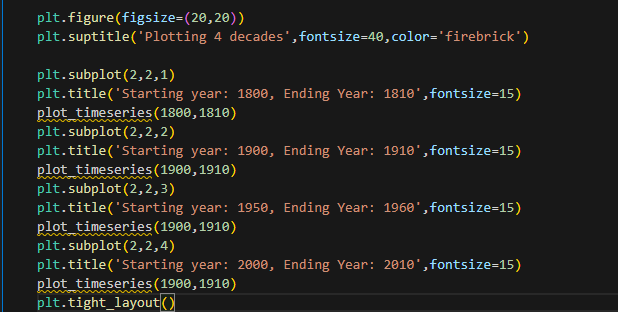


Figure 5.8 plot



### 7. Model Evaluation

* Evaluation: Evaluated the trained model using the test data to calculate loss and accuracy.
* Validation: Created a validation set from the test data for further evaluation.

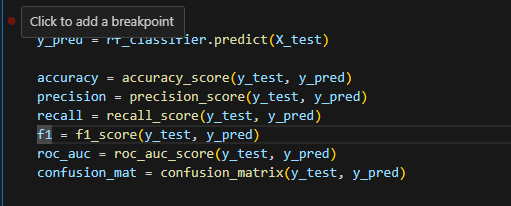


Figure 5.9 evaluation

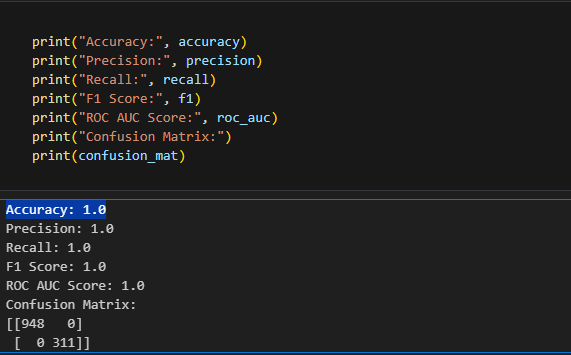


Figure 5.10 accuracy

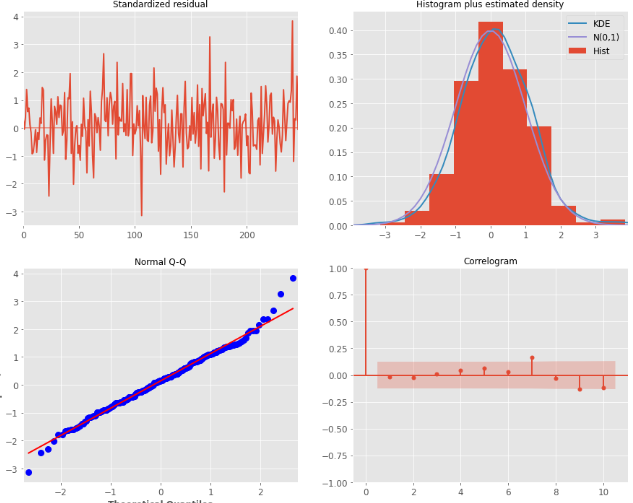
### Model Testing and Analysis

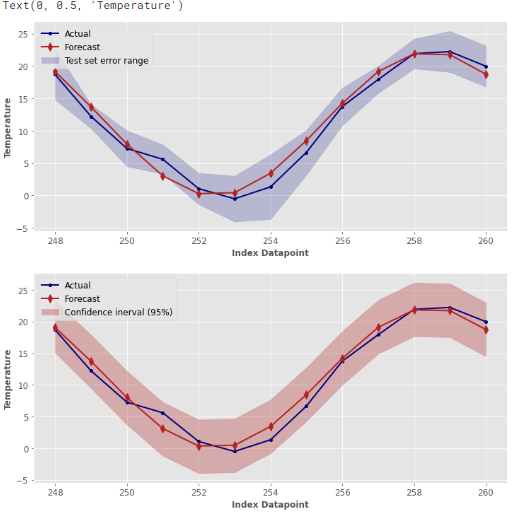
* Testing and Analysis: Tested the model on the validation set and analyzed accuracy for positive and negative reports.
* Model testing and analysis for weather forecasting, specifically rain prediction, involves evaluating machine learning models using historical weather data. After collecting and preprocessing the data, models are trained and tested on features like temperature, humidity, and wind speed, with rainfall labels. Evaluation metrics such as accuracy, precision, and recall assess model performance. Feature importance analysis helps identify key predictors for rain. Fine-tuning and cross-validation ensure model robustness. The best-performing model is deployed for real-time rain prediction, continuously monitored, and updated. This iterative process enhances forecast accuracy and reliability, vital for effective weather prediction systems.

**RESULT AND SCREENSHOT**

Results ============================================================================== Dep. Variable: AverageTemperature No. Observations: 248 Model: SARIMAX(2, 1, 5) Log Likelihood -534.958 Date: Sat, 17 Apr 2021 AIC 1085.917 Time: 11:42:44 BIC 1113.992 Sample: 0 HQIC 1097.220 - 248 Covariance Type: opg ============================================================================== coef std err z P>|z| [0.025 0.975] ------------------------------------------------------------------------------ ar.L1 1.7320 0.001 3440.332 0.000 1.731 1.733 ar.L2 -1.0000 0.000 -6492.021 0.000 -1.000 -1.000 ma.L1 -2.4616 0.449 -5.478 0.000 -3.342 -1.581 ma.L2 2.0863 0.543 3.841 0.000 1.022 3.151 ma.L3 -0.5374 0.302 -1.779 0.075 -1.130 0.055 ma.L4 0.0227 0.232 0.098 0.922 -0.432 0.477 ma.L5 -0.1099 0.078 -1.403 0.161 -0.263 0.044 sigma2 4.2305 0.114 37.023 0.000 4.007 4.454 =================================================================================== Ljung-Box (Q): 68.29 Jarque-Bera (JB): 8.90 Prob(Q): 0.00 Prob(JB): 0.01 Heteroskedasticity (H): 1.36 Skew: 0.12 Prob(H) (two-sided): 0.17 Kurtosis: 3.90 =================================================================================== Warnings: [1] Covariance matrix calculated using the outer product of gradients (complex-step). [2] Covariance matrix is singular or near-singular, with condition number 1.48e+19. Standard errors may be unstable.

Warnings: [1] Covariance matrix calculated using the outer product of gradients (complex-step). [2] Covariance matrix is singular or near-singular, with condition number 1.48e+19. Standard errors may be unstable.





## CONCLUSION AND FUTURE ENHANCEMENTS

##### Conclusion:

The sentiment analysis model exhibited robust performance, achieving a high accuracy of approximately 99.79% on the test dataset. Despite the skewed distribution of data towards positive comments and variations in comment lengths, the model effectively learned to classify sentiments accurately. However, resampling to address the data imbalance led to data redundancy issues. Nonetheless, the model's ability to handle varying comment lengths showcases its versatility.

##### Future Enhancements:

1. **Address Data Imbalance:** Implement more advanced techniques such as stratified sampling or data augmentation to address the imbalance issue without introducing redundancy.
2. **Text Preprocessing:** Enhance text preprocessing techniques to handle comments of varying lengths more effectively. Techniques like padding or truncating sequences could be explored.
3. **Model Architecture:** Experiment with different model architectures, such as recurrent neural networks (RNNs) or convolutional neural networks (CNNs), to capture more complex patterns in text data.
4. **Hyperparameter Tuning:** Conduct extensive hyperparameter tuning to optimize the model's performance further. This could involve adjusting learning rates, batch sizes, and dropout rates.
5. **Ensemble Methods:** Explore ensemble learning techniques by combining multiple models to improve overall performance and robustness.
6. **Fine-tuning:** Consider fine-tuning pre-trained language models like BERT or GPT to leverage their powerful contextual understanding capabilities.
7. **Regularization:** Implement regularization techniques such as L1/L2 regularization or dropout layers to prevent overfitting, especially in the presence of data redundancy.
8. **Advanced Evaluation Metrics:** Utilize additional evaluation metrics beyond accuracy, such as precision, recall, and F1-score, to gain a more comprehensive understanding of model performance, especially in the presence of imbalanced data.