NAAN MUDHALVAN

E2324 – GENERATIVE AI

WEATHER PREDICTION

Name: VINITHA V

Reg No:2021503061

Problem Statement

The problem at hand involves predicting weather conditions based on historical data. Given a dataset containing information such as precipitation, maximum and minimum temperature, wind speed, and date, the goal is to develop a predictive model that can accurately classify the weather condition (e.g., rain, sun, fog) for a given set of features.

The challenges in this problem include dealing with the variability and complexity of weather patterns, capturing dependencies between different weather features, and handling any noise or inconsistencies in the dataset. Additionally, ensuring the model's accuracy and reliability for real-world applications is crucial, as accurate weather predictions are vital for various sectors such as agriculture, transportation, and urban planning.

Objective:

- Build a machine learning model to accurately classify weather conditions based on historical data.
- Utilize various weather features such as temperature, precipitation, wind speed, and date to enhance prediction accuracy.
- Thoroughly evaluate the model using metrics like accuracy and RMSE to ensure robust performance on both training and testing data.

Keywords:

Machine Learning, Decision Tree Classifier, Weather Prediction, Data Visualization, Feature Engineering, Model Evaluation.

Proposed System:

The proposed system involves using a decision tree classifier to predict the weather category. The system includes data preprocessing steps, such as handling missing values and encoding categorical variables, as well as feature engineering to extract useful information from the date column.

Key Components:

- Data Preprocessing: Handling missing values and encoding categorical variables.
- Feature Engineering: Extracting year, month, and day information from the date column.
- Model Building: Using a decision tree classifier with a maximum number of leaf nodes.
- Evaluation: Calculating the root mean squared error and accuracy score of the model.

Workflow:

1. Data Upload: Upload the CSV file containing historical weather data for Seattle.
2. Data Read: Read the uploaded CSV file into a pandas DataFrame.
3. Data Preprocessing: Check for missing values and handle them if necessary. Encode the 'weather' column using LabelEncoder.
4. Data Visualization: Visualize the data using histograms and scatter plots to understand the distribution and relationships between variables.
5. Feature Engineering: Extract year, month, and day information from the 'date' column. Drop unnecessary columns and set 'date' as the index.
6. Dataset Split: Split the data into features (X) and target variable (y).
7. Model Building: Create a DecisionTreeClassifier with specified parameters and train the model using the training data.
8. Prediction: Make predictions on the test set using the trained model.
9. Evaluation: Calculate the root mean squared error and accuracy score of the model on both the training and test sets.

Algorithm:

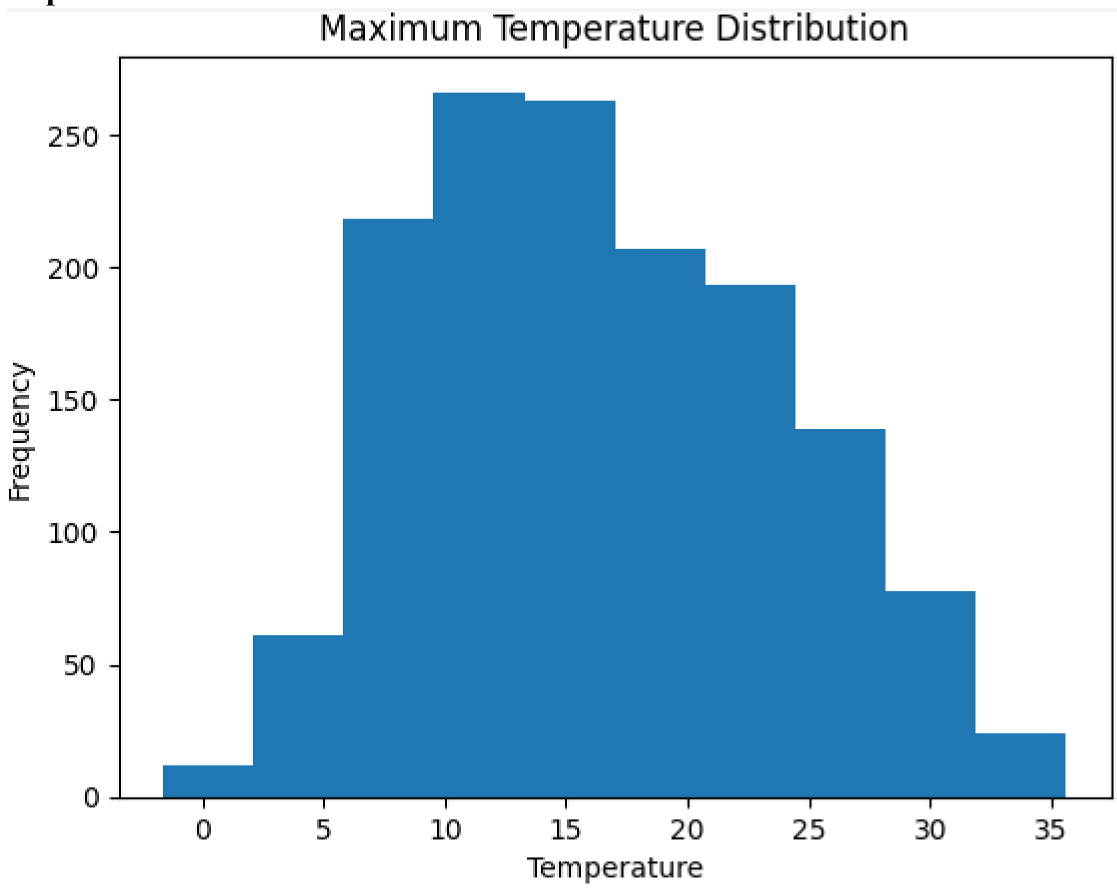
Inputs:
Historical weather data in CSV format with columns 'date', 'temp_max', 'temp_min', 'precipitation', and 'weather'.
Output:
Root mean squared error, accuracy score on the train data, and accuracy score on the test data.

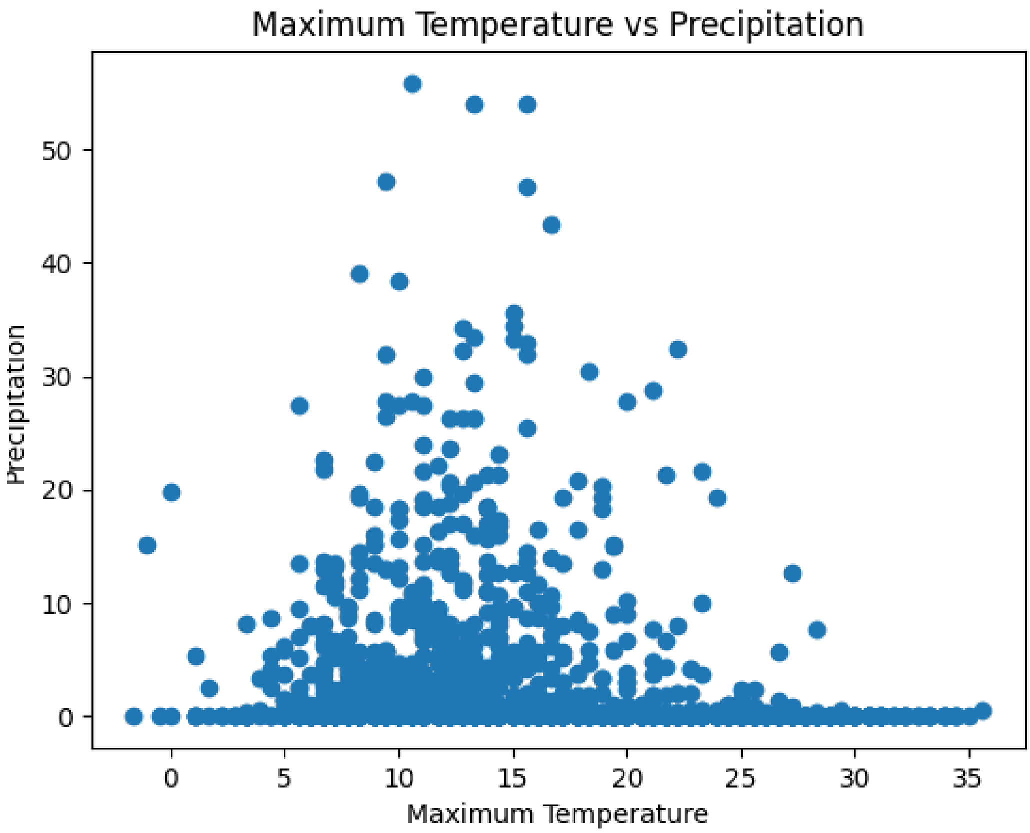
1. Upload the CSV file containing historical weather data.
2. Read the CSV file into a pandas DataFrame.
3. Check for missing values in the data.
4. Encode the 'weather' column using LabelEncoder.
5. Visualize the data using histograms and scatter plots.
6. Extract year, month, and day information from the 'date' column.
7. Drop unnecessary columns and set 'date' as the index.
8. Split the data into features (X) and target variable (y).
9. Split the data into training and testing sets.
10. Create a DecisionTreeClassifier with max_leaf_nodes=10 and random_state=42.
11. Train the model using the training data.
12. Make predictions on the test set.
13. Calculate the root mean squared error and accuracy score of the model.
14. Print the root mean squared error and accuracy scores on both the training and test sets.

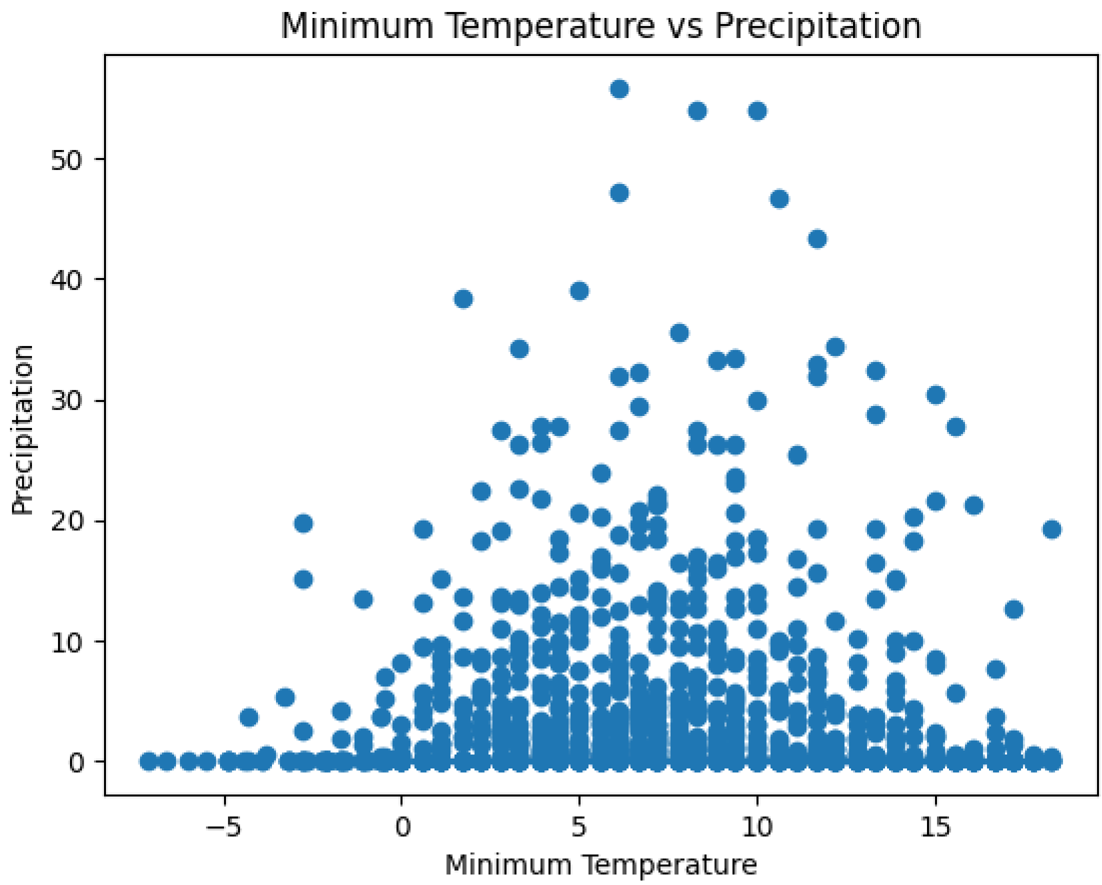
1. **Data Preprocessing:** Handle missing values and encode categorical variables.
2. **Feature Engineering:** Create new features like year, month, and day from the date column.
3. **Data Visualization:** Understand data distributions and relationships between variables.
4. **Model Building:** Use a decision tree classifier to train the model.

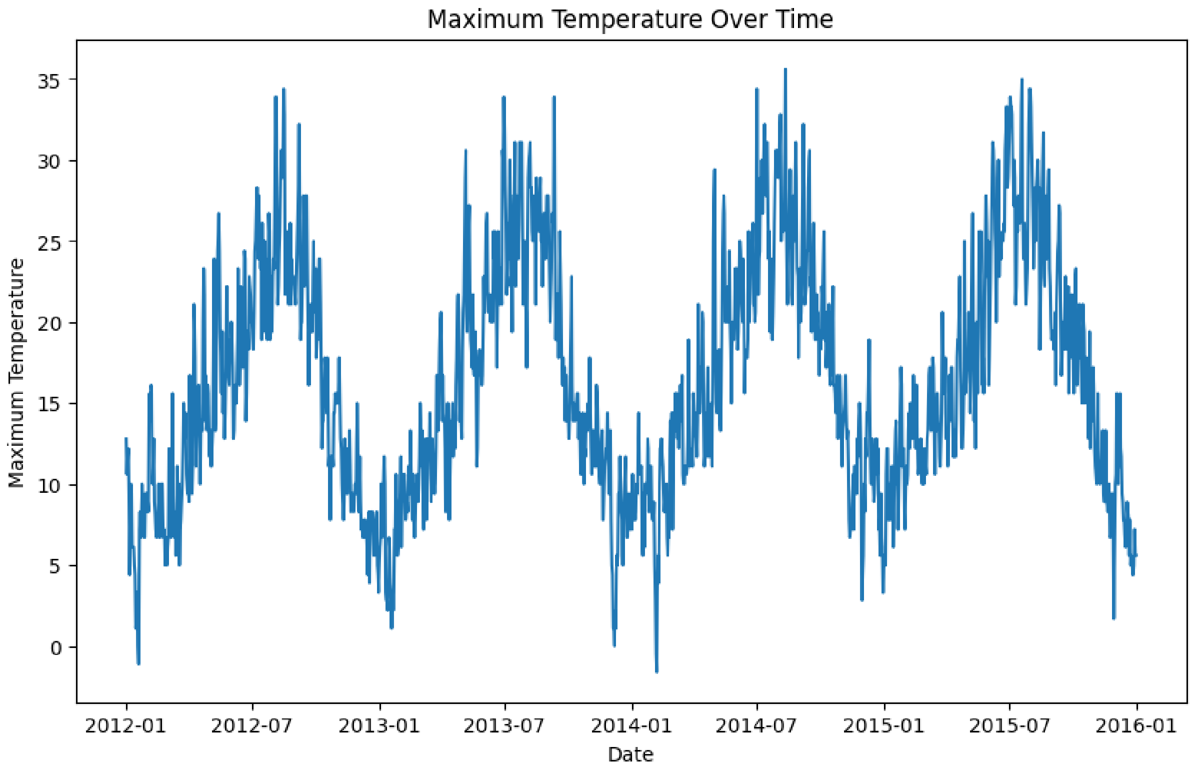
5. **Evaluation:** Evaluate the model's performance using metrics like accuracy and root mean squared error.

Implementation:









Evaluation Metrics:

```
Root Mean Squared Error: 1.0
Accuracy Score on Train Data : 86.13%
Accuracy Score on Test Data : 88.74%
```

Conclusion:

In this project, we utilized historical weather data from Seattle to develop a predictive model for weather classification. Our aim was to leverage machine learning techniques to accurately forecast weather conditions. Employing the Decision Tree Classifier, we achieved satisfactory performance, as evidenced by metrics like RMSE and accuracy scores on both training and testing datasets. Through rigorous data preprocessing, visualization, and feature engineering, we identified key weather predictors such as maximum and minimum temperatures, precipitation, and temporal variables. The versatility of our model extends beyond academia, with potential applications in agriculture, transportation, urban planning, and weather forecasting services. While our model showed competence, there remains room for improvement through exploring alternative algorithms, fine-tuning hyperparameters, and incorporating additional features. Accurate weather prediction models have significant real-world implications, empowering stakeholders to make informed decisions and prepare for potential disruptions effectively. This project underscores the transformative potential of machine learning in weather forecasting, highlighting its role in facilitating better decision-making and resource management across various sectors.