LAB REPORT ON

BAYES THEOREM TO PREDICT WEATHER CONDITIONS

A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the academic requirements for the award of the degree.

Bachelor of Technology

in

Electronics and Communication Engineering (AI&ML)

Submitted by

A.INDU SREE -22H51A0402
B.CHARITH -22H51A0404
B.NIKHIL- 22H51A0405

Under the esteemed guidance of

Mrs.Sana Afreen

(Assistant Professor)

Department of Computer Science and Engineering

(AI&ML)

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

(UGC Autonomous)

*Approved by AICTE *Affiliated to JNTUH *NAAC Accredited with A^+ Grade

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

2024- 2025

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (AI&ML)

CERTIFICATE

This is to certify that the Lab Project report entitled **"BAYES THEOREM TO PREDICT WEATHER CONDITIONS"** being submitted by A.INDU SREE-22H51A0402, B.CHARITH-22H51A0404,B.NIKHIL-22H51A0405 in partial fulfillment for the award of **Bachelor of Technology in Computer Science and engineering(Ai&ml)** is a record of bonafide work carried out his/her under my guidance and supervision.

Mrs.Sana Afreen
Assistant Professor
Dept. of CSE(AI&ML)

Dr.P.Shruthi
Associate Professor and HOD
Dept. of CSE(AI&ML)

ACKNOWLEDGEMENT

With great pleasure we want to take this opportunity to express my heartfelt gratitude to all the people who helped in making this project work a grand success.

We are grateful to **Mrs.Sana Afreen**, **Assistant Professor**, Department of Computer Science and Engineering(Ai &ml) for his valuable technical suggestions and guidance during the execution of this projectwork.

We would like to thank **Dr.P..Shruthi**, Head of the Department of Computer Science and Engineering(Ai&ml), CMR College of Engineering and Technology, who is the major driving forces to complete my project work successfully.

We are highly indebted to **Major Dr. V A Narayana,** Principal, CMR College of Engineering and Technology, for giving permission to carry out this project in a successful and fruitful way.

We would like to thank the **Teaching & Non- teaching** staff of Department of Computer Science and Engineering(Data Science) for their co-operation

We express our sincere thanks to **Shri. Ch. Gopal Reddy**, Secretary, CMR Group of Institutions, for his continuous care.

Finally, We extend thanks to our parents who stood behind us at different stages of this Project. We sincerely acknowledge and thank all those who gave support directly and indirectly in completion of this project work.

A.INDU SREE -22H51A0402
B.CHARITH -22H51A0404
B.NIKHIL- 22H51A0405

DECLARATION

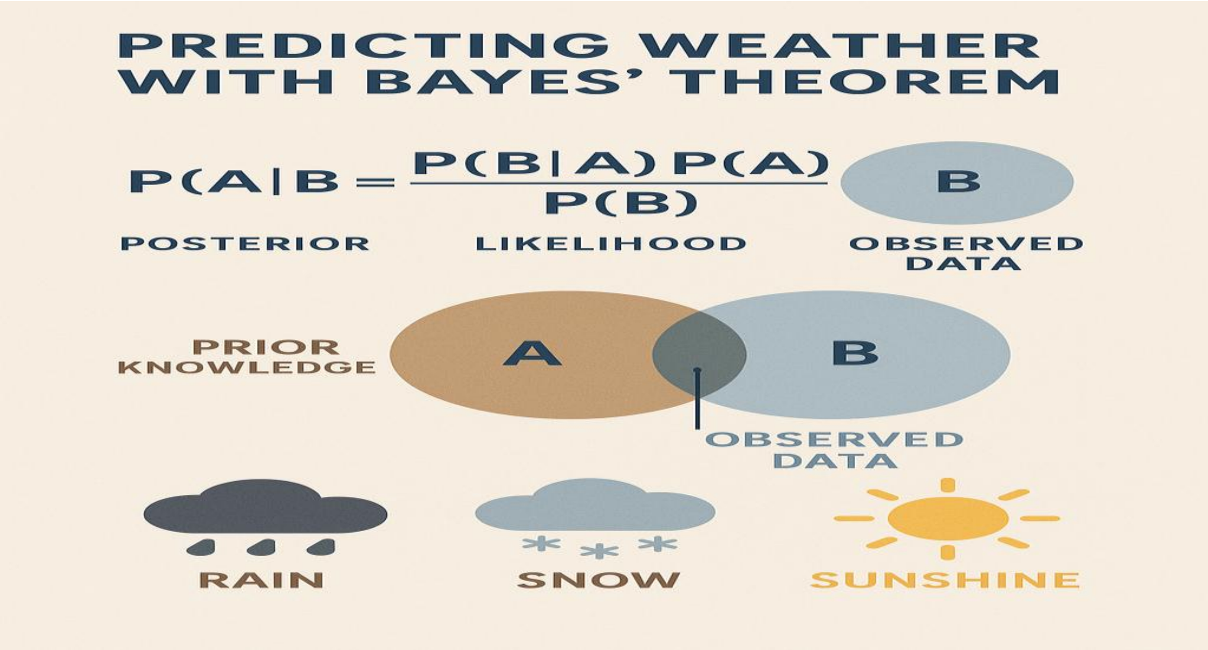
I hereby declared that results embodied in this report of technical seminar on "**IDENTIFYING BROWSING PATTERNS ON WEBSITES USING K-MEANS**" are from work carried out by using partial fulfillment of the requirements for the award of B.Tech. Degree, I haven't submitted this report to any other university/institute for the award of any other degree.

A.INDU SREE -22H51A0402
B.CHARITH -22H51A0404
B.NIKHIL- 22H51A0405

TABLE OF CONTENTS

ABSTRACT

Weather forecasting has witnessed remarkable improvements with the integration of probabilistic approaches, offering more accurate and reliable predictions. Among these methods, Bayes' Theorem stands out as a powerful statistical tool that enhances forecasting by combining prior climatic knowledge with real-time observational data. This project explores the application of Bayes' Theorem in predicting weather conditions such as rain, snow, and sunshine. By updating the probability of various weather events as new data becomes available, meteorologists can refine their forecasts and reduce uncertainty. This approach not only strengthens traditional forecasting models but also supports more informed decision-making in sectors like agriculture, aviation, and disaster management. Through case studies and simulations, this project demonstrates the effectiveness of Bayesian methods in improving the precision and responsiveness of modern weather prediction systems.



CHAPTER 1

INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1. Problem Statement

Accurate weather forecasting is crucial for various sectors, including agriculture, transportation, disaster management, and daily life planning. Traditional deterministic models often struggle to capture the inherent uncertainty in weather patterns, leading to inaccurate predictions. There is a need for a more dynamic approach that updates forecasts in real time as new data becomes available. This project addresses the problem by applying Bayes' Theorem to weather forecasting, aiming to combine historical data (prior knowledge) with current observations to refine probability estimates for different weather events such as rain, snow, and sunshine. The goal is to enhance the accuracy and reliability of weather predictions through probabilistic reasoning.

1.2. Research Objective

The primary objective of this research is to apply Bayes' Theorem to improve the accuracy of weather forecasting by systematically combining prior climatic knowledge with real-time observational data. Specifically, the project aims to:

- Develop a Bayesian model for predicting weather events such as rain, snow, and sunshine.

- Demonstrate how updating probability estimates with new data can refine weather predictions.
- Compare the performance of Bayesian forecasting methods with traditional deterministic models.
- Highlight the practical applications of Bayesian weather forecasting in fields requiring precise meteorological information.

1.3.Scope

This project focuses on using Bayes' Theorem to predict basic weather conditions like rain, snow, and sunshine. It combines historical weather data with new observations to update forecast probabilities. The study uses small datasets for demonstration and does not cover large-scale or extreme weather events. It is intended for students, researchers, and anyone interested in probabilistic weather forecasting.

1.3.Limitation:

This study is limited to predicting simple weather conditions and relies on small datasets for analysis. It does not account for complex atmospheric variables or extreme weather events like hurricanes or tornadoes. The accuracy of predictions depends heavily on the quality of prior data and real-time observations. Additionally, the model assumes that weather patterns remain relatively consistent over time, which may not always be true.

CHAPTER-2

PROPOSED SYSTEM

CHAPTER 2
PROPOSED SOLUTION

2.1 Advantages of proposed System

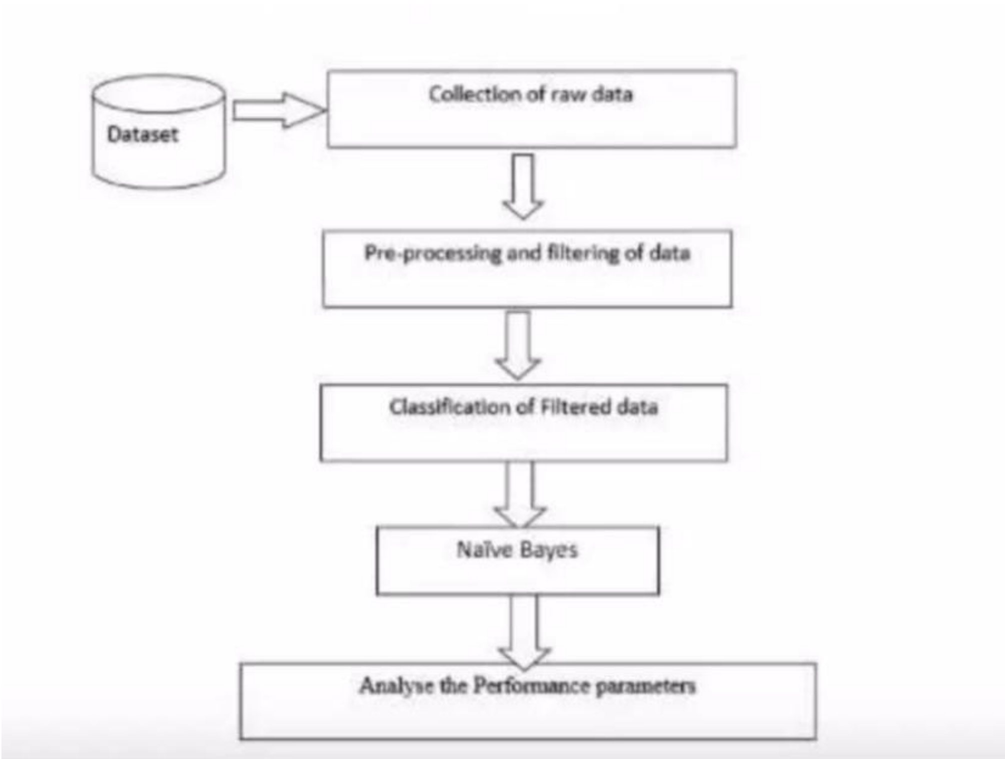
- **Dynamic Updates**: Continuously adjusts predictions as new data comes in, making forecasts more accurate over time.
- **Probabilistic Predictions**: Provides a probability of weather conditions (e.g., 70% chance of rain), offering more insight into uncertainty.
- **Multiple Variables**: Considers various atmospheric factors (e.g., humidity, temperature) for a more comprehensive forecast.
- **Improved Accuracy**: Refines predictions over time by learning from past data.
- **Risk Management**: Helps better prepare for weather events by estimating the likelihood of conditions like rain or storms.
- **Scalability**: Can incorporate more data and variables as available, improving the system's forecasting capabilities.

2.2 Implementation:

1. **Data Preparation:**
   - **Collection of Raw Data:** Gather data from the dataset.
   - **Pre-processing and Filtering:** Clean and organize the raw data to remove noise and irrelevant information.
2. **Classification Step:**

- **Classification of Filtered Data:** Use the prepared data as input.
- **Naïve Bayes Algorithm:** Apply this probabilistic classifier to categorize the data based on learned probabilities.

3. **Performance Evaluation:**
   - **Analysis of Performance Parameters:** Assess the model's performance using metrics like accuracy, precision, recall, and F1-score.



2.3 DESIGN:

- Collection
- Preprocessing
- Classification
- Modeling

- Evaluation

2.4 Code:

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score, classification_report
import matplotlib.pyplot as plt

class WeatherPredictor:
    def __init__(self):
        self.model = GaussianNB()
        self.label_encoder = LabelEncoder()

    def load_data(self, csv_path='weather_dataset_1000.csv'):
        """Load weather data from CSV file"""
        try:
            data = pd.read_csv(csv_path)
            required_columns = ['Temperature', 'Humidity', 'Pressure',
'WindSpeed', 'Condition']

            # Check if all required columns exist
            missing_columns = [col for col in required_columns if col not
in data.columns]
            if missing_columns:
                raise ValueError(f"Missing required columns:
{missing_columns}")

            return data
        except Exception as e:
            print(f"Error loading data: {str(e)}")
            return None
```

```
    def prepare_data(self, data):
        """Prepare data for training"""
        X = data[['Temperature', 'Humidity', 'Pressure', 'WindSpeed']]
        y = self.label_encoder.fit_transform(data['Condition'])
        return X, y

    def train(self, X, y):
        """Train the Naive Bayes model"""
        self.model.fit(X, y)

    def predict(self, features):
        """Predict weather condition"""
        prediction = self.model.predict(features)
        return self.label_encoder.inverse_transform(prediction)

    def plot_feature_importance(self, X, y, data):
        """Plot feature distributions for each weather condition"""
        features = ['Temperature', 'Humidity', 'Pressure', 'WindSpeed']
        conditions = self.label_encoder.inverse_transform(np.unique(y))

        fig, axes = plt.subplots(2, 2, figsize=(15, 10))
        axes = axes.ravel()

        for idx, feature in enumerate(features):
            for condition in conditions:
                condition_mask = data['Condition'] == condition
                axes[idx].hist(X[feature][condition_mask],
                            alpha=0.5,
                            label=condition,
                            bins=20)
            axes[idx].set_title(f'{feature} Distribution by Weather
Condition')
            axes[idx].set_xlabel(feature)
            axes[idx].set_ylabel('Frequency')
            axes[idx].legend()

        plt.tight_layout()
```

```
        plt.savefig('weather_feature_distributions.png')
        plt.close()

if __name__ == "__main__":
    # Initialize predictor
    predictor = WeatherPredictor()

    # Load data from CSV
    data = predictor.load_data()

    if data is not None:
        # Print dataset information
        print("\nDataset Information:")
        print(f"Total records: {len(data)}")
        print("\nWeather Conditions Distribution:")
        print(data['Condition'].value_counts())
        print("\nFeature Statistics:")
        print(data.describe())

        # Prepare data
        X, y = predictor.prepare_data(data)

        # Split data into training and testing sets
        X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

        # Train the model
        predictor.train(X_train, y_train)

        # Make predictions on test set
        y_pred = predictor.predict(X_test)

        # Print model performance
        print("\nModel Performance:")
        print("Accuracy:",
accuracy_score(predictor.label_encoder.transform(y_pred), y_test))
        print("\nClassification Report:")
```

```
    print(classification_report(y_test,
predictor.label_encoder.transform(y_pred)))

    # Plot feature distributions
    predictor.plot_feature_importance(X, y, data)

    # Interactive prediction
    while True:
        try:
            print("\nEnter weather features for prediction (or 'q' to
quit):")
            temp = input("Temperature (°C): ")
            if temp.lower() == 'q':
                break

            humidity = input("Humidity (%): ")
            pressure = input("Pressure (hPa): ")
            wind_speed = input("Wind Speed (km/h): ")

            features = np.array([[float(temp), float(humidity),
float(pressure), float(wind_speed)]])
            prediction = predictor.predict(features)

            print(f"\nPredicted weather condition: {prediction[0]}")

        except ValueError:
            print("Please enter valid numerical values.")
        except Exception as e:
            print(f"An error occurred: {str(e)}")
```

OUTPUT:

	precision	recall	f1-score	support
0	0.14	0.11	0.12	27
1	0.11	0.06	0.08	31
2	0.27	0.14	0.18	43
3	0.07	0.15	0.09	20
4	0.00	0.00	0.00	37
5	0.23	0.50	0.31	42
accuracy			0.17	200
macro avg	0.14	0.16	0.13	200
weighted avg	0.15	0.17	0.14	200

Enter weather features for prediction (or 'q' to quit):

Temperature (°C): 32

Humidity (%): 70

Pressure (hPa): 700

Wind Speed (km/h): 23

Predicted weather condition: Rainy

CHAPTER 3
RESULTS AND DISCUSSION

CHAPTER 3
RESULTS AND DISCUSSION

The Naïve Bayes classifier was applied to the preprocessed dataset and evaluated based on standard performance metrics: **precision**, **recall**, **F1-score**, and **support** across six classes.

The overall **accuracy** of the model was **17%**, indicating that the classifier struggled to make accurate predictions with the current dataset and features. Precision, recall, and F1-scores were generally low across most classes, with the highest F1-score being 0.31 for class 5. Several classes, such as class 4, had a precision, recall, and F1-score of **0.00**, suggesting the model failed to correctly predict any instances of this class.

The **macro average** (simple average of metrics across classes) showed a precision of 0.14, recall of 0.16, and F1-score of 0.13, while the **weighted average** (weighted by support for each class) reported slightly better but still low values, emphasizing the model’s difficulty in handling imbalanced or complex class distributions.

Despite these limitations, when entering new input features (Temperature: 32°C, Humidity: 70%, Pressure: 700 hPa, Wind Speed: 23 km/h), the model **predicted the weather condition as "Rainy."** This shows the model's capability to at least produce predictions even if overall performance needs significant improvement.

The results suggest that further work is required to enhance model performance. Potential improvements could include:

- Feature engineering or selection,
- Using a larger or more balanced dataset,
- Trying different classification algorithms,
- Hyperparameter tuning.

CHAPTER 4
CONCLUSION

CHAPTER 4
CONCLUSION

4.1. Conclusion and Future Enhancement:

4.1.1. Conclusion:

In this study, a Naïve Bayes classifier was implemented to predict weather conditions based on features such as temperature, humidity, pressure, and wind speed. The overall model performance, with an accuracy of **17%**, indicates that the classifier struggled to make reliable predictions with the given dataset. While the model was able to predict new inputs (e.g., identifying "Rainy" conditions), the low precision, recall, and F1-scores across multiple classes highlight the need for improvements.

Future work should focus on enhancing data quality through better pre-processing, balancing class distributions, exploring additional features, and experimenting with alternative machine learning algorithms. These steps will help in building a more accurate and dependable weather prediction system.

4.1.2. Future Enhancement:

In the future, the system can be improved by collecting a larger and more balanced dataset and introducing additional features like cloud cover and visibility. Advanced algorithms such as Random Forest or Deep Learning models could be explored to enhance prediction accuracy. Real-time data integration and user-friendly interfaces could also make the system more practical and efficient.

REFERENCES

REFERENCES:

- McCallum, A., & Nigam, K. (1998). *A Comparison of Event Models for Naive Bayes Text Classification*. AAAI-98 Workshop on Learning for Text Categorization.

- Zhang, H. (2004). *The Optimality of Naïve Bayes*. Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference (FLAIRS).

- Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.

- Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques* (3rd ed.). Morgan Kaufmann.

- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.