# **Machine learning Assignment 2**

Submitted To

Professor Navneet Goyal

Course Name: Machine Learning BITS F464

# 

BITS Pilani

Submitted By **Group 9**:

| **Name** | **BITS ID** |
| --- | --- |
| Rishika Kalra | 2021A3PS2651P |
| Ajey Malik | 2021A7PS2542P |
| Sanskar Kaushik | 2020A8PS1562P |

Birla Institute of Technology and Science, Pilani Campus, India

**DATASET :-**

We have used the following weather dataset taken from Kaggle for our assignment.

<https://www.kaggle.com/datasets/muthuj7/weather-dataset/data>

It is a weather dataset that contains historical weather data collected from a weather station. Here's an overview of the dataset columns :-

1. Formatted Date: The date and time of the observation.
2. Summary: A brief summary of the weather conditions, such as "Partly Cloudy", "Mostly Cloudy", "Clear", etc.
3. Precip Type: Type of precipitation, such as "rain" or "snow".
4. Temperature (C): The temperature measured in Celsius.
5. Apparent Temperature (C): The apparent temperature, which is the temperature perceived by humans, often influenced by factors like humidity and wind.
6. Humidity: The relative humidity, indicating the amount of moisture present in the air.
7. Wind Speed (km/h): The speed of the wind is measured in kilometers per hour.
8. Wind Bearing (degrees): The direction from which the wind is blowing, measured in degrees clockwise from north.
9. Visibility (km): The visibility in kilometers, indicating how far one can see horizontally in the atmosphere.
10. Pressure (millibars): The atmospheric pressure in millibars.
11. Daily Summary: A summary of the weather conditions for the day.

**REGRESSION** :-

The Algorithms used for regression are :

1. Linear Regression
2. Random forests
3. Extra Random forests
4. Adaboost

**1)Linear Regression:**

Linear regression is a statistical method used to understand the relationship between variables. It assumes a linear relationship between the dependent and independent variables. The model estimates a line (in simple linear regression) or a plane/hyperplane (in multiple linear regression) that best fits the data. It's widely used for prediction and understanding the direction and strength of relationships between variables in fields like economics, finance, and science.

**2)Random forests:**

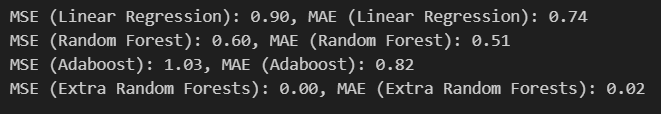
Regression using random forests is a machine learning method that predicts continuous values by using an ensemble of decision trees. Each tree is trained on a random subset of the data and features. The final prediction is the average of predictions made by all trees. It's robust against overfitting, handles large datasets well, and provides feature importance scores. It's implemented in libraries like scikit-learn in Python and is useful for various regression tasks.

**3)Extra Random forests:**

Regression using Extra Random Forests is a robust ensemble learning technique for predicting continuous outcomes. It introduces extra randomness in feature selection and splitting, reducing overfitting and increasing generalization. With scalability and insensitivity to feature scaling, it's ideal for various fields, including finance and healthcare, where accurate predictions are essential, and noise and outliers are common.

**4)AdaBoost:**

AdaBoost, when used for regression, combines multiple weak learners into a strong model. Initially, each data point has equal weight, and a weak learner (often a decision tree) is trained on the data. Subsequent learners focus more on the previously mispredicted data points. The final prediction is a weighted sum of individual learner predictions, with higher weights given to more accurate learners.



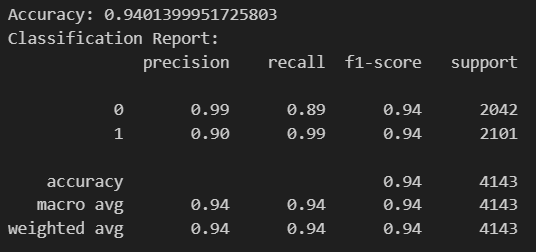
**CLASSIFICATION :-**

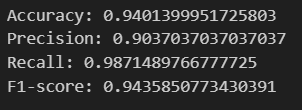
The Algorithms used for classification are :

1. Naive Bayes
2. KNN
3. SVM
4. Decision Trees
5. **Naive Bayes :**

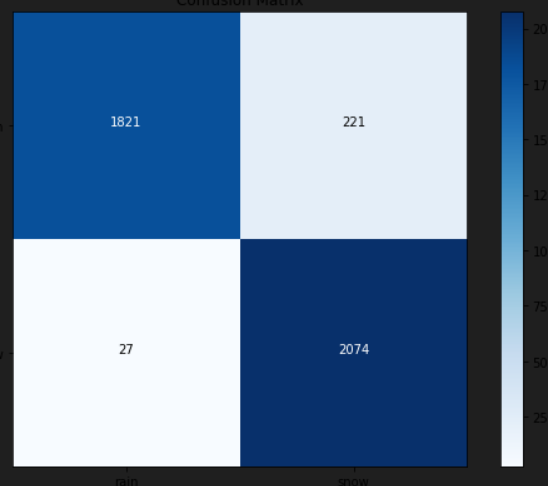
Naive Bayes is a popular and simple classification algorithm based on Bayes' Theorem, with an assumption of independence between features. Despite its simplicity, Naive Bayes often performs surprisingly well in many real-world situations, particularly in text classification and spam filtering.

After applying Naive Bayes on the Dataset , we get the following outcome :





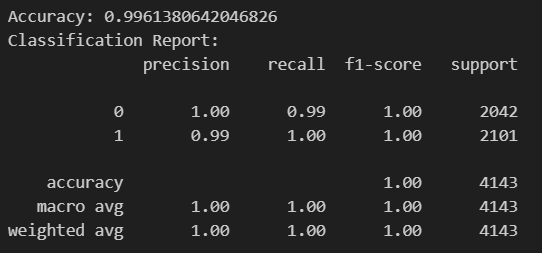
We get the following confusion matrix :

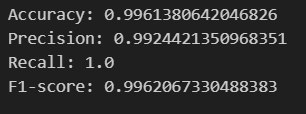


1. **K-Nearest Neighbor :**

In KNN classification, the class of an unseen data point is predicted based on the class of its nearest neighbors. The "K" in KNN represents the number of nearest neighbors to consider.KNN works based on the assumption that similar data points are close to each other in the feature space.

After applying the KNN algorithm on the dataset , we gained the following:





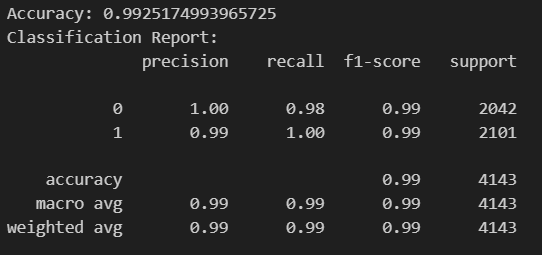
We get the following confusion matrix :

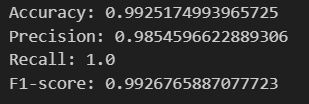


1. **Support Vector Machines :**

SVMs work by finding the optimal hyperplane that best separates data points belonging to different classes. The "support vectors" are the data points closest to the decision boundary. SVM aims to maximize the margin between these support vectors and the decision boundary.

After applying the SVM algorithm on the dataset we get the following results:





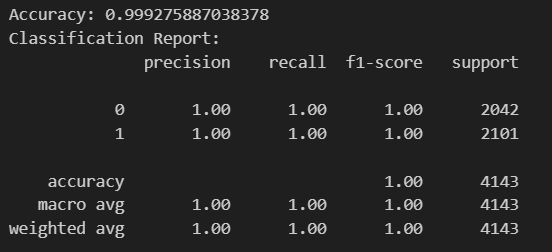
We get the following confusion matrix:

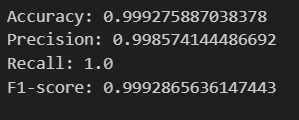


1. **Decision Trees :**

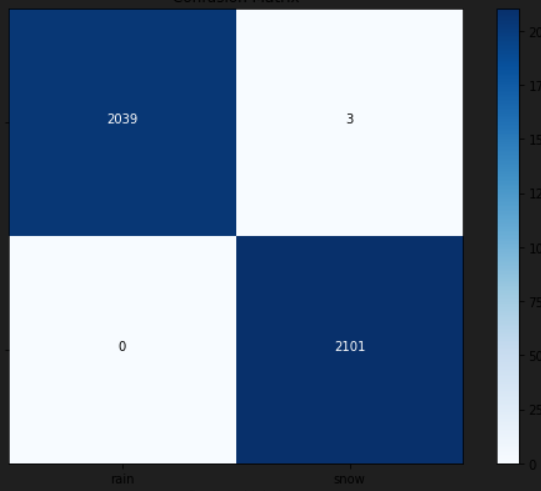
Decision Trees make decisions by splitting the data into smaller subsets based on the features that best separate the classes or minimize the variance in the target variable. These splits create a tree-like structure where each internal node represents a decision based on a feature, each branch represents an outcome of that decision, and each leaf node represents the final prediction.

After applying Decision trees on the dataset we get the following outcome:





We get the following confusion matrix :



1. **Naive Bayes VS K-Nearest Neighbor VS Support Vector Machines VS Decision Trees :**

|  | **Naive Bayes** | **KNN** | **SVM** | **Decision Trees** |
| --- | --- | --- | --- | --- |
| **Accuracy** | **0.9401** | **0.99613** | **0.9925** | **0.99927** |
| **Precision** | **0.9037** | **0.99244** | **0.9854** | **0.99857** |
| **Recall** | **0.9871** | **1** | **1** | **1** |
| **F1 Score** | **0.9435** | **0.99620** | **0.9926** | **0.99928** |

**CONCLUSION :**

Decision Trees performed the best overall, with the highest scores in accuracy, precision, recall, and F1 score. KNN and SVM also performed well, especially in terms of accuracy and F1 score. Naive Bayes had the lowest scores in all metrics, indicating its relatively poorer performance compared to the other algorithms. However, it still achieved respectable scores, especially in recall. In summary, for this particular classification task, Decision Trees are the preferred choice due to their superior performance across multiple metrics.

**CLUSTERING :**

The Algorithms used for Clustering are :

1. K-Means
2. EM-clustering
3. K-medoids

**ELBOW METHOD**: The elbow method is a heuristic used to determine the optimal number of clusters (K) in a dataset for clustering algorithms such as K-means.

Here's how it works:

1. Choose a range of K values: Start by selecting a range of K values (e.g., from 1 to 10).
2. Compute SSE for each K: For each value of K, run the clustering algorithm and compute the SSE.
3. Plot SSE against K: Plot the SSE values against the corresponding K values.
4. Identify the "elbow" point: Look for the point where the rate of decrease in SSE slows down (forming an elbow-like curve). This point represents the optimal K value.

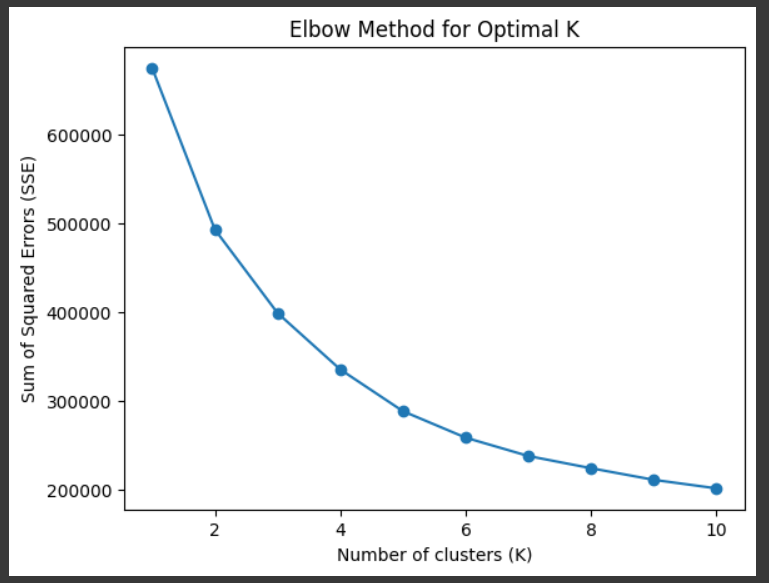
The idea behind the elbow method is that as the number of clusters increases, the SSE tends to decrease. However, beyond a certain point, adding more clusters does not significantly reduce the SSE, resulting in a less pronounced decrease in SSE and the formation of an "elbow" in the SSE plot.

1. **K-Means:**

K-means clustering is one of the most popular and simple unsupervised machine learning algorithms used for clustering data. It aims to partition data into K clusters where each data point belongs to the cluster with the nearest mean (centroid).

After applying K-means clustering on the dataset :

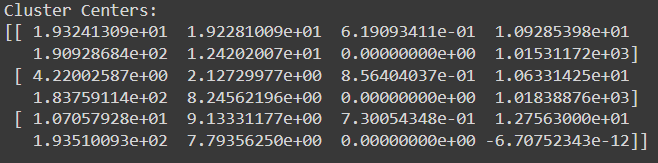
We get the following SSE graph for different K :

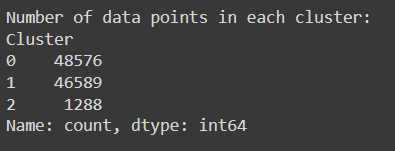


By analyzing the graph , we get the optimal value for k that is equal to 3.

After applying the clustering algorithm for k=3 ,

We get the following cluster centers and data points for each cluster:



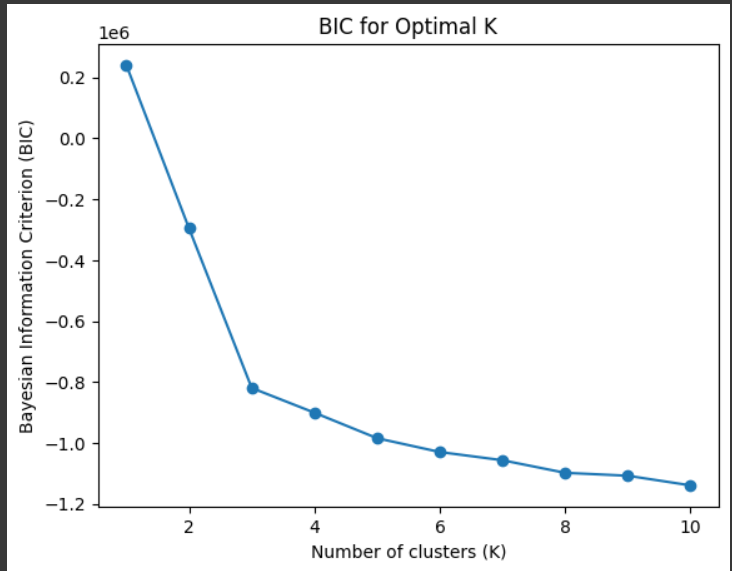


1. **EM-Clustering :**

EM clustering using the Gaussian model is an algorithm that identifies clusters in data assuming a Gaussian (normal) distribution. It iteratively estimates parameters such as means, covariances, and mixing coefficients to maximize the likelihood of the observed data. EM alternates between an Expectation step, where it computes the probability of each data point belonging to each cluster, and a Maximization step, where it updates the parameters based on these probabilities. This algorithm is useful for applications like image segmentation, anomaly detection, and customer segmentation.

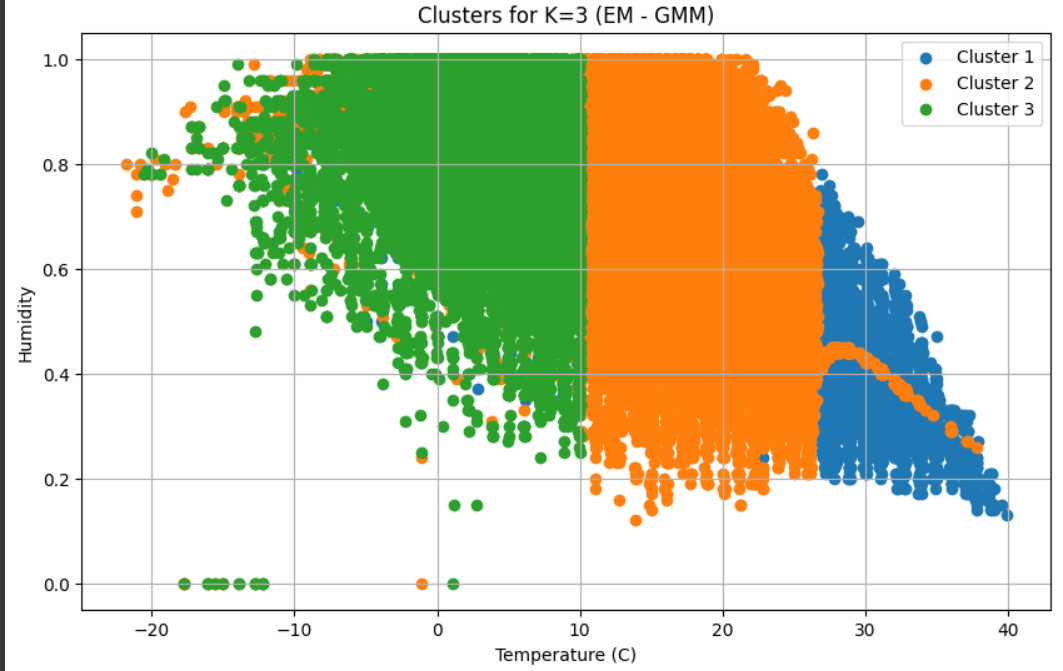
After applying EM-clustering ,we get the following SSE graph:

Bayesian Information Criterion (BIC) is used as a measure of SSE for GMM



We can see the elbow occurs at k=3 , therefore k=3 is the optimal value .

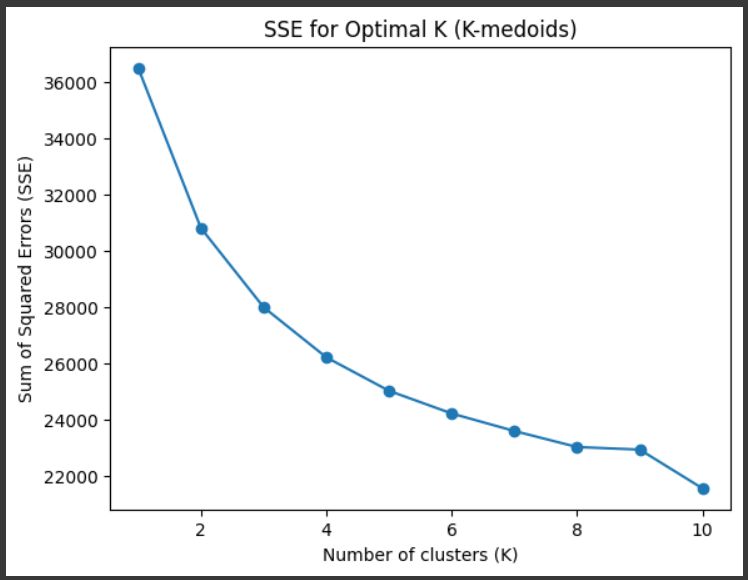
Plotting clusters for k=3 , we get :



1. **K-medoids :**

K-medoids is a clustering algorithm that uses actual data points (medoids) to represent clusters and partitions data into K clusters by minimizing the dissimilarity between data points and their corresponding medoids. It is useful in scenarios where using centroids as representatives may not be meaningful, and the choice of medoids provides more interpretable results.

After applying the K-medoids , we get the following graph :



Choosing k=3 and plotting the clusters we get :

