Report :

# Introduction

The report presents an overview of six basic algorithms: Linear Regression, Logistic Regression, Polynomial Regression, K-Means Clustering, K-Nearest Neighbors, Neural Networks, and Decision Trees. The working mechanism, when to use it, and its tunable hyperparameters of each algorithm are specified.

**1. Linear Regression**

**Linear Regression** is a supervised learning algorithm used to predict a continuous target variable based on one or more input features.

**Key Features:**

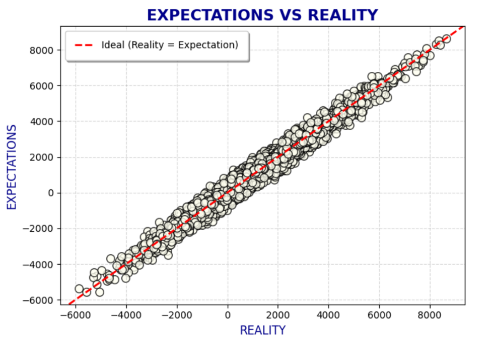
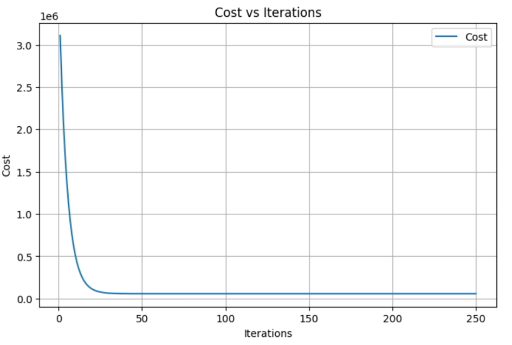
* Models the relationship between independent variables and the target variable by fitting a straight line.
* Minimizes the mean squared error (MSE) to find the best-fit line.

**Hyperparameters:**

* Learning Rate (alpha) = 0.1
* Iterations = 250

**Metrices:**

* Mean Squared Error for Training Data: 116055.78
* Mean Squared Error for Development Data: 116121.87

**Figures: **

**2. Logistic Regression**

**Logistic Regression is used for binary classification tasks, predicting probabilities of class membership.**

**Key Features:**

* **Uses the sigmoid function to map predicted values to probabilities.**
* **Thresholding is applied (e.g., 0.5) to classify.**

**Hyperparameters:**

* Learning Rate(alpha): 0.5
* Iterations: 2000

**Metrices:**

* Train Accuracy: 0.93484375
* Development Accuracy: 0.9357291666666666
* Precision for Development Set: 0.9298561151079137
* Precision for Training Set: 0.9295192899200281
* Recall for Development Set: 0.8596607914865314
* Recall for Training Set: 0.8615296896635986
* f1 for Training Set: 0.8942340209671964
* f1 for Development Set: 0.8933817176429929

**3. Polynomial Regression**

**Polynomial Regression is an extension of linear regression that models the relationship between the target variable and features as an nth-degree polynomial.**

**Key Features:**

* **Captures non-linear relationships.**
* **Increases the feature space by adding polynomial terms.**

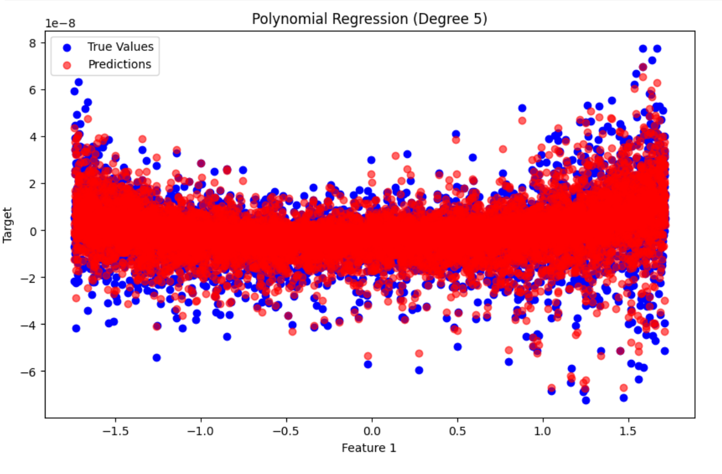
**Hyperparameters:**

* **Learning Rate(alpha) : 0.01**
* **Iterations: 1000**
* **L2 regularization strength: 1**
* **Degree: 5**
* **Gradient\_clip\_value: 10**

**Metrices:**

* **R2 Score on Train Data: 0.9457485892957942**
* **R2 Score on Dev Data: 0.9434478220226037**

**Figures:**

****

**4. K-Means Clustering**

**K-Means is an unsupervised learning algorithm used for clustering data points into clusters.**

**Key Features:**

* **Minimizes the variance within clusters.**
* **Iteratively updates cluster centroids and assigns points.**

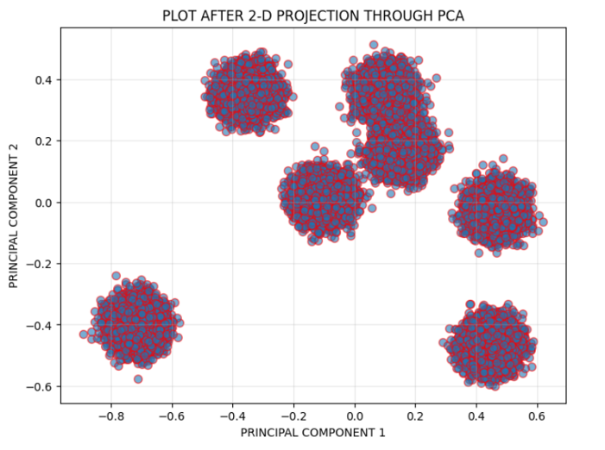
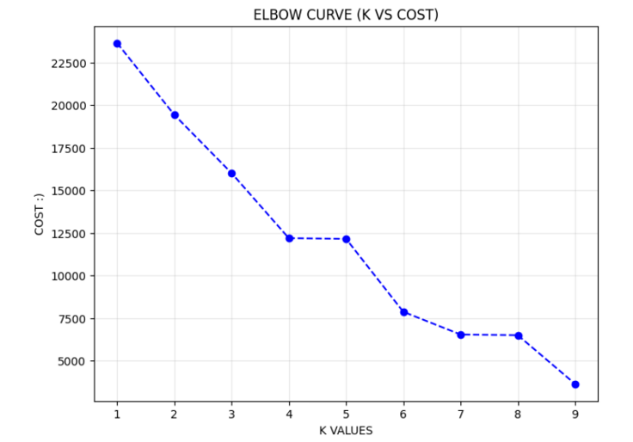
**Algorithm:**

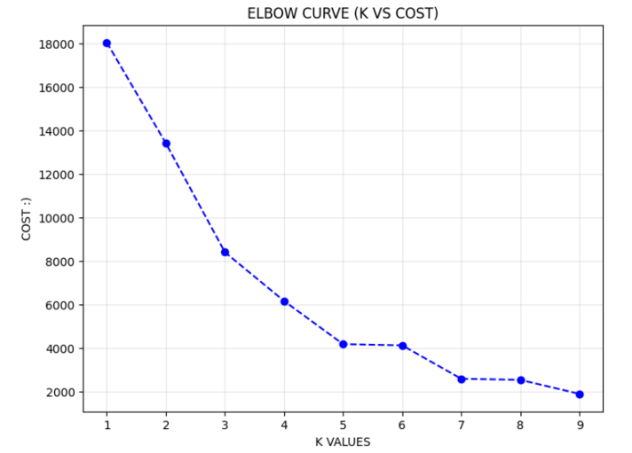
1. **Initialize cluster centroids.**
2. **Assign each point to the nearest centroid.**
3. **Update centroids by calculating the mean of assigned points.**
4. **Repeat until convergence.**

**Hyperparameters:**

* **K = 4**
* **Iterations = 100**

**Figures:**

****

****

**5. K-Nearest Neighbors (KNN)**

**KNN** is a simple, instance-based supervised learning algorithm used for both classification and regression.

**Key Features:**

* Finds the nearest data points to a query point and uses their majority class (classification) or average (regression).

**Hyperparameters:**

* K = 6

**Metrices:**

* TRAINING F1 SCORE IS: 0.9521009870701075
* DEVELOPMENT F1 SCORE IS: 0.9435975574672717

**6. Neural Networks**

**Neural Networks** are a class of algorithms inspired by the human brain, capable of modeling complex relationships in data.

**Key Features:**

* Consists of layers of interconnected neurons with activation functions.
* Can handle non-linear and high-dimensional data.

**Components:**

* **Input Layer**: Takes input features.
* **Hidden Layers**: Perform transformations using weights, biases, and activation functions.
* **Output Layer**: Produces the final prediction.

**Hyperparameters:**

* **For Binary Class:**

1. layer\_sizes **=** [X**.**shape[1], 128, 64, 1]
2. learning\_rate **=** 0.0001
3. epochs **=** 50
4. batch\_size **=** 64
5. beta1 **=** 0.9
6. beta2 **=** 0.999
7. epsilon **=** 1e-8
8. lambda\_reg **=** 1

* **For Multi Class:**

1. layer\_sizes **=** [X**.**shape[1], 128, 64, num\_classes]
2. learning\_rate **=** 0.0001
3. epochs **=** 50
4. batch\_size **=** 64
5. beta1 **=** 0.9
6. beta2 **=** 0.999
7. epsilon **=** 1e-8
8. lambda\_reg **=** 1

**Metrices**:

For Binary:

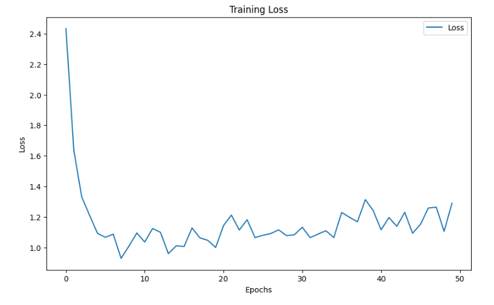
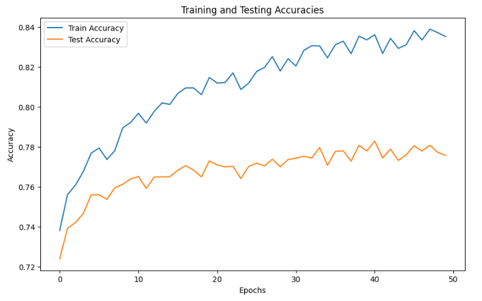
* f1 for Training Set: 0.8387983986798278
* f1 for Development Set: 0.7815182321787301

For Multi:

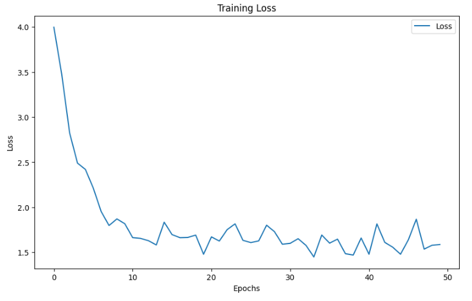
* The Training F1 SCORE IS 0.5627528622968111
* The Development F1 SCORE IS 0.5268375701243153

**Figures:**

For Binary:



For Multi:



**7. Decision Trees**

**Decision Trees** are a supervised learning algorithm used for classification and regression tasks.

**Key Features:**

* Splits data into subsets based on feature values, creating a tree-like structure.
* Uses impurity measures like Gini Index or Entropy.

**Hyperparameters:**

* **Max Depth**: 5
* **Min Samples Split**: 10
* **Min Samples Leaf**: 5

**Metrices**:

f1 for Training Set: 0.8858604531189544

f1 for Development Set: 0.8842179015589421