Project Documentation

VEHICLE SALES DATA

1. Project Overview

This project explores machine learning techniques for analyzing and predicting outcomes using a dataset, car_prices.csv.

It includes:

Data preprocessing and exploratory data analysis.

Implementation of machine learning models such as Linear Regression, Logistic Regression, KMeans, PCA, Gaussian Naive Bayes, and SVM.

Evaluation of models using metrics and visualization tools.

2. Dependencies

The project uses the following Python libraries:

pandas: Data manipulation and analysis.

numpy: Numerical computations.

scikit-learn: Machine learning models and preprocessing.

Other utilities: Metrics for model evaluation and StandardScaler for normalization.

Ensure all libraries are installed using:

pip install pandas numpy scikit-learn

3. Dataset

The dataset, car_prices.csv, contains features and labels for predicting car prices. It is loaded into a DataFrame for analysis:

Load the dataset

data = pd.read_csv('/content/drive/MyDrive/Project/car_prices.csv')

4. Workflow

Data Loading and Preprocessing

Read and inspect the dataset.

Handle missing values, outliers, and normalize the data using StandardScaler.

Exploratory Data Analysis (EDA)

Visualize data distributions.

Identify correlations and perform dimensionality reduction using PCA.

Model Training

Train multiple machine learning models:

Regression: Linear and Logistic Regression.

Classification: Gaussian Naive Bayes, Support Vector Classifier (SVC).

Clustering: KMeans.

Split the dataset using train_test_split.

Evaluation

Compute metrics such as accuracy and loss.

Evaluate using Euclidean distances or other statistical measures.

5. Example Code

Linear Regression Example:

```
# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Train the model
lr = LinearRegression()
lr.fit(X_train, y_train)

# Evaluate
predictions = lr.predict(X_test)
print("Linear Regression Predictions:", predictions)
```

KMeans Clustering Example:

```
# Fit KMeans model
kmeans = KMeans(n_clusters=3, random_state=42)
kmeans.fit(data)

# Cluster centroids
print("Cluster Centers:", kmeans.cluster_centers_)
```

7. Conclusions

This project demonstrates:

The use of various machine learning models for both regression and classification tasks.

Data dimensionality reduction techniques for better visualization and analysis.

Practical application of clustering for grouping data without labels.