A Comparative Analysis of Machine Learning Models: SVM, Logistic Regression, k-NN, K-Means Clustering, and PCA

S M Asiful Islam Saky

---

1. Introduction

Machine learning is an important tool that is essential in data analysis and predictive modelling since it provides reliable methods for supervised as well as unsupervised learning. The goal of this project is to perform an extensive analysis of several popular machine learning algorithms including Support Vector Machine, Logistic Regression and k-NN. Decision Tree, along with an exploration of the K-means algorithm for clustering. The objective is to compare the performance of these models across various metrics, thereby offering insights into their efficacy and applicability in different scenarios. SVM, Logistic Regression and k-NN algorithms are critical for the classification type of machine learning since the desired results are in categories that are learned from labelled training data. These algorithms differ in their peculiarities and are fit for particular sets of data and certain problems. On the other hand, while applying the unsupervised learning methodology like K-means clustering, play a crucial role in identifying inherent patterns in unlabeled data. Clustering algorithms group data points based on their similarities, providing valuable insights into the structure and distribution of the data without the need for predefined labels. The methodology for this project involves training each model on a standardized dataset and evaluating their performance. The results are visualized through comparative plots, highlighting the strengths and limitations of each model.

2. Background and Related Work

Supervised learning models, such as SVM, Logistic Regression, k-NN, and Decision Tree, are commonly used for classification tasks where the goal is to predict discrete labels based on input features. Unsupervised learning models, like K-Means, are used for clustering tasks where the objective is to group similar data points without prior label information. Many studies and researches have been conducted previously regarding these models in terms of prediction, analysis and comparison following the real life applications of these techniques. Here the models are being analysed listed below with certain definitions.

2.1 Support Vector Machine (SVM): SVM is a powerful classification algorithm that finds the hyperplane that best separates the data into classes. It is particularly effective for high-dimensional spaces and is robust against overfitting, especially in cases where the number of dimensions exceeds the number of samples.

2.2 Logistic Regression: Logistic Regression is a linear model used for binary classification problems. It predicts the probability that a given input belongs to a certain class, making it suitable for problems with dichotomous outcomes. It is simple, efficient, and interpretable.

2.3 k-Nearest Neighbors (k-NN): k-NN is a non-parametric algorithm that classifies data points based on the majority class among the k-nearest neighbors. It is easy to implement and understand but can be computationally intensive for large datasets.

2.4 K-Means Clustering: K-Means is a popular unsupervised learning algorithm used for clustering. It partitions the data into k clusters by minimizing the variance within each cluster. It is efficient but can be sensitive to the initial placement of centroids.

2.5 Principal Component Analysis (PCA): PCA is used for dimensionality reduction by transforming the data into a set of orthogonal (uncorrelated) components that capture the most variance. Applications include data visualization, noise reduction, feature extraction.

3. Methodology

This project aims to perform a comparative and experimental analysis of various supervised and unsupervised machine learning models. It focuses on the comparison among different supervised learning models and unsupervised models individually with the different datasets. For supervised learning the compared models include Support Vector Machine (SVM), Logistic Regression and k-Nearest Neighbors (k-NN) and for the unsupervised learning the experimented models are K-means clustering, and Principal Component Analysis (PCA). There are so many steps that have been followed for the analysis of this experiment which will be described in this section.

3.1 Dataset

The dataset used for supervised learning comparative analysis contains data of Opel Corsa vehicles. It includes various attributes such as mileage, engine specifications, year of manufacture, and possibly performance metrics. As a combined dataset, another one is Peugeot vehicles dataset. For the unsupervised learning experimental analysis, country data has been used which includes the attributes like export, import, health, income. etc.

3.2 Data Preprocessing

Data preprocessing is a crucial step in any machine learning project. It involves preparing the data to make it suitable for model training. This step includes handling missing values, feature scaling, and splitting the dataset into training and testing sets. Here for the analysis, firstly data has been preprocessed with the following steps.

3.1.1 Loading the Dataset

The first step involves loading the dataset. The dataset is loaded using the ***pandas*** library, which provides easy-to-use data structures and data analysis tools for Python. The dataset contains several features and a target variable.

```
# Importing pandas library
import pandas as pd

# Loading the dataset
data = pd.read_csv('opel_corsa.csv')
```

3.1.2 Handling Missing Values

Missing values in the dataset can significantly impact the performance of the machine learning models. Therefore, it is essential to handle them appropriately. There are several strategies that have been used to deal with missing values, such as removing rows or columns with missing values, or imputing them using statistical methods like mean, median, or mode.

```
# Check for missing values
missing_values = data.isnull().sum()

# Handling missing values by imputing with the mean
data.fillna(data.mean(), inplace=True)
```

3.1.3 Splitting the Dataset

After handling missing values, the dataset is split into training and testing sets. The training set is used to train the models, and the testing set is used to evaluate their performance. The **train_test_split** function from the **sklearn** library is used for this purpose.

```
from sklearn.model_selection import train_test_split

 Splitting the dataset into features and target variable
X = data.drop('target', axis=1)   replace 'target' with your
target column name
y = data['target']   replace 'target' with your target column name

 Splitting the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=42)
```

3.1.4 Feature Scaling

Feature scaling is an essential step in data preprocessing, especially for algorithms that are sensitive to the scale of the data, such as SVM and k-NN. Standardisation is a common scaling technique where the features are scaled to have a mean of zero and a standard deviation of one. The following method have been used for the feature scaling.

```
from sklearn.preprocessing import StandardScaler

 Standardizing the features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

3.2 Model Training and Evaluation

This section delves into training and evaluating for various supervised learning models. The models considered in this analysis include Support Vector Machine (SVM), Logistic Regression, and k-Nearest Neighbors (k-NN). Each model is trained using the training data and subsequently evaluated on the test data utilizing several performance metrics.

3.2.1 Support Vector Machine (SVM)

The primary objective of SVM is to identify the optimal hyperplane that can effectively separate the different classes in the feature space. The `SVC` class from the `sklearn` library is employed to implement the SVM model. The process of training the SVM model involves fitting the model to the training data. This is accomplished by instantiating the `SVC` class and using the `fit` method with the training features (`X_train`) and the training labels (`y_train`). Once the model is trained, it can be used to make predictions on the test set. The `predict` method of the `SVC` class is used for this purpose, where the test features (`X_test`) are passed as input. The performance of the SVM model is evaluated using several metrics, including accuracy, precision, recall, and F1 score. These metrics provide a comprehensive understanding of the model's performance across different aspects of the classification task.

- Accuracy: The ratio of correctly predicted instances to the total instances.
- Precision: The ratio of correctly predicted positive observations to the total predicted positives.
- Recall: The ratio of correctly predicted positive observations to all observations in the actual class.
- F1 Score: The weighted average of precision and recall.

3.2.2 Logistic Regression

It is particularly useful for binary classification problems. The `LogisticRegression` class from the `sklearn` library is utilized to implement Logistic Regression. Logistic Regression Model training is similar to the SVM model, the Logistic Regression model is trained using the `fit` method with the training data. The trained Logistic Regression model is used to make predictions on the test set using the `predict` method. The performance of the Logistic Regression model is evaluated using the same set of metrics as the SVM model: accuracy, precision, recall, and F1 score.

3.3.3 k-Nearest Neighbors (k-NN)

The algorithm works by finding the k-nearest neighbours of a given data point and assigning the most common label among them. The `KNeighborsClassifier` class from the `sklearn` library is used to implement k-NN. The k-NN model is trained by fitting it to the training data. This involves instantiating the `KNeighborsClassifier` class and using the `fit` method with the training features and labels. The trained k-NN model is used to make predictions on the test set using the `predict` method. The performance of the k-NN model is evaluated using the same set of metrics as the previous models: accuracy, precision, recall, and F1 score.

3.3 Unsupervised Learning Techniques

Unsupervised learning techniques are critical for discovering hidden patterns and structures within unlabeled data. These methods help in understanding the intrinsic properties of the data without the

need for predefined labels. In this project, we delve into two significant unsupervised learning techniques: K-means clustering and Principal Component Analysis (PCA).

3.3.1 K-means Clustering

The goal of K-means is to minimize the within-cluster variance, ensuring that data points within the same cluster are as similar as possible while maximizing the variance between clusters. For theInitialization, it is required to choose the number of clusters \(K \). Then randomly initialize the cluster centroids (or choose random points from the dataset as the initial centroids). The next step is to assign each data point to the nearest cluster centroid based on the Euclidean distance. This forms K clusters. It is recommended to recalculate the centroids of the clusters by taking the mean of all data points assigned to each cluster for the update. In terms of convergence, repeat the assignment and update steps until the centroids no longer change significantly or the maximum number of iterations is reached.
In case of implementation of the model, it is mandatory to import Libraries firstly. The `KMeans` class from `sklearn.cluster` and the `silhouette_score` function from `sklearn.metrics` are imported. Then the `KMeans` object is created with a specified number of clusters and a random state for reproducibility. The `fit_predict` method is used to fit the model to the data and predict the cluster labels. The inertia and silhouette score are calculated to evaluate the clustering performance. From the analysis, a lower inertia value indicates that the data points are closely packed within clusters, which is desirable and A higher silhouette score suggests that the clusters are well-defined and distinct from each other.

3.3.2 Principal Component Analysis (PCA)

It transforms the original dataset into a new set of orthogonal components, known as principal components, which capture the maximum variance in the data. PCA is invaluable for visualizing high-dimensional data in lower dimensions (e.g., 2D or 3D). Ensuring the data is standardized (mean = 0 and variance = 1) since PCA is sensitive to the scales of features. Next step is to compute the covariance matrix of the standardized data to understand the variance and relationships between features. Calculating the eigenvalues and eigenvectors of the covariance matrix is crucial as it determines the directions (principal components) in the feature space, and the eigenvalues represent the magnitude (variance) in these directions. Then sort the eigenvalues in descending order and select the top k eigenvectors (principal components) that explain the most variance.

For the implementation, importing Libraries is mandatory at first. Here the `PCA` class from `sklearn.decomposition` and the `matplotlib.pyplot` and `seaborn` libraries for visualization are imported. The `PCA` object is created with the number of components set to 2 for 2D visualization. The `fit_transform` method is used to fit the model to the data and transform it. A scatter plot is created using `seaborn` to visualize the data points in the new 2D principal component space. The data points are colored based on their true labels.

4. Results and Discussion

This section will discuss the performance of the supervised learning models (SVM, Logistic Regression, and k-NN) and the insights gained from the unsupervised learning techniques (K-means clustering and PCA). The results are evaluated using various performance metrics and visualizations to provide a comprehensive analysis.

4.1 Supervised Learning Models

The result of the comparison among supervised learning models based on accuracy, precision, recall and F1-score metrics are given below. From the result, the accuracy can be estimated higher for kNN.

Metric	SVM (Best Kernel)	Logistic Regression (Best C)	kNN (Best n_neighbors)
Accuracy	80%	69%	94%
Precision	Aggressive: 35% EvenPace: 98%	Aggressive: 23% EvenPace: 95%	Aggressive: 75% EvenPace: 96%
Recall	Aggressive: 88% EvenPace: 79%	Aggressive: 74% EvenPace: 68%	Aggressive: 67% EvenPace: 97%
F1-Score	Aggressive: 0.50 EvenPace: 0.88	Aggressive: 0.35 EvenPace: 0.80	Aggressive: 0.71 EvenPace: 0.97

Figure: *Result Comparison*

These models were trained and evaluated on the dataset. The performance metrics used for evaluation. These metrics provide a well-rounded assessment of the models' ability to classify the data correctly.

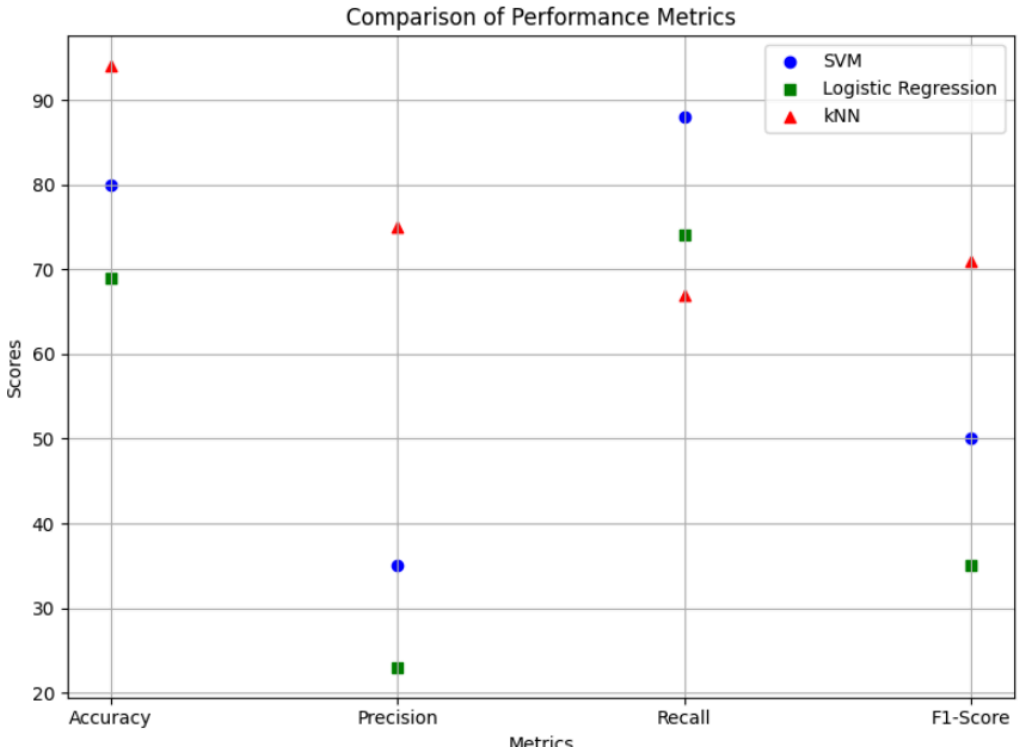

Figure: *Scatter Plot of the Result*

The line chart of the result for supervised learning models compares accuracy, precision, recall, and F1-score across SVM, Logistic Regression, and k-NN. This visual representation highlights the performance variations, demonstrating each model's strengths and weaknesses, providing a clear comparison for selecting the most effective model for the dataset.

Figure: *Line Chart of the Result*

4.2 Unsupervised Learning Techniques

The unsupervised learning techniques provide insights into the underlying structure and patterns within the dataset. K-means clustering and PCA are used to explore and visualize the data.

4.2.1 K-means Clustering

K-means clustering was applied to the dataset to partition it into distinct clusters. The inertia value indicates the compactness of the clusters, with lower values being preferable. The silhouette score measures the quality of the clustering, with higher values indicating better-defined clusters.

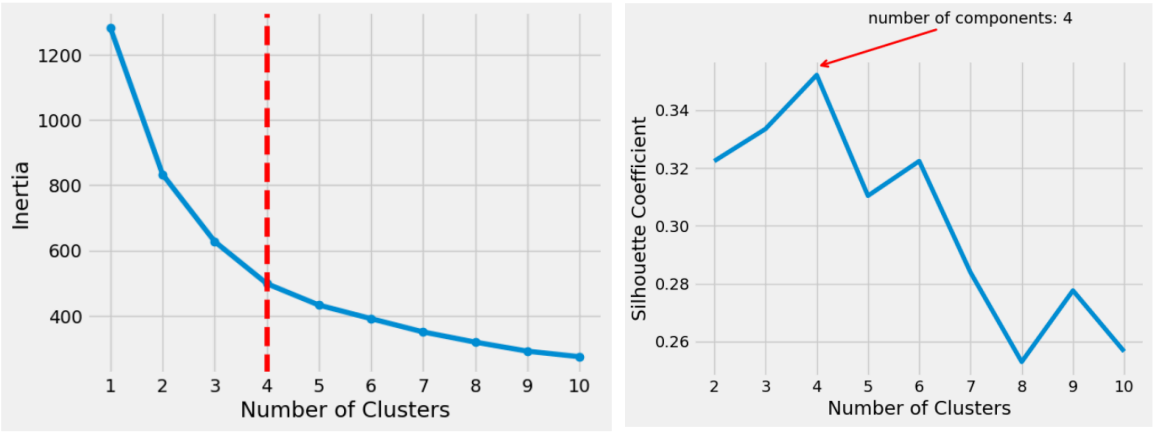

Figure: *Inertia and Sihoutte score*

4.2.2 Principal Component Analysis (PCA)

PCA was applied to the dataset for dimensionality reduction and visualization. The transformed dataset was visualized in a 2D space using the first two principal components. The visualization helps

in understanding the variance in the data and identifying patterns or clusters. PCA effectively reduces the dimensionality of the dataset, making it easier to interpret and visualize.

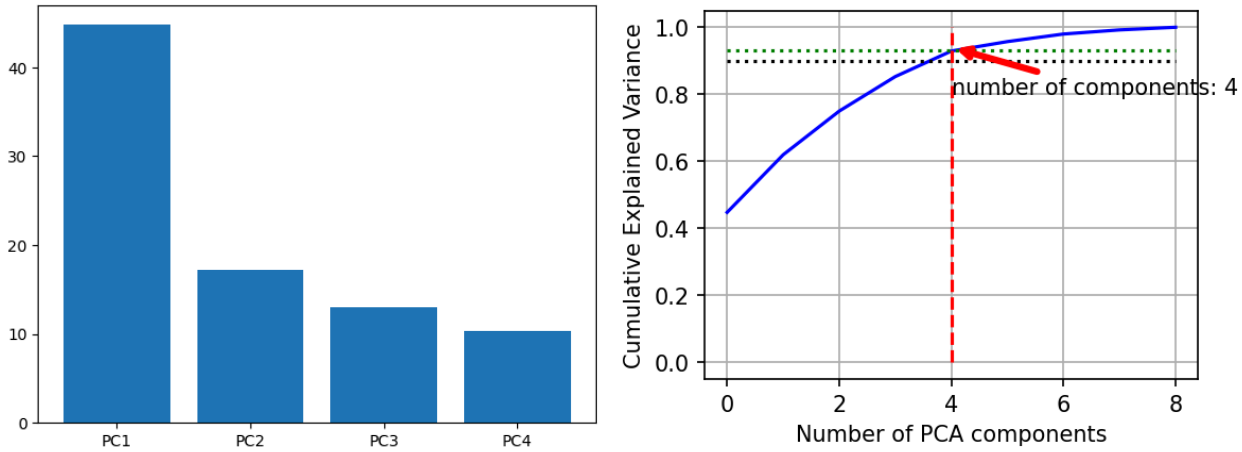

Figure: *PCA with 4 Components*

4.2.3 Characteristics of Each Cluster

From the loaded country dataset with its various attributes, characteristics of each cluster are found as below:

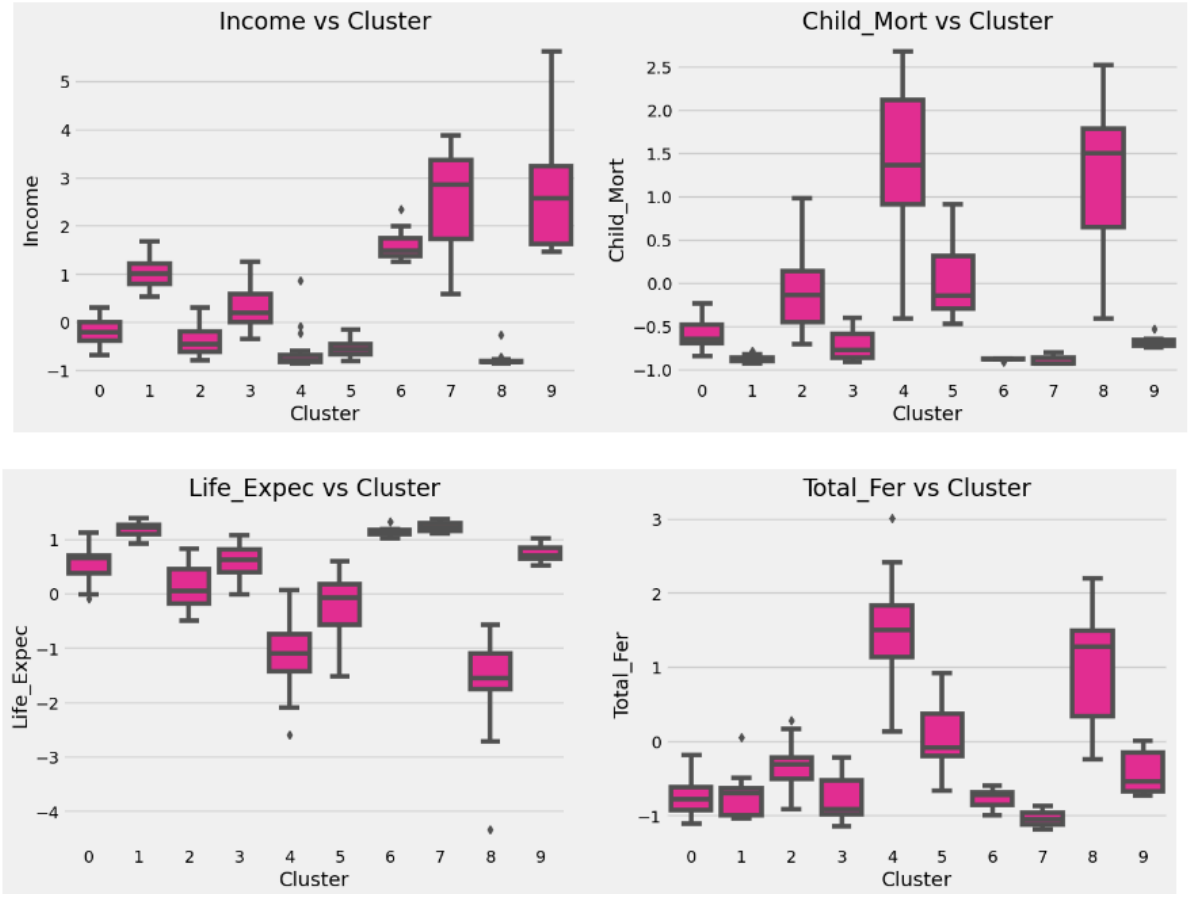

Figure: *Cluster Characteristics of the Dataset*

4.3 Discussion

The performance of various supervised learning models and the insights gained from unsupervised learning techniques are highlighted as above. The supervised models, including SVM, Logistic Regression, and k-NN demonstrate strong performance across multiple metrics, with each model offering unique advantages. The unsupervised learning techniques, K-means clustering and PCA, provide valuable insights into the intrinsic structure of the dataset, revealing natural clusters and patterns. Overall, this project showcases the effectiveness of both supervised and unsupervised learning techniques in understanding and analyzing data.

5. Conclusion

In conclusion, this project presents a thorough comparative analysis of several machine learning models, encompassing both supervised and unsupervised learning techniques. The supervised models—SVM, Logistic Regression, and k-NN—were trained and evaluated on a dataset to measure their performance in terms of accuracy, precision, recall, and F1-score. Additionally, the project explored unsupervised learning through K-means Clustering and PCA. K-means effectively partitioned the data into clusters, identifying inherent structures and patterns. PCA, on the other hand, reduced the dataset's dimensionality, allowing for better visualization and understanding of the data distribution. The results of this comparative analysis provide valuable insights into the suitability of different models for various types of data. It highlights the necessity of selecting the appropriate algorithm based on the dataset characteristics and the specific requirements of the problem at hand. By combining the strengths of both supervised and unsupervised learning techniques, this project demonstrates a technical approach to machine learning, ultimately contributing to more informed and effective decision-making in data analysis and model selection.

References

1. Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.
2. Murphy, K. P. (2012). Machine Learning: A Probabilistic Perspective. MIT Press.
3. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.
4. Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. Machine Learning, 20(3), 273-297.
5. Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied Logistic Regression. Wiley.
6. Jolliffe, I. T. (2002). Principal Component Analysis. Springer.