Machine Learning Classification

Project Report

DATE:

December 20, 2024

Submitted By:

Frais Asghar

CMS ID: 516900

ME-16(Section-A)

SMME
SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERNG

Project Report

Abstract:

The current paper examines regression and classification modeling using an authentic data set retrieved from the **Kaggle Library**. A Linear Regression model is applied for the purpose of regression prediction. On the other hand, **DT** and **KNN** are used as classifiers. It carefully discusses how the chosen data set can be appropriately applied for the purpose of predictive modeling by identifying trends, boundaries in decision making, and the accuracy in classification.

Key performance metrics, such as **True Positive (TP)**, **True Negative (TN)**, **False Positive (FP)**, and **False Negative (FN)**, are analyzed in conjunction with precision, recall, and **F1-score** to determine the effectiveness of the models. A confusion matrix is used for a comprehensive analysis of classification outcomes, while a correlation matrix is used to explore relationships among independent variables in regression.

This report also adds theoretical explanations, mathematical derivations, and visual illustrations of decision boundaries for the **DT** and **KNN** classifier. It brings out into focus how crucial algorithmic choice and parameter tuning have become, which ensures perfect results in tasks of prediction modeling. The methodologies illustrated and evaluated demonstrate the hands-on utility of these methodologies in actual machine learning work.

Theory:

Regression Modeling: Linear Regression

Linear Regression is a basic supervised learning algorithm to predict continuous values. It assumes that there exists a linear relationship between the **dependent variable**, **output** and **independent variables**, **input variables**. The objective here is to fit a line so that the error of difference between the **predicted values** and **actual values** is at the minimum.

Mathematically represented as:

$$y = B_0 + B_1 x_1 + \ldots\ldots\ldots\ldots\ldots\ldots\ldots + B_n x_n + \epsilon$$

- y *is the* : Dependent variable (output)
- $x_1, x_2, \ldots x_n$: Independent variables (inputs/features)
- B_0 *is the* : Intercept (value of y in case all x is zero)
- $B_1, B_2, \ldots B_n$: Coefficients (degree of influence of each on the output)
- ϵ *is the* : Residual error term (difference of actual and predicted y)

Steps in Regression Analysis:

- **Data Splitting:** Split the dataset into **train** and **test** subsets (say 80:20 split).
- **Model Training:** Fit the **Linear Regression** model using training data.
- **Prediction:** Predict outcomes for **test data** using the **trained model**.
- **Evaluation:** Calculate **performance metrics** to gauge the prediction's correctness.

Performance Metrics for Linear Regression:

1. Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

2. Mean Squared Error (MSE):

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

3. Root Mean Squared Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

4. R-Squared (R^2):

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y}_i)^2}$$

Classification Modeling:

1. Decision Tree (DT):

A decision tree is a supervised learning algorithm using the tree structure for classifying data or for making predictions on continuous outcomes. The algorithm works iteratively splitting the dataset into sub-parts based on thresholds of the features while maximizing the purity at each node.

Key Concepts in Decision Trees:

I. Splitting Criterion:

The splitting of data is determined using metrics like **Gini Impurity**, **Entropy**, or **Mean Squared Error** in a **Decision Tree**.

- **Gini Impurity:**

$$\text{Gini} = 1 - \sum_{i=1}^{c} p_i^2$$

$$\text{Entropy} = -\sum_{i=1}^{c} p_i log_2(p_i)$$

II. Overfitting Prevention:

Pruning techniques are used to constrain the depth of the tree to avoid overfitting the training data.

II. Decision Boundaries:

In classification, DT results in axis-aligned boundaries that split feature space into separate class regions.

Advantages:

- Easy and interpretable.
- Both categorical and numerical data can be handled.

Disadvantages:

- Prone to overfitting.
- Performance degrades with noisy data.

2. K-Nearest Neighbors (KNN):

KNN is a non-parametric algorithm that classifies data points based on the majority class among their **k** nearest neighbors.

Mathematical Representation:

1. Euclidean Distance:

$$\mathbf{d\ (p,\ q)} = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$$

2. Decision Rule:

Assign the class with the highest count among the **k** nearest neighbors.

Key Features of KNN:

- Effective for small datasets.
- No explicit training phase; predictions are computed at runtime.

Challenges:

Requires careful tuning of **k**; a small **k** may lead to overfitting, while a large **k** may over-smooth the decision boundaries.

Evaluation Metrics and Confusion Matrix:

The confusion matrix is a summary of how the classification model has done:

CATEGORIES	PREDICTED POSITIVE	PREDICTED NEGATIVE
ACTUAL POSITIVE	True Positive (TN)	False Negative (FN)
ACTUAL NEGATIVE	False Positive (FP)	True Negative (TN)

Formulas for Important Metrics:

1. Accuracy:

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

2. Precision:

$$\textbf{Precision} = \frac{\textbf{TP}}{(\textbf{TP} + \textbf{FP})}$$

3. Recall (Sensitivity):

$$\textbf{Recall} = \frac{\textbf{TP}}{(\textbf{TP} + \textbf{FN})}$$

4. F1-Score:

$$\textbf{F1} = 2 \times \frac{\textbf{Precision} \times \textbf{Recall}}{\textbf{Precision} + \textbf{Recall}}$$

Correlation Matrix:

The correlation matrix gives information regarding relationships between features; multicollinearity can be inferred, and which are more important.

Correlation Coefficient Formula:

$$\textbf{Correlation (r)} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 (y_i - \bar{y})^2}}$$

Decision Boundaries:

Decision boundaries are graphical representations of how classifiers partition the feature space:

DT Boundaries: Axis-aligned and splits data into different classes.

KNN Boundaries: Non-linear and dependent on neighboring data points density.