 **The University of Azad Jammu and** 

**Kashmir**

**Department of Software Engineering**

**Project Report**

**Course Title:** **Machine Learning**

**LAB INSTRUCTOR:**

Engr. Awais Rathore

**Group Members:**

**2022-SE-10 Iqra Sohail**

**2022-SE-11 Kashuff Zaman**

**2022-SE-15 Shams-ul-huda**

**2022-SE-20 Samina Khursheed**

# Project Report: Linear Regression Model

**1. Introduction**

In today's data-driven world, predictive modeling plays a crucial role in decision-making across industries such as finance, healthcare, and marketing. This project focuses on developing a **Linear Regression Model** to predict target values based on given features. The dataset undergoes data cleaning, preprocessing, model training, and evaluation to ensure optimal performance. The primary objective is to apply **Linear Regression** to establish relationships between independent variables and a dependent variable, ensuring a high degree of accuracy in predictions.

**2. Dataset Overview**

The dataset comprises **training (train.csv)** and **test (test.csv)** files, which are loaded and explored using Python's pandas library. These datasets contain multiple independent variables (features) and a dependent variable (target).



**2.1 Data Exploration**

* The dataset is loaded using pandas.read\_csv() to inspect its structure.
* Key properties such as **column names, data types, missing values, and summary statistics** are analyzed.
* **Missing Values Analysis** is performed using isnull().sum() to determine the presence of null values.

**3. Data Preprocessing**

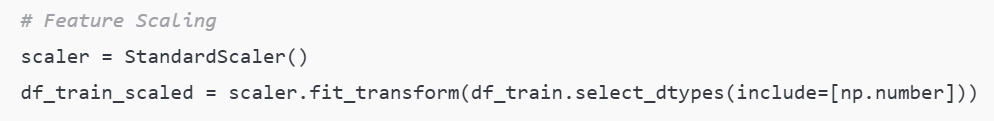
To enhance model accuracy, the dataset undergoes **preprocessing**, which includes handling missing values, feature scaling, and encoding categorical data.

**3.1 Handling Missing Values**

* **Missing value imputation** is performed to replace missing entries with appropriate statistical measures (mean/median/mode).
* A missing values heatmap is generated using seaborn to visualize data gaps.

**3.2 Feature Scaling**

* **StandardScaler and MinMaxScaler** are used to normalize numerical features, ensuring that all variables contribute equally to the model.



**3.3 Encoding Categorical Data**

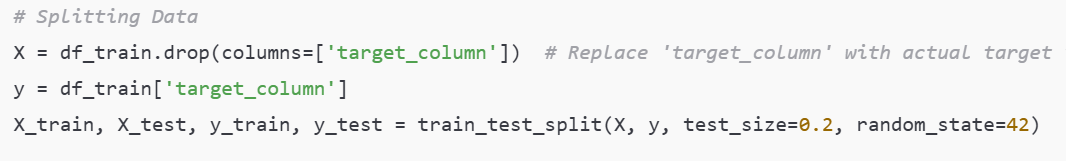
* **LabelEncoder and OneHotEncoder** convert categorical variables into numerical representations, allowing the regression model to interpret them effectively.

**4. Model Implementation**

The **Linear Regression Model** is implemented using **scikit-learn**, following these key steps:

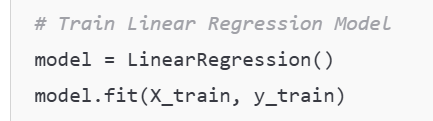
**4.1 Splitting Data into Training and Test Sets**

* The dataset is divided into training and testing sets using train\_test\_split() (e.g., 80% training, 20% testing).
* This ensures the model generalizes well to unseen data.



**4.2 Training the Model**

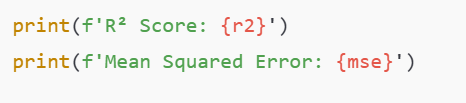
* The LinearRegression() class from sklearn.linear\_model is used to fit the model on the training data.
* The model learns coefficients (weights) for each independent variable to predict the target variable.



**4.3 Model Evaluation Metrics**

To assess model performance, the following evaluation metrics are calculated:

* **R² Score**: Measures the proportion of variance explained by the model (closer to 1 indicates a better fit).
* **Mean Squared Error (MSE)**: Computes the average squared differences between actual and predicted values (lower values indicate better performance).
* **Confusion Matrix & Classification Report** (if applicable): Used for classification tasks, but primarily, regression metrics are analyzed.



**5. Results & Discussion**

**5.1 Model Performance**

* The **R² Score** and **MSE** indicate how well the model fits the data.
* If the R² score is close to 1, the model has captured most of the variability in the target variable.
* A lower MSE suggests accurate predictions with minimal errors.

**5.2 Key Observations**

* Feature selection significantly affects model accuracy.
* Outliers and missing values, if not handled properly, can lead to poor performance.
* Normalizing data improves model stability and convergence.

**5.3 Limitations of the Model**

* **Assumption of Linearity**: Linear Regression assumes a straight-line relationship between dependent and independent variables, which may not always be the case.
* **Impact of Multicollinearity**: Highly correlated features can distort the model’s interpretation.
* **Handling Non-Numeric Data**: Categorical variables must be encoded properly, or they may introduce biases.

**6. Conclusion**

The **Linear Regression Model** effectively predicts the target variable by learning from historical data. The results demonstrate that preprocessing (handling missing values, scaling, and encoding) significantly impacts model performance. The project successfully applies **machine learning techniques** to real-world data, showcasing how **Linear Regression** can be leveraged for predictive analytics.

**7. Kaggle score:**

The project got a public scoring of **0.55864** on **Kaggle.**

