Table of Contents

PROJECT 1 – STUDENT SCORE PREDICTOR

1 Student Score Predictor

1.1 Project Objective

The goal was to predict a student's exam score based on the number of study hours they put in.
We used **Linear Regression** as this is a case of predicting a continuous numerical value.

1.2 Concepts Covered

1.2.1 Supervised Learning

- We had labeled data: hours studied (input) and scores obtained (output).

- The model learns the relationship between these two variables.

1.2.2 Linear Regression

- A statistical method that fits a straight line to data points, expressed as:

$$y = mX + c$$

where:

- yyy = predicted score
- XXX = study hours
- mmm = slope (change in score per hour of study)
- ccc = intercept (score if hours studied = 0)

1.2.3 Cost Function: Mean Squared Error (MSE)

- Measures the average squared difference between predicted and actual values.
- Formula:

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

- Lower MSE = better fit.

1.2.4 Model Evaluation Metrics

- **RMSE** (Root Mean Squared Error): $\sqrt{MSE}$, easier to interpret in the same units as the target.
- **R^2 Score**: Proportion of variance in the dependent variable explained by the model.

1.3 Process Followed

1.3.1 Data Collection:

- We loaded the dataset (hours vs scores).

1.3.2 Data Visualization:

- Used scatter plots to visually confirm a positive correlation.

1.3.3 Splitting the Data:

- Used train_test_split to divide into training and testing sets.

1.3.4 Training the Model:

- Fitted LinearRegression from scikit-learn on training data.

1.3.5 Making Predictions:

- Used the model to predict test set results.

1.4 Evaluation:

- Calculated MSE, RMSE, and R^2 score to judge performance.

1.5 Key Learnings

- Linear regression is best suited when the relationship is linear.
- Overfitting can be avoided by splitting data into train and test sets.
- R^2 close to 1 indicates a strong model fit.

PROJECT 1 – TITANIC SURVIVAL PREDICTOR

2 Titanic Survival Predictor

2.1 Project Objective

The goal was to predict whether a passenger survived the Titanic disaster using passenger details such as class, age, sex, fare, and more. We used **Logistic Regression** as the target variable (survived or not) is binary.

2.2 Concepts Covered

2.2.1 Supervised Learning (Classification)

- Input features: passenger details
- Output label: Survived (1 = survived, 0 = did not survive)

2.2.2 Logistic Regression

- Unlike linear regression, logistic regression predicts probabilities of belonging to a class using the sigmoid function:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

- The output is mapped between 0 and 1, then classified into binary categories.

2.2.3 Feature Encoding

- Converted categorical data (like "male"/"female") into numeric form using one-hot encoding.

2.2.4 Handling Missing Values

- Missing numeric values (e.g., Age) were filled with the median.

- Missing categorical values were filled with the mode.

2.2.5 Feature Scaling

- Used StandardScaler to standardize numerical features so that they have a mean of 0 and standard deviation of 1. This helps gradient-based models converge faster.

2.2.6 Model Evaluation Metrics

- **Accuracy**: Proportion of correct predictions.
- **Confusion Matrix**: Table showing True Positives, False Positives, True Negatives, False Negatives.
- **Precision, Recall, F1-score**:
  - Precision: TP / (TP + FP)
  - Recall: TP / (TP + FN)
  - F1-score: Harmonic mean of precision and recall.

2.3 Process Followed

2.3.1 Data Collection:

- Loaded train.csv from the Titanic dataset.

2.3.2 Data Cleaning & Preprocessing:

- Dropped irrelevant columns like Name, Ticket, Cabin.
- Filled missing values:
  - Age → median value
  - Embarked → most frequent value
- Converted Sex and Embarked into numeric using one-hot encoding.

2.3.3 Splitting the Data:

- Train/test split with 80% training and 20% validation.

2.3.4 Feature Scaling:

- Applied StandardScaler to numerical columns.

2.3.5 Training the Model:

- Fitted LogisticRegression to the scaled training data.

2.3.6 Making Predictions:

- Predicted on the validation set.

2.3.7 Evaluation:

- Calculated accuracy.
- Displayed classification report and confusion matrix.

2.4 Key Learnings

- Logistic regression is ideal for binary classification problems.

- Preprocessing is crucial — models fail if missing values are not handled.
- Scaling improves model stability.
- Metrics beyond accuracy are important when dealing with imbalanced datasets.

---

3 Combined Reflection

Across both projects:

- We moved from **predicting continuous values** (regression) to **predicting categorical outcomes** (classification).
- We learned:
    - How to load and preprocess datasets.
    - How to handle missing data.
    - The difference between regression and classification models.
    - The importance of splitting data and evaluating models using proper metrics.
    - How scaling and encoding impact model performance.