Name : Vaishnavi Eknath Avhad
Class : D15C
Roll No. : 14

Practical 1

Aim : Implement Linear and Logistic Regression on real-world datasets

1. Dataset Source

The dataset used for this experiment is the Medical Cost Personal Dataset.

- Source Link : Kaggle - Medical Cost Personal Datasets

---

2. Dataset Description

This dataset contains 1,338 records representing patient data for health insurance. It is used to predict medical expenses and classify smoking habits.

Feature	Description	Type
age	Age of the primary beneficiary.	Numeric
sex	Gender (male/female).	Categorical
bmi	Body mass index (kg/m^2).	Numeric
children	Number of children/dependents covered.	Numeric
smoker	Smoking status (yes/no).	Categorical (Target for Log. Reg)
region	Residential area in the US.	Categorical
charges	Individual medical costs billed.	Numeric (Target for Lin. Reg)

---

3. Mathematical Formulation

A. Linear Regression

Linear regression models the relationship between a scalar response and one or more explanatory variables.

Equation :

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \epsilon$$

The objective is to minimize the Sum of Squared Errors (SSE) :

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

B. Logistic Regression

Logistic regression is used for binary classification. It uses the Sigmoid Function to output a probability between 0 and 1.

Sigmoid Equation :

$$P(y = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

The model is trained using the Log-Loss (Cross-Entropy) cost function :

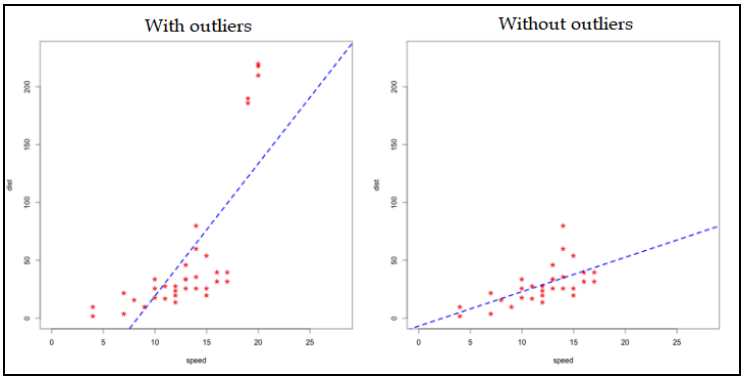
$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} [y^{(i)} \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))]$$

---

4. Algorithm Limitations

A. Linear Regression Limitations

1. Sensitivity to Outliers : Linear Regression uses the "Least Squares" method, which tries to minimize the distance between the line and every data point. A single extreme outlier can pull the entire regression line away from the actual trend, significantly skewing the predictions.

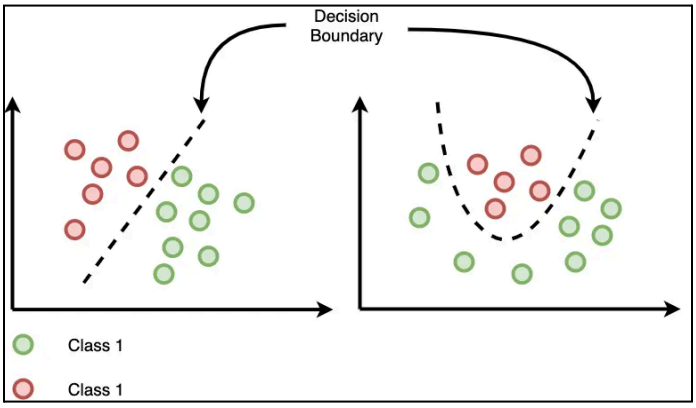


2. Assumption of Linearity : The model assumes that the relationship between the independent variables (X) and the dependent variable (y) is a straight line. If the actual relationship is curved (non-linear), the model will have a high "bias" and perform poorly.

3. Multicollinearity : This occurs when two or more independent variables are highly correlated (e.g., predicting charges using both age and birth_year). This makes it difficult for the model to determine the individual effect of each feature, leading to unstable coefficients.

B. Logistic Regression Limitations

1. Linear Decision Boundary : Logistic Regression is inherently a linear classifier. It attempts to draw a straight line (or a flat plane in higher dimensions) to separate "Smokers" from "Non-Smokers." If the data classes are intertwined in a circular or complex way, Logistic Regression will fail unless you manually create "Interaction Terms" or "Polynomial Features."

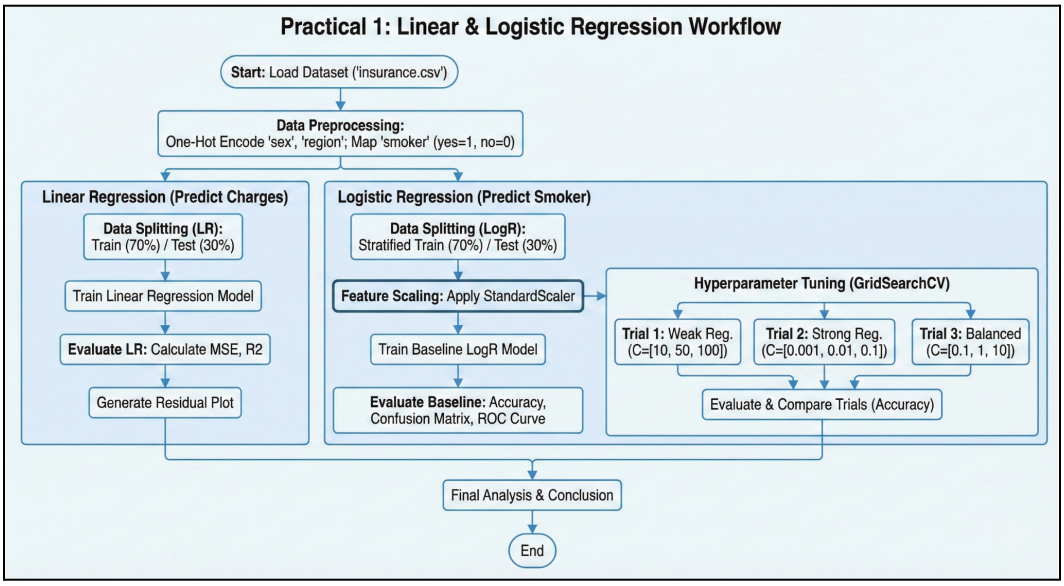


2. Requirement for Large Sample Sizes : Because Logistic Regression relies on Maximum Likelihood Estimation (MLE) rather than Ordinary Least Squares, it requires a substantially larger dataset to reach "convergence." On small datasets, the model may fail to find the optimal coefficients or provide highly inaccurate probability scores.

3. Assumption of Independent Observations : The model assumes that each row in your dataset is independent of the others. If your data includes repeated measurements of the same person over time (Time Series or Grouped data), the standard Logistic Regression model will underestimate the error and provide overconfident results.

---

5. Methodology / Workflow



Step 1 : Data Preprocessing

We convert categorical text data into numerical format using One-Hot Encoding and Mapping.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
# Load and Preprocess
df = pd.read_csv('insurance.csv')
df = pd.get_dummies(df, columns=['sex', 'region'], drop_first=True)
df['smoker'] = df['smoker'].map({'yes': 1, 'no': 0})
```

Step 2 : Linear Regression Implementation

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
X_lr = df.drop('charges', axis=1)
y_lr = df['charges']
```

```
X_train_lr, X_test_lr, y_train_lr, y_test_lr = train_test_split(X_lr, y_lr, test_size=0.3, random_state=42)
linear_model = LinearRegression()
linear_model.fit(X_train_lr, y_train_lr)
y_pred_lr = linear_model.predict(X_test_lr)
```

Step 3: Logistic Regression Implementation

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
X_log = df.drop('smoker', axis=1)
y_log = df['smoker']
X_train_log, X_test_log, y_train_log, y_test_log = train_test_split(X_log, y_log, test_size=0.3, random_state=42, stratify=y_log)
scaler = StandardScaler()
X_train_log = scaler.fit_transform(X_train_log)
X_test_log = scaler.transform(X_test_log)
log_model = LogisticRegression(max_iter=1000)
log_model.fit(X_train_log, y_train_log)
y_pred_log = log_model.predict(X_test_log)
```

---

6. Performance Analysis

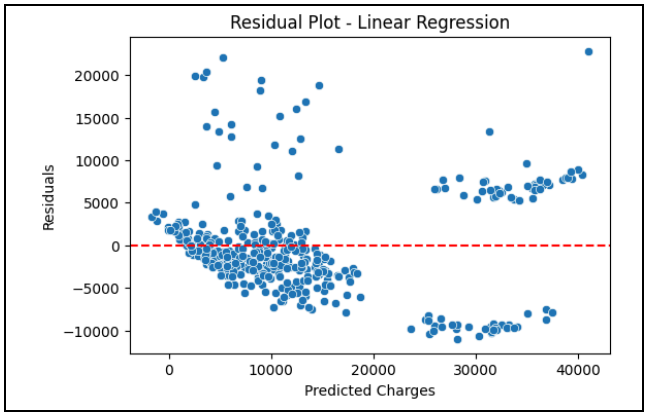
A. Linear Regression Results

Metric	Value
Mean Squared Error (MSE)	33780509.57
R-squared (R^2) Score	0.77

Residual Plot :

A Residual Plot is a graph that shows the residuals (the difference between observed and predicted values) on the vertical axis and the predicted values on the horizontal axis.

- What it represents : It visualizes the "errors" of the model.
- Ideal Result : You want to see a random scatter of points around the horizontal line (zero). This indicates that the model's errors are random and that a linear model is appropriate for the data.
- Warning Signs : If the points form a pattern (like a "U" shape or a funnel), it suggests that your data might be non-linear or that "heteroscedasticity" is present, meaning your model's accuracy changes depending on the price of the insurance.



B. Logistic Regression Results

Metric	Value
Accuracy	0.95

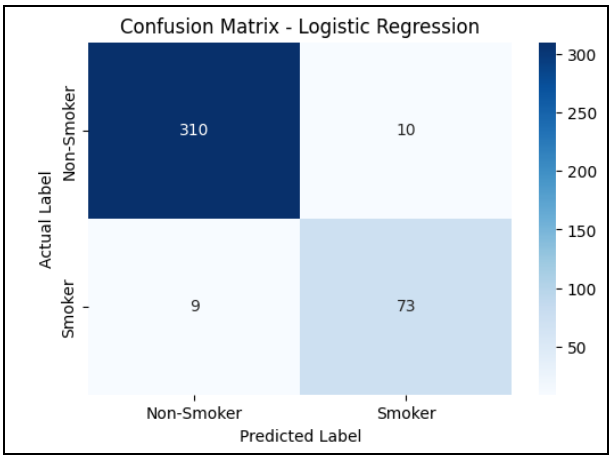
Classification Report :

```
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.97      0.97       320
           1       0.88      0.89      0.88        82

    accuracy                           0.95       402
   macro avg       0.93      0.93      0.93       402
weighted avg       0.95      0.95      0.95       402
```

Confusion Matrix & ROC Curve :

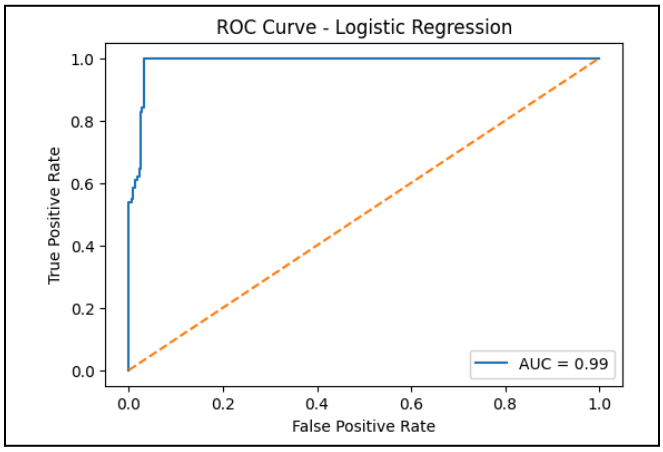
The Confusion Matrix is a 2X2 table that summarizes the prediction results on a classification problem.

- What it represents : It breaks down the correct and incorrect predictions into four categories:
    - True Positive (TP) : Correctly predicted as a smoker.
    - True Negative (TN) : Correctly predicted as a non-smoker.
    - False Positive (FP) : Predicted as a smoker, but is actually a non-smoker (Type I Error).
    - False Negative (FN) : Predicted as a non-smoker, but is actually a smoker (Type II Error).
- Goal : You want high numbers in the diagonal (Top-Left and Bottom-Right) and low numbers elsewhere.



The Receiver Operating Characteristic (ROC) curve is a plot of the True Positive Rate (Sensitivity) against the False Positive Rate (1 - Specificity) at various threshold settings.

- What it represents : It shows the trade-off between sensitivity and specificity. The AUC (Area Under the Curve) tells you how well the model is distinguishing between the two classes.
- Ideal Result : An AUC of 1.0 represents a perfect model. An AUC of 0.5 (the diagonal dashed line) represents a model that is no better than random guessing.
- Interpretation : For your insurance dataset, a high AUC means the model is very good at identifying who is a smoker versus who is not based on the other features.



---

7. Hyperparameter Tuning

We used GridSearchCV to optimize the Logistic Regression model by testing different values of the regularization parameter C.

code

```
from sklearn.model_selection import GridSearchCV
trials = {
    "Trial 1 (Weak Regularization)": {'C': [10, 50, 100], 'solver': ['liblinear']},
    "Trial 2 (Strong Regularization)": {'C': [0.001, 0.01, 0.1], 'solver': ['liblinear']},
    "Trial 3 (Balanced Range)": {'C': [0.1, 1, 10], 'solver': ['liblinear']}
}
results = []
print("\n--- Running Hyperparameter Tuning Trials ---")
for trial_name, param_grid in trials.items():
    grid = GridSearchCV(
        LogisticRegression(max_iter=1000),
        param_grid,
        cv=5,
        scoring='accuracy'
    )
    grid.fit(X_train_log, y_train_log)
    best_model = grid.best_estimator_
    y_pred_trial = best_model.predict(X_test_log)
    trial_accuracy = accuracy_score(y_test_log, y_pred_trial)
    results.append({
        "Trial": trial_name,
        "Best C": grid.best_params_['C'],
        "Accuracy": trial_accuracy
    })
    print(f"{trial_name} completed.")
print("\n================ HYPERPARAMETER TUNING SUMMARY
================\n")
results_df = pd.DataFrame(results)
print(results_df.to_string(index=False))
best_trial = results_df.loc[results_df['Accuracy'].idxmax()]
print(f"\nConclusion: {best_trial['Trial']} performed best with an accuracy of
{best_trial['Accuracy']:.4f}")
```

Comparative Results

```
--- Running Hyperparameter Tuning Trials ---
Trial 1 (Weak Regularization) completed.
Trial 2 (Strong Regularization) completed.
Trial 3 (Balanced Range) completed.

================ HYPERPARAMETER TUNING SUMMARY ================

                         Trial  Best C  Accuracy
  Trial 1 (Weak Regularization)    10.0  0.957711
Trial 2 (Strong Regularization)     0.1  0.942786
       Trial 3 (Balanced Range)     1.0  0.960199

Conclusion: Trial 3 (Balanced Range) performed best with an accuracy of 0.9602
```

Tuning Interpretation :

- Higher Accuracy : Indicates the optimal balance between bias and variance.
- Impact of 'C' : Smaller values of C increase regularization strength, which helps if the model is overfitting, whereas larger values allow the model to fit the training data more closely.

---

8. Conclusion

In this practical, Linear Regression successfully predicted medical charges with an R^2 score that indicates the percentage of variance explained. Logistic Regression effectively classified smokers with high accuracy, further improved by hyperparameter tuning. Comparing the three trials in tuning showed that selecting the right regularization strength is critical for model stability.