Assignment-1 Report

Introduction

This report contains two experiments using datasets available in scikit-learn.

Linear Regression (Diabetes)

1. Introduction

- Explain what Linear Regression is:
  A statistical method used to model the relationship between an independent variable (feature) and a dependent variable (target).
- why this problem is important:
  Diabetes progression prediction helps in healthcare and treatment planning.

2. Dataset Description

- Dataset: Diabetes dataset from Scikit-learn.
- Contains 442 samples with 10 baseline variables such as age, sex, BMI, blood pressure, etc.
- I use only one feature (BMI or blood sugar-related measure) for visualization.
- Target: A quantitative measure of disease progression after one year.

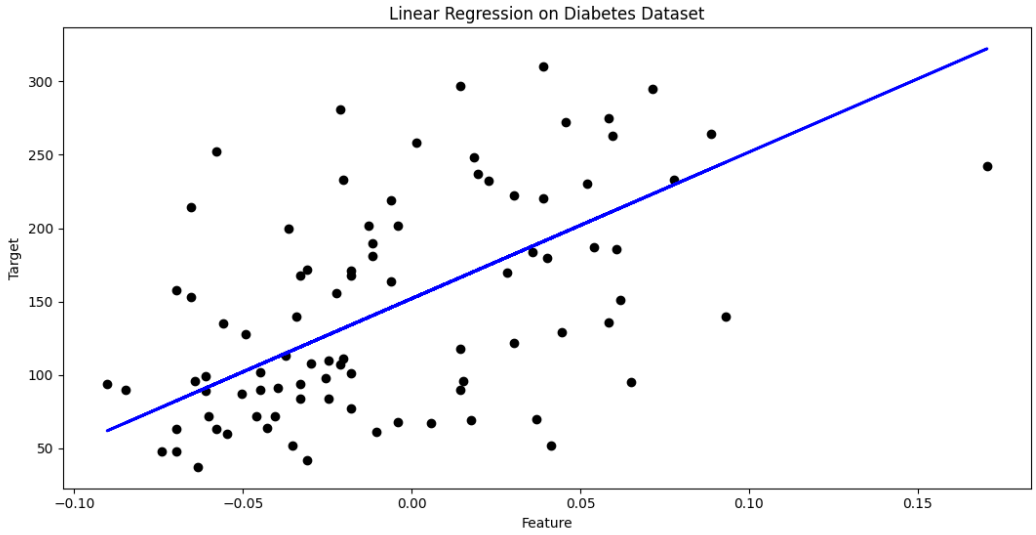
3. Methodology

- Step 1: Load dataset using load_diabetes().
- Step 2: Select one feature (X = diabetes.data[:, np.newaxis, 2]).
- Step 3: Split dataset into training (80%) and testing (20%).
- Step 4: Train model using LinearRegression().
- Step 5: Predict on test data.
- Step 6: Evaluate using Mean Squared Error (MSE) and R^2 score.
- Step 7: Visualize results using matplotlib (scatter + regression line).

Code:

```
# linear_regression_diabetes.py
import matplotlib.pyplot as plt
import numpy as np
from sklearn.datasets import load_diabetes
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Load dataset
diabetes = load_diabetes()
X = diabetes.data[:, np.newaxis, 2]   # take just one feature for visualization
y = diabetes.target

# Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train model
model = LinearRegression()
model.fit(X_train, y_train)

# Predict
y_pred = model.predict(X_test)

# Evaluate
print("Mean squared error:", mean_squared_error(y_test, y_pred))
print("R2 score:", r2_score(y_test, y_pred))

# Plot
plt.scatter(X_test, y_test, color="black")
plt.plot(X_test, y_pred, color="blue", linewidth=2)
plt.title("Linear Regression on Diabetes Dataset")
plt.xlabel("Feature")
plt.ylabel("Target")
plt.show()
```

OUTPUT



Logistic Regression (Breast Cancer)

1. Introduction

- Explain what Linear Regression is:
  A statistical method used to model the relationship between an independent variable (feature) and a dependent variable (target).
- why this problem is important:
  Diabetes progression prediction helps in healthcare and treatment planning.

2. Dataset Description

- Dataset: Breast Cancer Wisconsin dataset from Scikit-learn.
- Contains 569 samples with 30 features (like mean radius, texture, smoothness, etc.).
- Target:
  - 0 → Malignant (cancerous)
  - 1 → Benign (non-cancerous)

3. Methodology

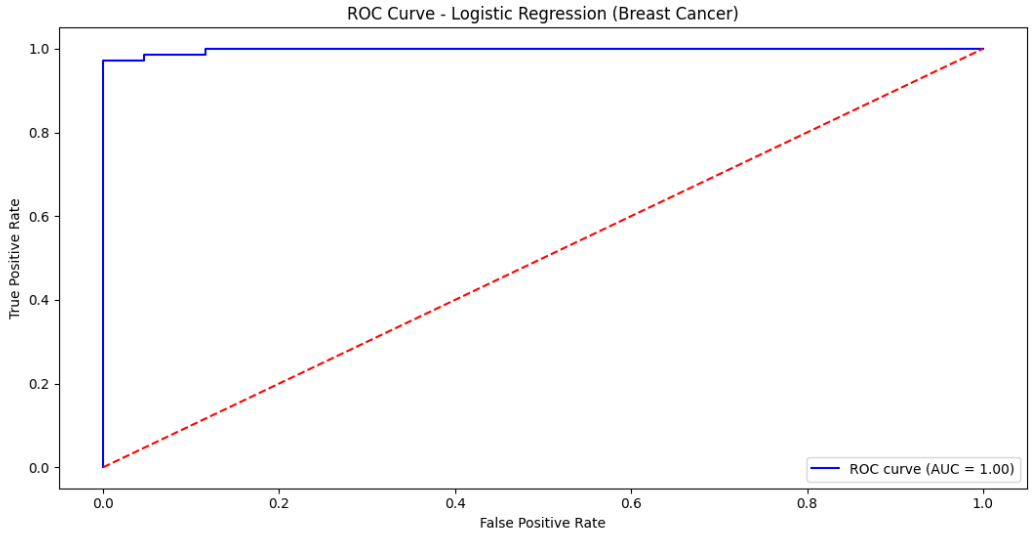
- Step 1: Load dataset using load_breast_cancer().
- Step 2: Split dataset into training (80%) and testing (20%).
- Step 3: Train model using LogisticRegression(max_iter=5000).
- Step 4: Predict outcomes on test data.
- Step 5: Evaluate with:
  - Confusion Matrix (True Positive, True Negative, False Positive, False Negative).
  - Classification Report (Precision, Recall, F1-score, Accuracy).
  - ROC Curve with AUC (measures how well the model distinguishes between classes).

Code:

```
# logistic_regression_breastcancer.py
import matplotlib.pyplot as plt
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, classification_report, roc_curve, auc

# Load dataset
cancer = load_breast_cancer()
X = cancer.data
y = cancer.target

# Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train model
model = LogisticRegression(max_iter=5000)
model.fit(X_train, y_train)
```

```
# Predict
y_pred = model.predict(X_test)


# Evaluate
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("Report:\n", classification_report(y_test, y_pred))

# ROC Curve
y_prob = model.predict_proba(X_test)[:, 1]
fpr, tpr, _ = roc_curve(y_test, y_prob)
roc_auc = auc(fpr, tpr)

plt.plot(fpr, tpr, color="blue", label=f"ROC curve (AUC = {roc_auc:.2f})")
plt.plot([0, 1], [0, 1], color="red", linestyle="--")
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC Curve - Logistic Regression (Breast Cancer)")
plt.legend()
plt.show()
```

OUTPUT



Conclusion

Both models demonstrate simple applications of Linear and Logistic Regression using scikit-learn datasets.