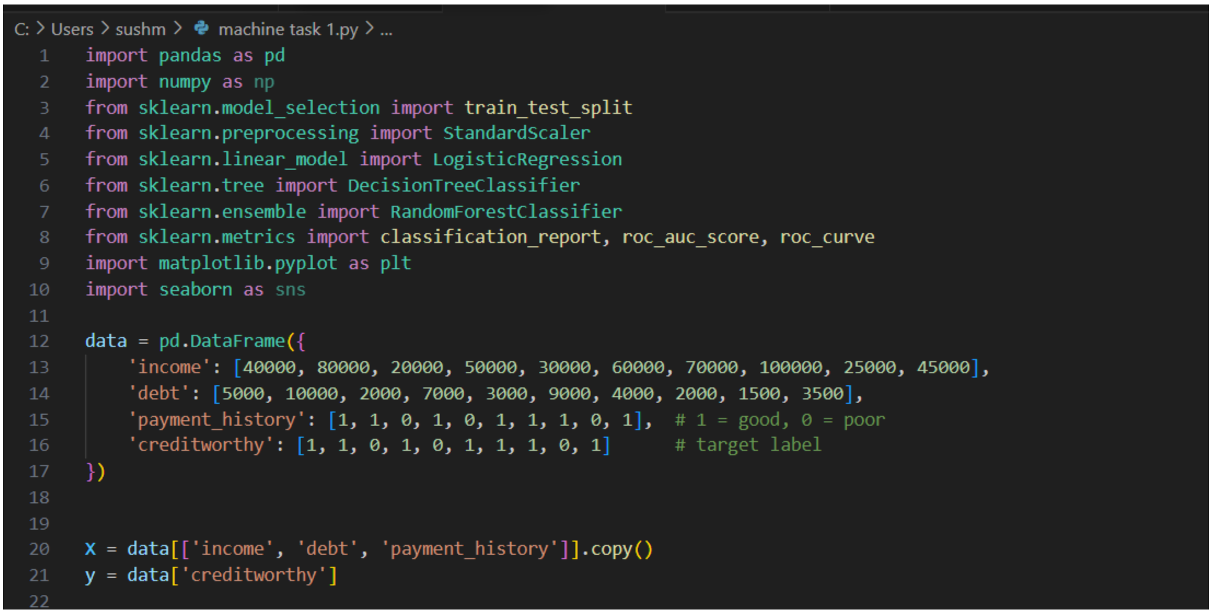
week 1 task

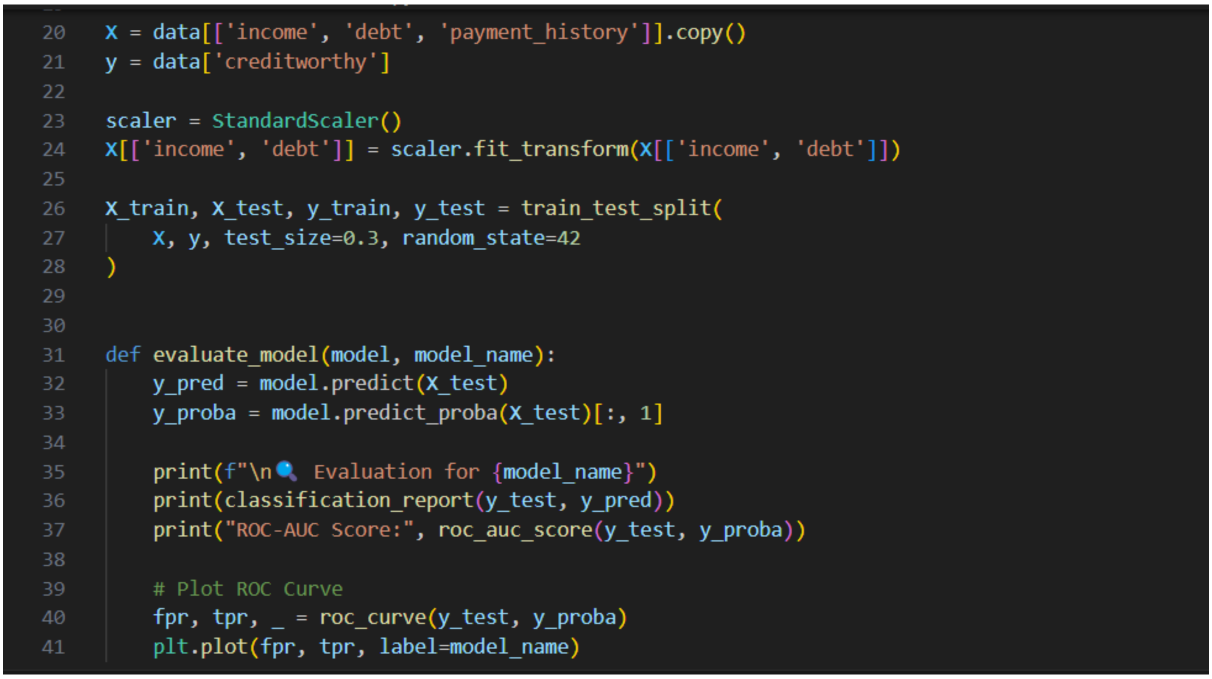
TASK 1: Credit Scoring Model

Objective: Predict an individual's creditworthiness using past financial data.
Approach: Use classification algorithms like Logistic Regression, Decision Trees, or Random Forest.
Key Features:

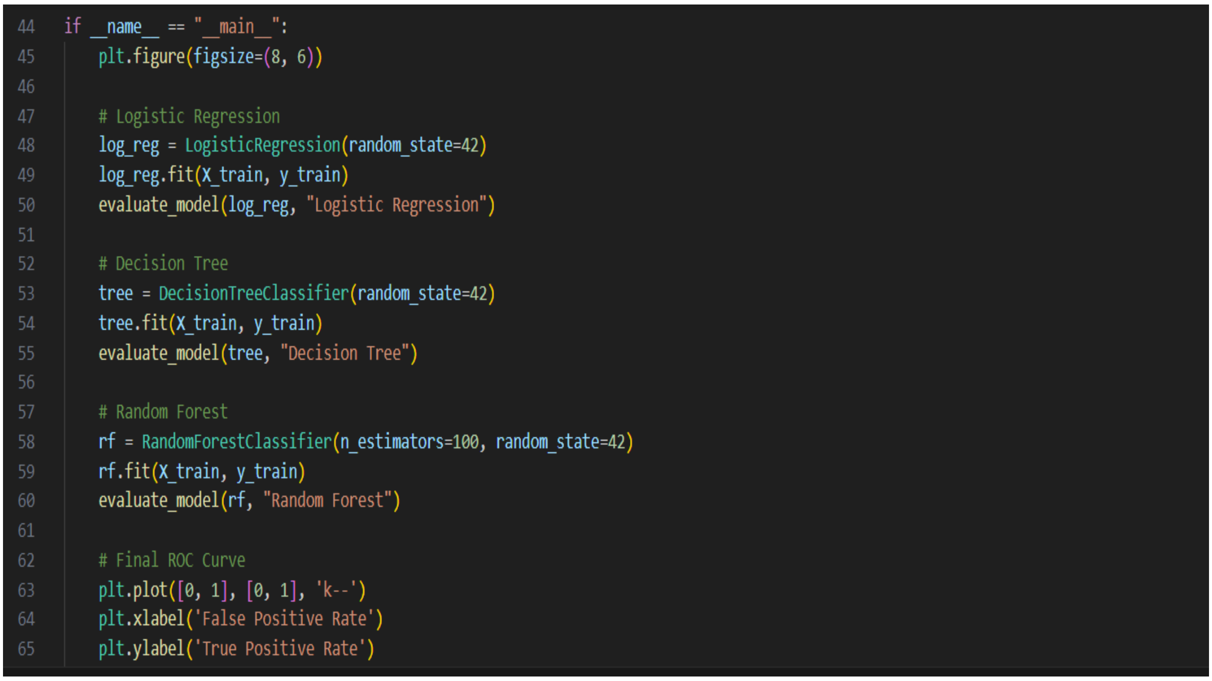
- Feature engineering from financial history.
- Model accuracy assessment using metrics like Precision, Recall, F1 -Score, ROC-AUC.
- Dataset could include: income, debts, payment history, etc.



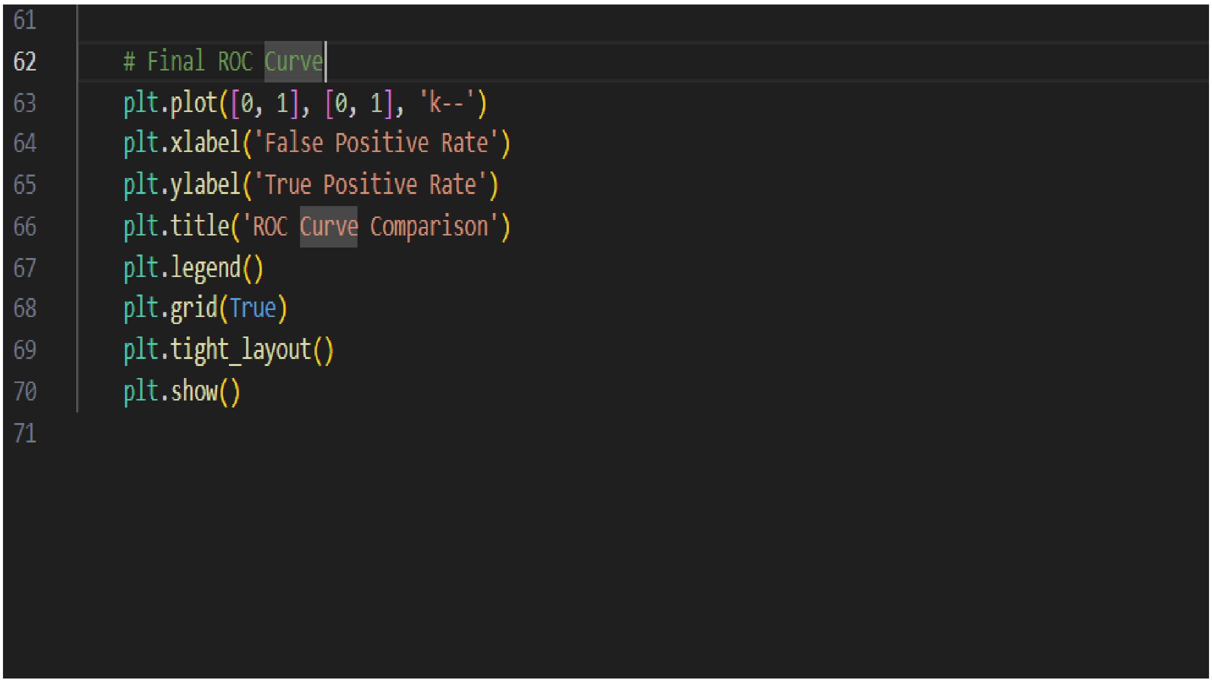
```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, roc_auc_score, roc_curve
import matplotlib.pyplot as plt
import seaborn as sns

data = pd.DataFrame({
    'income': [40000, 80000, 20000, 50000, 30000, 60000, 70000, 100000, 25000, 45000],
    'debt': [5000, 10000, 2000, 7000, 3000, 9000, 4000, 2000, 1500, 3500],
    'payment_history': [1, 1, 0, 1, 0, 1, 1, 1, 0, 1],  # 1 = good, 0 = poor
    'creditworthy': [1, 1, 0, 1, 0, 1, 1, 1, 0, 1]      # target label
})


X = data[['income', 'debt', 'payment_history']].copy()
y = data['creditworthy']

```



```python
X = data[['income', 'debt', 'payment_history']].copy()
y = data['creditworthy']

scaler = StandardScaler()
X[['income', 'debt']] = scaler.fit_transform(X[['income', 'debt']])

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42
)


def evaluate_model(model, model_name):
    y_pred = model.predict(X_test)
    y_proba = model.predict_proba(X_test)[:, 1]

    print(f"\n🔍 Evaluation for {model_name}")
    print(classification_report(y_test, y_pred))
    print("ROC-AUC Score:", roc_auc_score(y_test, y_proba))

    # Plot ROC Curve
    fpr, tpr, _ = roc_curve(y_test, y_proba)
    plt.plot(fpr, tpr, label=model_name)
```

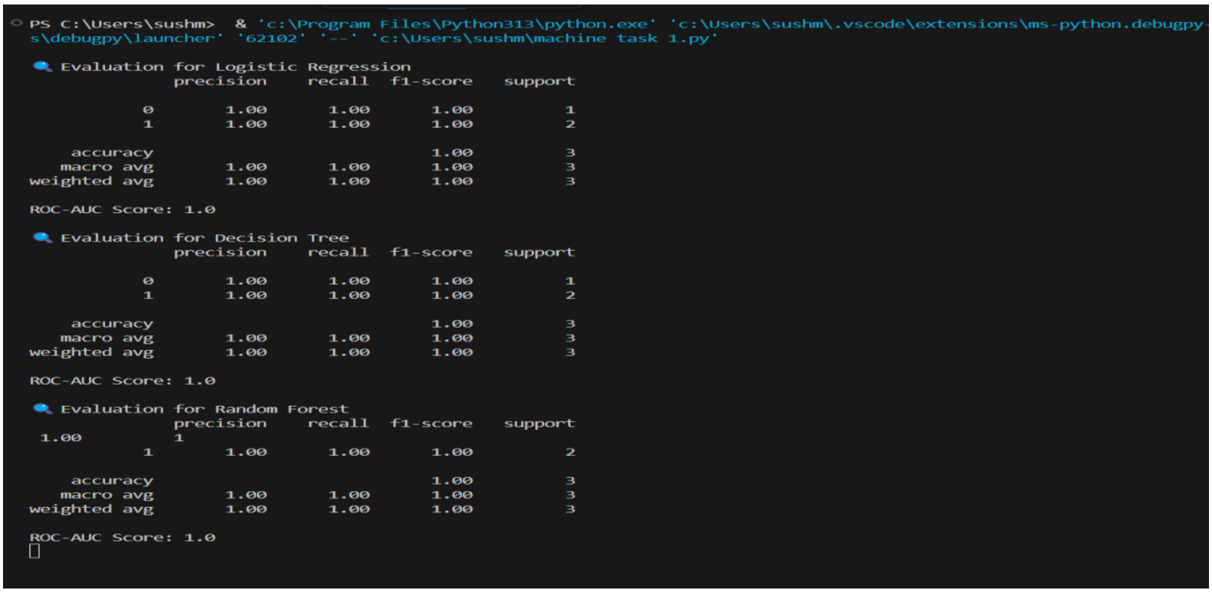


```python
if __name__ == "__main__":
    plt.figure(figsize=(8, 6))

    # Logistic Regression
    log_reg = LogisticRegression(random_state=42)
    log_reg.fit(X_train, y_train)
    evaluate_model(log_reg, "Logistic Regression")

    # Decision Tree
    tree = DecisionTreeClassifier(random_state=42)
    tree.fit(X_train, y_train)
    evaluate_model(tree, "Decision Tree")

    # Random Forest
    rf = RandomForestClassifier(n_estimators=100, random_state=42)
    rf.fit(X_train, y_train)
    evaluate_model(rf, "Random Forest")

    # Final ROC Curve
    plt.plot([0, 1], [0, 1], 'k--')
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
```



```python

    # Final ROC Curve
    plt.plot([0, 1], [0, 1], 'k--')
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title('ROC Curve Comparison')
    plt.legend()
    plt.grid(True)
    plt.tight_layout()
    plt.show()

```

Output:-



```
○ PS C:\Users\sushm>  & 'c:\Program Files\Python313\python.exe' 'c:\Users\sushm\.vscode\extensions\ms-python.debugpy-
  s\debugpy\launcher' '62102' '--' 'c:\Users\sushm\machine task 1.py'

  🔍 Evaluation for Logistic Regression
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         1
           1       1.00      1.00      1.00         2

    accuracy                           1.00         3
   macro avg       1.00      1.00      1.00         3
weighted avg       1.00      1.00      1.00         3

ROC-AUC Score: 1.0

  🔍 Evaluation for Decision Tree
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         1
           1       1.00      1.00      1.00         2

    accuracy                           1.00         3
   macro avg       1.00      1.00      1.00         3
weighted avg       1.00      1.00      1.00         3

ROC-AUC Score: 1.0

  🔍 Evaluation for Random Forest
              precision    recall  f1-score   support
 1.00                  1
           1       1.00      1.00      1.00         2

    accuracy                           1.00         3
   macro avg       1.00      1.00      1.00         3
weighted avg       1.00      1.00      1.00         3

ROC-AUC Score: 1.0
▯
```

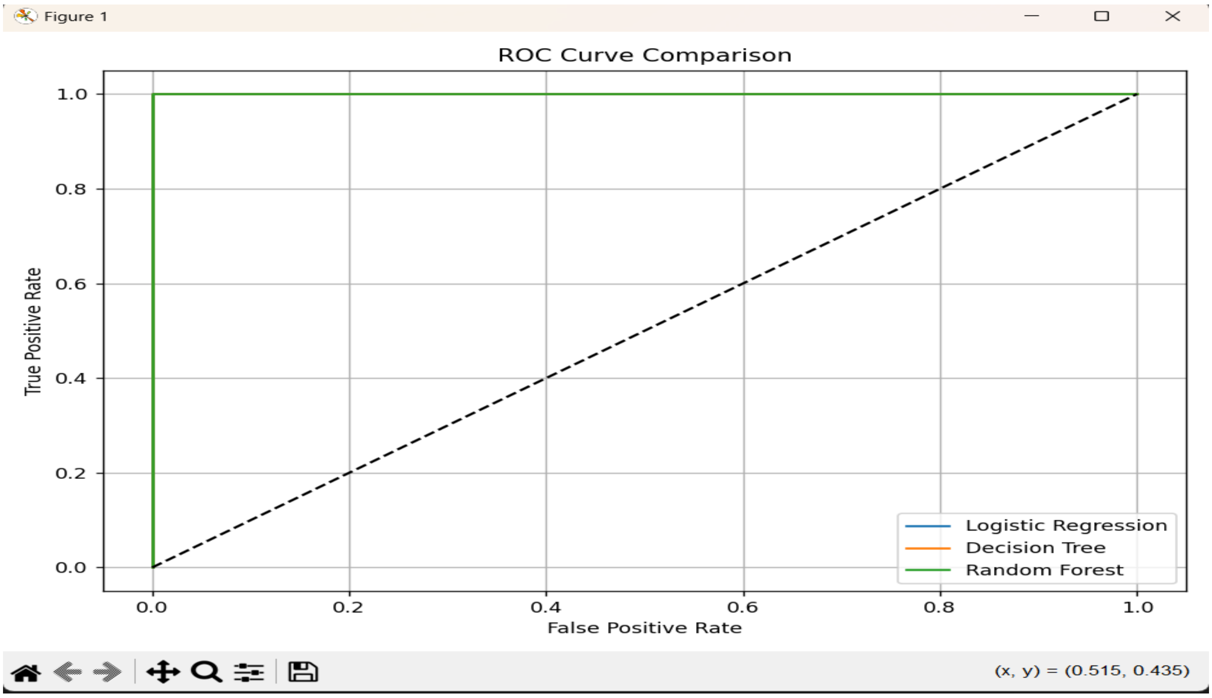


Figure 1
ROC Curve Comparison
True Positive Rate
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
Logistic Regression
Decision Tree
Random Forest
(x, y) = (0.515, 0.435)

Machine Learning Tasks & Instructions — CodeAlpha

Machine Learning — Internship Overview

This internship program provides hands-on experience in machine learning algorithms and model development. CodeAlpha is a leading software development company driving innovation through AI and intelligent systems. The internship empowers students to work with Python, Scikit-learn, TensorFlow, and other ML libraries to build and train models for real-world applications. Interns will learn data preprocessing, supervised and unsupervised learning, model evaluation and optimization techniques. With expert mentorship and live projects, interns will gain practical knowledge in deploying machine learning solutions to solve complex problems.

Internship Perks

- Internship Offer Letter
- Completion Certificate (QR Verified)
- Unique ID Certificate
- Letter of Recommendation (based on performance)
- Job Opportunities / Placement Support
- Resume Building Support

Instructions for Interns

1. Share your internship status on **LinkedIn**, tagging **@CodeAlpha**.
2. Complete the **assigned projects** within the mentioned time frame.
3. Upload your complete source code to **GitHub** in a repository named: `CodeAlpha_ProjectName`
4. Post a **video explanation** of your project on LinkedIn with GitHub repo link.
5. Submit your completed task using the **Submission Form**.
6. **Complete any 3 or 2 out of the 4 tasks** listed below (from your domain).

Machine Learning Task List

(Complete any 2 or 3 of the following tasks)

TASK 1: Credit Scoring Model

Objective: Predict an individual's **creditworthiness** using past financial data.
Approach: Use **classification algorithms** like Logistic Regression, Decision Trees, or Random Forest.
Key Features:

- Feature engineering from financial history.
- Model accuracy assessment using metrics like **Precision, Recall, F1-Score, ROC-AUC**.
- Dataset could include: income, debts, payment history, etc.
