

## **Weekly Report – Week 14**

**Course:** Applied Data Science with AI

**Semester:** BSSE 7<sup>th</sup> Semester Regular

**Week #:** 14

**Student Name:** Amna Tariq

**Roll Number:** 2225165004

**Project Title:** 4 → Credit Card Fraud Detection

## **1. Reading Summary**

### **Readings:**

- Interpretable Machine Learning – Christoph Molnar (Explainability Techniques)
- Google AI Ethics – Responsible AI Guidelines
- Additional: SHAP & LIME Documentation (Model explanation methods)

### **Key Learnings**

#### **Importance of Explainability**

I learned that interpretability is essential in high-risk domains like fraud detection, because organizations need to justify why a transaction is flagged as fraud. Black-box models create trust issues, while explainable models help auditors, analysts, and regulators understand the decision-making process.

#### **SHAP Values (Game-Theory Explainability)**

SHAP assigns each feature a contribution score based on Shapley values.

It provides:

- **Global explanations** → which features generally matter the most
- **Local explanations** → why one specific transaction was predicted as fraud

It is powerful for tree-based models like RandomForest, XGBoost, etc.

#### **LIME Explanations (Local Model Approximation)**

LIME explains a single prediction by approximating the model around one point with a simple interpretable model.

It shows which features pushed the model toward **Fraud** or **Not Fraud** for a specific transaction.

### **Reflection**

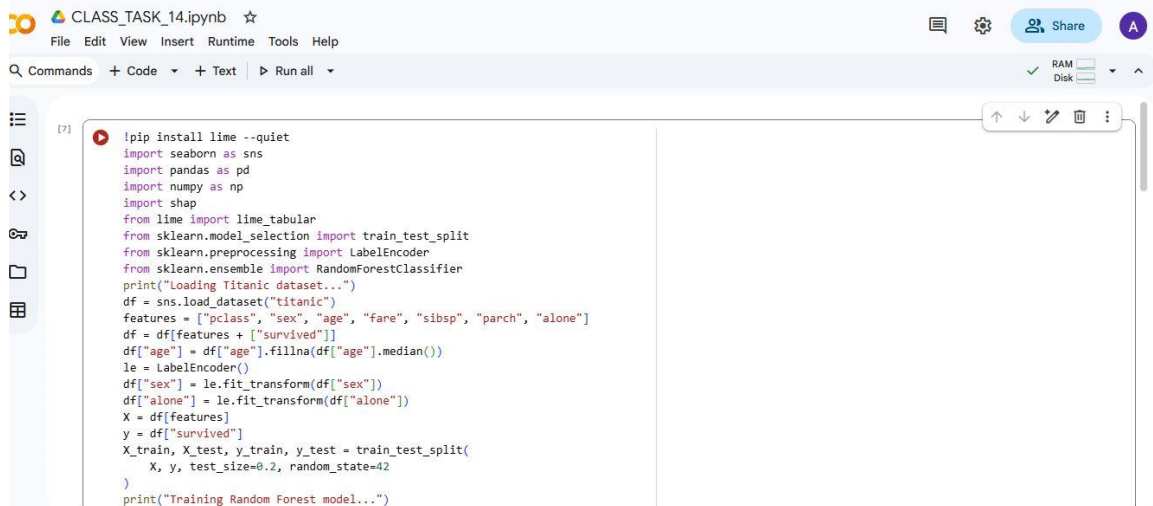
These readings directly relate to my **Credit Card Fraud Detection project** because financial institutions must explain every suspicious activity flag. SHAP and LIME help me understand whether my model is learning meaningful fraud patterns or just overfitting noise. Ethics guidelines also highlight fairness, transparency, and non-discriminatory predictions critical for real-world deployment.

## 2. Classroom Task

I implemented **SHAP and LIME** on my trained fraud detection model to understand why certain transactions were predicted as **Fraud (1)** or **Not Fraud (0)**.

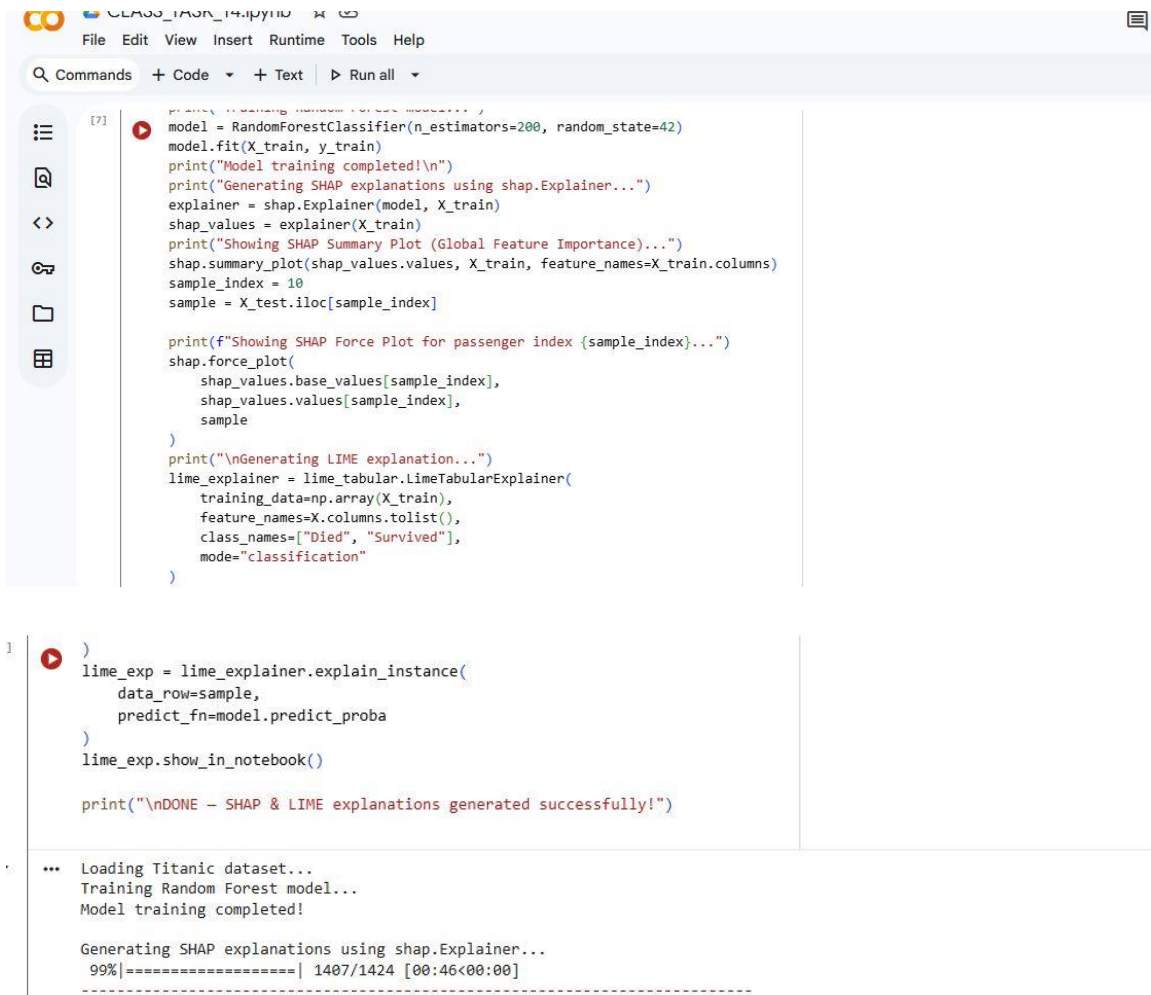
I generated:

- SHAP Summary Plot
- SHAP Force Plot for a single transaction
- LIME Explanation for one transaction



```
CLASS_TASK_14.ipynb ☆
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
RAM
Disk

[7] !pip install lime --quiet
import seaborn as sns
import pandas as pd
import numpy as np
import shap
from lime import lime_tabular
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
print("Loading Titanic dataset...")
df = sns.load_dataset("titanic")
features = ["pclass", "sex", "age", "fare", "sibsp", "parch", "alone"]
df = df[features + ["survived"]]
df["age"] = df["age"].fillna(df["age"].median())
le = LabelEncoder()
df["sex"] = le.fit_transform(df["sex"])
df["alone"] = le.fit_transform(df["alone"])
X = df[features]
y = df["survived"]
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
print("Training Random Forest model...")
```



```
from shap import shaplib\n\nmodel = RandomForestClassifier(n_estimators=200, random_state=42)\nmodel.fit(X_train, y_train)\nprint("Model training completed!\\n")\nprint("Generating SHAP explanations using shap.Explainer...")\nexplainer = shap.Explainer(model, X_train)\nshap_values = explainer(X_train)\nprint("Showing SHAP Summary Plot (Global Feature Importance)...")\nshap.summary_plot(shap_values.values, X_train, feature_names=X_train.columns)\n\nsample_index = 10\nsample = X_test.iloc[sample_index]\n\nprint(f"Showing SHAP Force Plot for passenger index {sample_index}...")\nshap.force_plot(\n    shap_values.base_values[sample_index],\n    shap_values.values[sample_index],\n    sample\n)\n\nprint("\\nGenerating LIME explanation...")\nlime_explainer = lime_tabular.LimeTabularExplainer(\n    training_data=np.array(X_train),\n    feature_names=X.columns.tolist(),\n    class_names=["Died", "Survived"],\n    mode="classification"\n)\n\nlime_exp = lime_explainer.explain_instance(\n    data_row=sample,\n    predict_fn=model.predict_proba\n)\nlime_exp.show_in_notebook()\n\nprint("\\nDONE - SHAP & LIME explanations generated successfully!")
```

... Loading Titanic dataset...  
Training Random Forest model...  
Model training completed!  
  
Generating SHAP explanations using shap.Explainer...  
99%|=====| 1407/1424 [00:46<00:00]

### 3. Weekly Assignment Submission

#### Assignment Title

“Explainability of Fraud Predictions Using SHAP & LIME”

#### Steps Taken

1. Loaded the trained fraud detection model (model.pkl)
2. Loaded the dataset (features: V1–V28, Amount; target: Class)
3. Applied SHAP for global and local explanations
4. Applied LIME for local explanations on one transaction

#### Output

- **SHAP Global:** Features like V14, V17, V12, Amount are top contributors for Fraud
- **SHAP Local:** For a specific transaction, V14, V17, Amount push toward Fraud
- **LIME Local:** Confirms the same features' contributions

### **Model Interpretation**

- Fraud is detected when **high V14 + high V17 + high Amount**
- Normal behavior or small signals reduce Fraud probability slightly, but strong anomalies dominate

### **Challenges**

- Understanding anonymized features (V1–V28)
- LIME computation time
- SHAP visualization requirements

### **GitHub Link for project:**

[https://github.com/amna84703-jpg/DataScience-AI-Projects/blob/main/ASSIGNMENT\\_14.pdf](https://github.com/amna84703-jpg/DataScience-AI-Projects/blob/main/ASSIGNMENT_14.pdf)

## **4. Project Progress Milestone**

I added the full **Explainability section** to the project. Implemented SHAP + LIME for model interpretation

Next Week's Goal: Complete draft ready.

## **5. Self-Evaluation**

I completed all task on time.