Assignment -3

Batch - 35

Hall Ticket - 2303A52330

Name - Rithwik Bandi

1	**Assignment Problem 1: Bank Marketing Campaign**	

Problem Statement:

Bank dataset predicts deposit subscription. LIME shows why a client said yes/no.

Tasks:

1. Load dataset
2. Train Gradient Boosting
3. Explain with LIME
4. Visualize contributions
5. Discuss marketing insights

Deliverables:

- Code
- Outputs
- Report

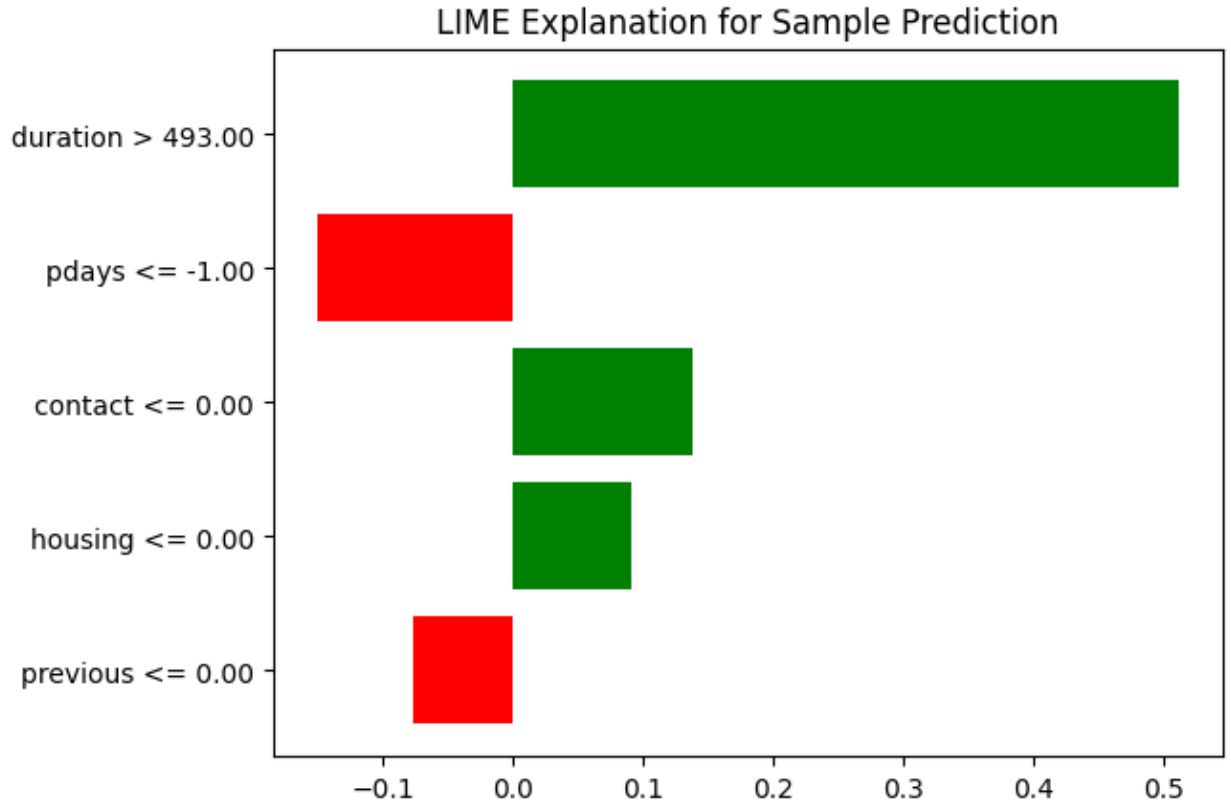
Problem 1

Code

```
import os
import pandas as pd
import kagglehub
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.metrics import classification_report, accuracy_score
import lime
import lime.lime_tabular
import matplotlib.pyplot as plt

path = kagglehub.dataset_download("janiobachmann/bank-marketing-dataset")
print("Path to dataset files:", path)

files = os.listdir(path)
csv_files = [f for f in files if f.endswith(".csv")]
if not csv_files:
    raise FileNotFoundError("No CSV file found in dataset folder")
csv_path = os.path.join(path, csv_files[0])
```

```
print("Using CSV file:", csv_path)

df = pd.read_csv(csv_path)
print("Dataset shape:", df.shape)
print(df.head())

label_encoders = {}
for col in df.select_dtypes(include=['object']).columns:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col])
    label_encoders[col] = le

target_col = "deposit" if "deposit" in df.columns else df.columns[-1]
X = df.drop(target_col, axis=1)
y = df[target_col]

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

model = GradientBoostingClassifier(random_state=42)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))

explainer = lime.lime_tabular.LimeTabularExplainer(
    training_data=X_train.values,
    feature_names=X_train.columns,
    class_names=['no', 'yes'],
    mode='classification'
)

i = 10
exp = explainer.explain_instance(
    X_test.iloc[i].values,
    model.predict_proba,
    num_features=5
)

print("\nLIME explanation for instance:", i)
print(exp.as_list())

fig = exp.as_pyplot_figure()
plt.title("LIME Explanation for Sample Prediction”)
plt.show()
```

```
Path to dataset files: /kaggle/input/bank-marketing-dataset
Using CSV file: /kaggle/input/bank-marketing-dataset/bank.csv
Dataset shape: (11162, 17)
   age           job  marital  education default  balance housing loan  contact  \
0   59        admin.  married  secondary      no     2343     yes   no  unknown
1   56        admin.  married  secondary      no       45      no   no  unknown
2   41   technician  married  secondary      no     1270     yes   no  unknown
3   55     services  married  secondary      no     2476     yes   no  unknown
4   54        admin.  married   tertiary      no      184      no   no  unknown

   day month  duration  campaign  pdays  previous poutcome deposit
0    5   may      1042         1     -1         0  unknown     yes
1    5   may      1467         1     -1         0  unknown     yes
2    5   may      1389         1     -1         0  unknown     yes
3    5   may       579         1     -1         0  unknown     yes
4    5   may       673         2     -1         0  unknown     yes
Accuracy: 0.8217644424540976

Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.81      0.83      1166
           1       0.80      0.84      0.82      1067

    accuracy                           0.82      2233
   macro avg       0.82      0.82      0.82      2233
weighted avg       0.82      0.82      0.82      2233

/usr/local/lib/python3.12/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but GradientBoostingClassifier was fitted with feature na
  warnings.warn(
```



2	**Assignment Problem 2: House Price Classification**	

Problem Statement:

California dataset classifies expensive vs cheap houses. LIME explains predictions.

Tasks:

1. Load dataset
2. Prepare binary target
3. Train Random Forest
4. Apply LIME
5. Interpret results

Deliverables:

- Code
- Outputs
- Report

Problem 2

Code

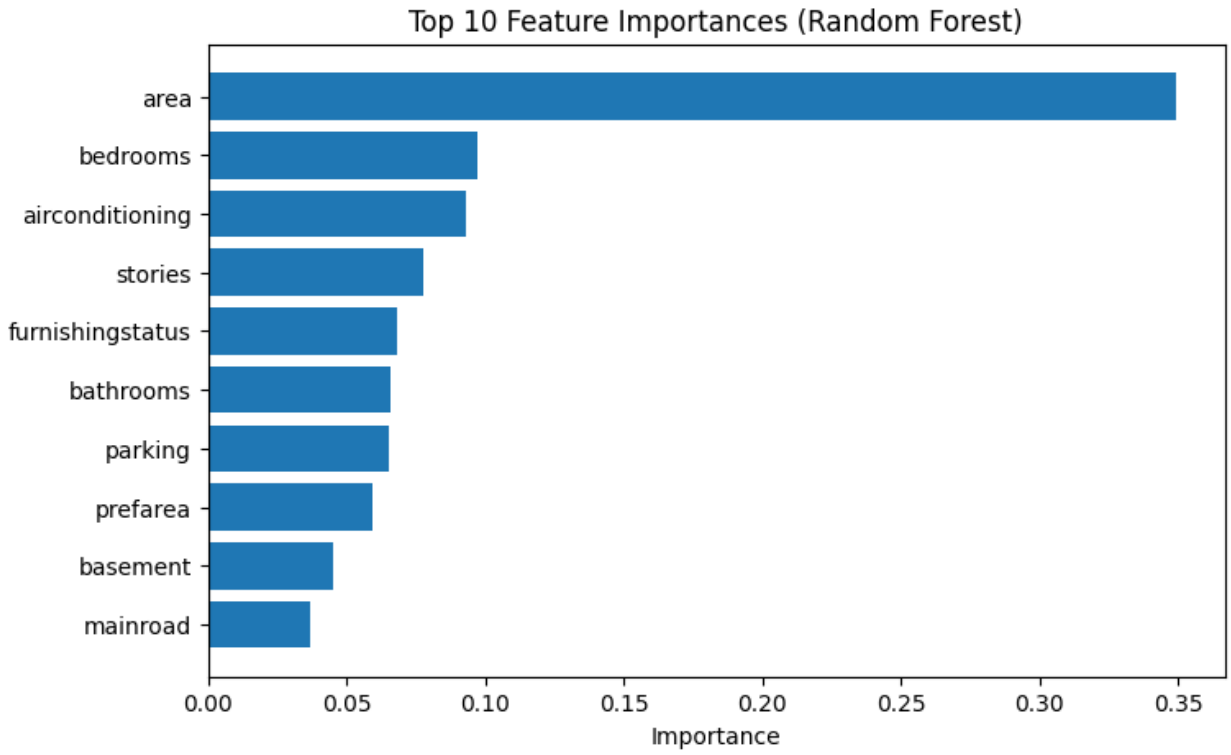
```
import os
import pandas as pd
import kagglehub
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, accuracy_score
import lime
import lime.lime_tabular

path = kagglehub.dataset_download("yasserh/housing-prices-dataset")
print("Path to dataset files:", path)

files = os.listdir(path)
csv_files = [f for f in files if f.endswith(".csv")]
if not csv_files:
    raise FileNotFoundError("No CSV file found in dataset folder")
csv_path = os.path.join(path, csv_files[0])
print("Using CSV file:", csv_path)

df = pd.read_csv(csv_path)
print("Dataset shape:", df.shape)
print(df.head())
```

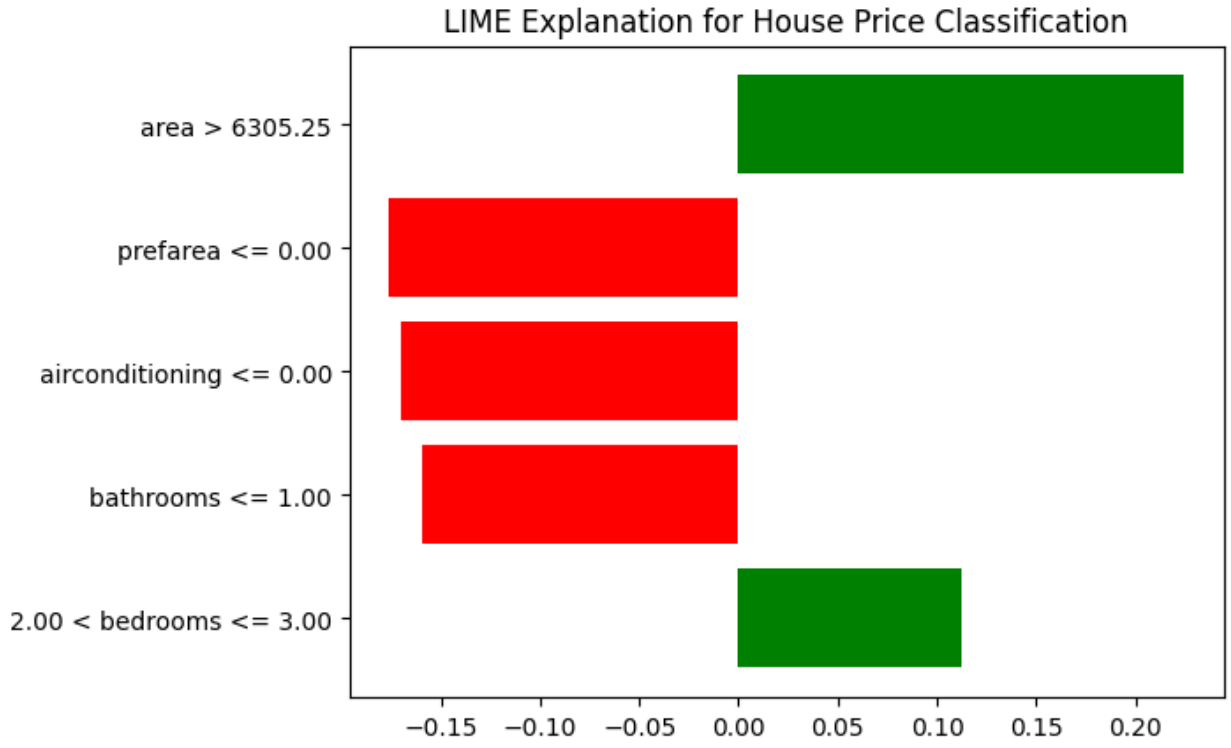
```
price_candidates = [c for c in df.columns if "price" in c.lower() or c.lower() in ["medv"]]
if not price_candidates:
    raise ValueError(f"Could not find price column. Columns: {list(df.columns)}")
target_col = price_candidates[0]
print("Detected price column:", target_col)

df = df.dropna(subset=[target_col])

median_price = df[target_col].median()
df["expensive"] = (df[target_col] > median_price).astype(int)

X = df.drop([target_col, "expensive"], axis=1)
y = df["expensive"]

for col in X.select_dtypes(include=['object']).columns:
    le = LabelEncoder()
    X[col] = le.fit_transform(X[col].astype(str))

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

model = RandomForestClassifier(n_estimators=300, random_state=42)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))

importances = pd.DataFrame({
    "feature": X.columns,
    "importance": model.feature_importances_
}).sort_values("importance", ascending=False)

print("\nTop 10 Feature Importances:\n", importances.head(10))

plt.figure(figsize=(8,5))
plt.barh(importances["feature"].head(10)[::-1], importances["importance"].head(10)[::-1])
plt.xlabel("Importance")
plt.title("Top 10 Feature Importances (Random Forest)")
plt.show()

explainer = lime.lime_tabular.LimeTabularExplainer(
    training_data=X_train.values,
    feature_names=X_train.columns,
    class_names=["cheap", "expensive"],
    mode="classification"
)

i = 5
exp = explainer.explain_instance(
    X_test.iloc[i].values,
    model.predict_proba,
    num_features=5
```

```
)

print("\nLIME explanation for instance:", i)
print(exp.as_list())

fig = exp.as_pyplot_figure()
plt.title("LIME Explanation for House Price Classification")
plt.show()
```

```
Path to dataset files: /kaggle/input/housing-prices-dataset
Using CSV file: /kaggle/input/housing-prices-dataset/Housing.csv
Dataset shape: (545, 13)
      price  area  bedrooms  bathrooms  stories mainroad guestroom basement  \
0  13300000  7420         4          2        3      yes        no       no
1  12250000  8960         4          4        4      yes        no       no
2  12250000  9960         3          2        2      yes        no      yes
3  12215000  7500         4          2        2      yes        no      yes
4  11410000  7420         4          1        2      yes       yes      yes

  hotwaterheating airconditioning  parking prefarea furnishingstatus
0              no             yes        2      yes        furnished
1              no             yes        3       no        furnished
2              no              no        2      yes   semi-furnished
3              no             yes        3      yes        furnished
4              no             yes        2       no        furnished
Detected price column: price
Accuracy: 0.7981651376146789

Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.80      0.80        55
           1       0.80      0.80      0.80        54

    accuracy                           0.80       109
   macro avg       0.80      0.80      0.80       109
weighted avg       0.80      0.80      0.80       109
```

```
Top 10 Feature Importances:
             feature  importance
0               area    0.349385
1           bedrooms    0.097042
8    airconditioning    0.093366
3            stories    0.077934
11  furnishingstatus    0.068284
2          bathrooms    0.065647
9            parking    0.065340
10          prefarea    0.059265
6           basement    0.045495
4           mainroad    0.036691
```


Top 10 Feature Importances (Random Forest)
area
bedrooms
airconditioning
stories
furnishingstatus
bathrooms
parking
prefarea
basement
mainroad
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
Importance


LIME Explanation for House Price Classification
area > 6305.25
prefarea <= 0.00
airconditioning <= 0.00
bathrooms <= 1.00
2.00 < bedrooms <= 3.00
−0.15
−0.10
−0.05
0.00
0.05
0.10
0.15
0.20