import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.linear\_model import LinearRegression, Ridge, Lasso

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import mean\_squared\_error, r2\_score

df = pd.read\_csv("/content/Housing.csv")

categorical\_cols = ['mainroad', 'guestroom', 'basement', 'hotwaterheating',

'airconditioning', 'prefarea', 'furnishingstatus']

df\_encoded = pd.get\_dummies(df, columns=categorical\_cols, drop\_first=True)

X = df\_encoded.drop('price', axis=1)

y = df\_encoded['price']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2,

random\_state=42)

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

models = {

"Linear Regression": LinearRegression(),

"Ridge Regression": Ridge(alpha=1.0),

"Lasso Regression": Lasso(alpha=0.1),

"Random Forest": RandomForestRegressor(n\_estimators=100,

random\_state=42)

}

for name, model in models.items():

if 'Forest' in name:

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

else:

model.fit(X\_train\_scaled, y\_train)

y\_pred = model.predict(X\_test\_scaled)

rmse = np.sqrt(mean\_squared\_error(y\_test, y\_pred))

r2 = r2\_score(y\_test, y\_pred)

print(f"\n{name}")

print(f"RMSE: {rmse:.2f}")

print(f"R² Score: {r2:.4f}")

rf\_model = models["Random Forest"]

importances = rf\_model.feature\_importances\_

indices = np.argsort(importances)[-10:]

plt.figure(figsize=(10, 6))

plt.title("Top 10 Feature Importances (Random Forest)")

plt.barh(range(len(indices)), importances[indices], color="skyblue")

plt.yticks(range(len(indices)), [X.columns[i] for i in indices])

plt.xlabel("Relative Importance")

plt.tight\_layout()

plt.show()