```
import pandas as pd

# Load the dataset
data = pd.read_csv("Dataset.csv")  # Replace with the actual file path

# Display first few rows of the dataset to understand its structure
print(data.head())
```

```
   Age          Eduacation      Race         Hisp MaritalStatus  Nodeg  \
0   45  LessThanHighSchool  NotBlack  NotHispanic       Married      1
1   21        Intermediate  NotBlack  NotHispanic    NotMarried      0
2   38          HighSchool  NotBlack  NotHispanic       Married      0
3   48  LessThanHighSchool  NotBlack  NotHispanic       Married      1
4   18  LessThanHighSchool  NotBlack  NotHispanic       Married      1

   Earnings_1974  Earnings_1975  Earnings_1978
0      21516.670      25243.550      25564.670
1       3175.971       5852.565      13496.080
2      23039.020      25130.760      25564.670
3      24994.370      25243.550      25564.670
4       1669.295      10727.610       9860.869
```

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import LabelEncoder

# Load the dataset
data = pd.read_csv("Dataset.csv")  # Replace with the actual file path

# Convert categorical variables to numeric
data['Race'] = data['Race'].apply(lambda x: 1 if x == 'black' else 0)
data['Hisp'] = data['Hisp'].apply(lambda x: 1 if x == 'yes' else 0)
data['MaritalStatus'] = data['MaritalStatus'].apply(lambda x: 1 if x == 'yes' else 0)

# Convert 'Eduacation' column to numerical using Label Encoding
le = LabelEncoder()
data['Eduacation'] = le.fit_transform(data['Eduacation'])

# Select features and target variable
X = data[['Age', 'Race', 'Eduacation', 'Hisp', 'MaritalStatus', 'Earnings_1974', 'Earnings_1975']]
y = data['Earnings_1978']

# Splitting data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

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

# Make predictions on the test set
y_pred = model.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("Mean Squared Error:", mse)
print("R-squared Score:", r2)

# Output feature coefficients
coefficients = pd.DataFrame(model.coef_, X.columns, columns=['Coefficient'])
print(coefficients)
```

```
Mean Squared Error: 48605603.329378024
R-squared Score: 0.4769305724023293
                Coefficient
Age           -1.094257e+02
Race          -9.030372e+02
Eduacation     1.996069e+02
Hisp           1.833200e-12
MaritalStatus  0.000000e+00
Earnings_1974  2.854266e-01
Earnings_1975  4.807440e-01
```