Business Problem

"The labor problem" is the economics term widely used toward the turn of the twentieth century with various applications.It has been defined in many ways, such as "the problem of improving the conditions of employment of the wage-earning classes." It encompasses the difficulties faced by wage-earners and employers who began to cut wages for various reasons including increased technology, desire for lower costs or to stay in business.

Now the laor organization, https://labour.gov.in/lcandilasdivision/india-ilo the indian version of this recently published the data for the 1974 and 1975 and wanted to demo of data science such that the data of 1978 labor can be predicted. Now the problem is the data is a copy of USA version and have race in it.

Data Contains Age, Race, Educational detail and Labour earning for 1974, 1975. The problem we are solving is the prediction of the future labours earning. The earning can be dependant on many of the variables. We have data for following

Age of the person.

Race : Is he/she is black or not black.

Education Details : How qualified the person is?

Hispanic : Is that person is Hispanic or not?

Married : Does marriage affect the earnings. And other informations.

```
# Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import OneHotEncoder
from sklearn.metrics import mean_squared_error
import numpy as np
```

```
# Load the dataset
data = pd.read_csv('TheLaborProblem.csv')
```

```
# Check for any missing values
print(data.isnull().sum())
```

```
Age              0
Eduacation       0
Race             0
Hisp             0
MaritalStatus    0
Nodeg            0
Earnings_1974    0
Earnings_1975    0
Earnings_1978    0
dtype: int64
```

```
# Define categorical columns
categorical_cols = ['Eduacation', 'Race', 'Hisp', 'MaritalStatus']
```

```
# Apply OneHotEncoder
data = pd.get_dummies(data, columns=categorical_cols, drop_first=True)
```

```
# Define feature columns (X) and target column (y)
X = data.drop(['Earnings_1978'], axis=1)
y = data['Earnings_1978']
```

```
# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Solution Approach:

As we need to predict Labour earning for 1978 which is continuous in nature, Linear Regression can be used for prediction

```
# Initialize and fit the multiple linear regression model
model = LinearRegression()
model.fit(X_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
# Predict on the test set
y_pred = model.predict(X_test)
```

```
# Evaluate the model using Mean Squared Error
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
print(f'Root Mean Squared Error: {rmse}')
```

```
Root Mean Squared Error: 6973.217621811093
```

```
# Print coefficients to understand the impact of each feature
print("Coefficients: ", model.coef_)
print("Intercept: ", model.intercept_)
```

```
Coefficients:  [-1.07350492e+02 -2.93390186e+01  2.83039030e-01  4.75134632e-01
  2.74149983e+02 -2.93390186e+01  1.05909511e+03  1.03313773e+03
 -8.57181352e+02 -4.42428781e+02 -8.63504844e+01]
Intercept:  7918.751772334645
```