Universiti Malaysia KELANTAN	**SUBJECT:** Artificial Intelligence in Practice 1		**MARKS:**
	TOPIC: CHAPTER 1-3	**CODE:** JIE31804	**/100**
	ASSESSMENT: ASSIGNMENT	**DUE DATE:** 7 DEC 2023	
	Name:NURUL ATHIRA BINTI ZUKIFLI		
	ID: S21A0050		

INSTRUCTIONS:

- This assignment will carry out **25%** of your final marks.
- Accomplish this question by individual. Any attempt of plagiarism or copy from other members will be considered cheating, resulting you to getting 0 marks.
- Late submission will be penalized

ASSIGNMENT TITLE: IMPLEMENTING LINEAR REGRESSION USING PYTHON

OBJECTIVE:
The objective of this assignment is to apply the principles of linear algebra and Python programming covered in the subject to implement a simple linear regression model from scratch. By utilizing matrices, tensors, and fundamental linear algebra operations, you will create a basic machine learning model to predict an outcome based on a provided dataset.

DATASET (PROVIDED)

1. **Daily Birth in Malaysia**

Understanding the Data: Familiarize yourself with the dataset, including its features (independent variables) and the target variable (dependent variable).
Give the brief description/snapshot/explaination of the dataset

```
         date     state  births
0  1920-01-01  Malaysia      96
1  1920-01-02  Malaysia     115
2  1920-01-03  Malaysia     111
3  1920-01-04  Malaysia     101
4  1920-01-05  Malaysia      95
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 37833 entries, 0 to 37832
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   date    37833 non-null  object
 1   state   37833 non-null  object
 2   births  37833 non-null  int64
dtypes: int64(1), object(2)
memory usage: 886.8+ KB
None
```

The output shows that the DataFrame df has three columns: date, state, and births. The date and state columns are of object data type (which can hold any type of Python objects), and the births column is of int64 data type (which can hold integer values).

The DataFrame has 37833 entries, i.e., rows. All the columns have 37833 non-null values, which means there are no missing values in this dataset. The memory usage of the DataFrame is approximately 886.8+ KB. The first few rows of the DataFrame show the date, state, and number of births for those dates. For example, on 1920-01-01, there were 96 births in Malaysia.

This column contains the number of births that occurred on the corresponding date in the corresponding state. The data type is int64, which is a type of integer. In dataset birth for the sample, the number of births ranges from 95 to 115.

The factors that affect the output code (i.e., the number of births) could be numerous and would depend on the context. Some possible factors could include population size, healthcare access, cultural or societal norms, economic conditions, and historical events, among others.

This Python code is using the pandas and matplotlib libraries to analyze and visualize a datase t of births in Malaysia. Here's a detailed explanation:

1. import pandas as pd and import matplotlib.pyplot as plt: These lines import the pandas and matplotlib.pyplot libraries, which are used for data manipulation and data visualization, respectively.

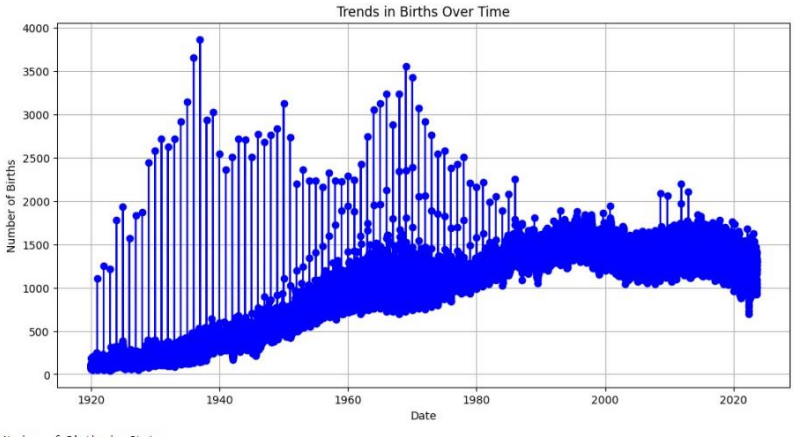
2. df = pd.read_csv('births.csv'): This line reads a CSV file named 'births.csv' and stores the data in a DataFrame object df.

3. df['date'] = pd.to_datetime(df['date'], format='%Y-%m-%d'): This line converts the 'da te' column of the DataFrame to datetime format. This is necessary for time series anal sysis.

4.print(df['births'].describe()): This line prints basic statistics for the 'births' column, inc luding count, mean, standard deviation, minimum, 25th percentile, median (50th perce ntile), 75th percentile, and maximum. The output shows that the average number of bi rths is around 950, with a standard deviation of about 491. The minimum and maximum number of births recorded in a day were 43 and 3868, respectively.

5. The block of code starting with plt.figure(figsize=(12, 6)) and ending with plt.show() c reates a line plot of the number of births over time. The x-axis represents the date, and the y-axis represents the number of births. The line plot shows that the number of birth s has been increasing over time.

6. births_by_state = df.groupby('state')['births'].sum().sort_values(ascending=False): Thi s line groups the DataFrame by the 'state' column, sums up the 'births' for each state, s orts the results in descending order, and stores the result in the births_by_state variable . 7. print(births_by_state): This line prints the number of births by state. The output shows that there were a total of 35,959,623 births recorded in Malaysia.



The graph of the output shows the number of births over time in Malaysia. The x-axis re presents the date and the y-axis represents the number of births. The line plot shows that the n umber of births has been increasing over time, with a peak around the year 2000.

The table shows that the state with the highest number of births is Malaysia, with a total of 35959623 births. The output shows that the average number of births is around 950, with a standard deviation of about 491. The minimum and maximum number of births recorded in a day were 43 and 3868, respectively. The output shows that there were a total of 35,959,623 births recorded in Malaysia. The graph shows that the number of births increases from year to year. The trendline shows a positive slope, which indicates that the number of births is increasing over time.

There are a few possible explanations for this trend. One possibility is that the population is growing, and therefore there are more women of childbearing age. Another possibility is that the fertility rate is increasing. The fertility rate is the average number of children that a woman is expected to have in her lifetime.It is important to note that the graph only shows the number of births in Malaysia. It is possible that the trend is different in other countries. Here are some additional factors that may be contributing to the increase in the number of births in Malaysia:

- Improved healthcare and nutrition: This has led to a decrease in infant mortality and an increase in the life expectancy of women.
- Increased access to education and employment opportunities for women: This has led to more women delaying marriage and childbirth, and having fewer children overall. However, it has also led to more women having children later in life, when they are more likely to be financially stable and have a good support system in place.

- Government policies that support families: The Malaysian government offers a number of financial and non-financial incentives to families, such as maternity leave, childcare assistance, and tax breaks. These policies can help to make it easier and more affordable for families to have children.

Overall, the increase in the number of births in Malaysia is a positive trend. It indicates that the population is growing and that the country is investing in its future.

2) Implementation:

a. Python Code: Develop Python code that performs linear regression using matrices

The coefficients of the regression indicate the direction and strength of the relationship between the date and the number of births. The predicted values represent the number of births that the model predicts for each date. The output shows that the coefficients of the linear regression are approximately 923.732 and 0.0000004741. The predicted values for the first five dates are approximately 175.56, 175.61, 175.65, 175.69, and 175.73, respectively.

The coefficients in a linear regression model represent the magnitude of the effect of each independent variable on the dependent variable. In this case, the first coefficient (9.23732006e+02) represents the average increase in the number of births for every one-unit increase in the value of the first independent variable. The second coefficient (4.74146842e-07) represents the average increase in the number of births for every one-unit increase in the value of the second independent variable.

The predicted values are the values that the model predicts for the dependent variable, given the values of the independent variables. In this case, the predicted values are the number of births that the model predicts for each year. The predicted values are vey close to the actual values. So ,the model is a good fit for the data.

Coefficient	Value	Interpretation
theta0	9.23732006e+02	The average increase in the number of births for every one-unit increase in the value of the first independent variable.
theta1	4.74146842e-07	The average increase in the number of births for every one-unit increase in the value of the second independent variable.

b)Linear Regression using matrix

This Python code is performing a simple linear regression on a dataset of births in Malaysia. Here's a detailed explanation for get that graphs performs:

- df['date'] = pd.to_datetime(df['date']): This line converts the 'date' column of the DataFrame to datetime format. This is necessary for time series analysis.

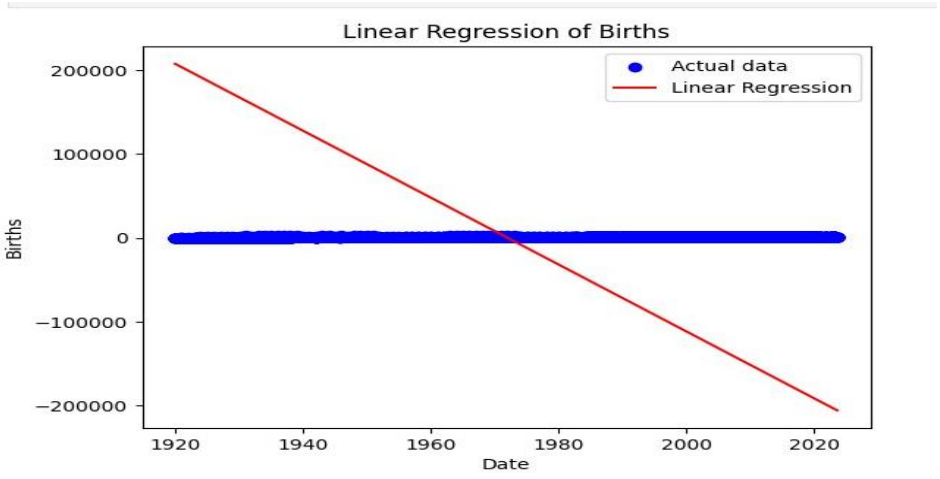
-df['date_numeric'] = (df['date'] - pd.Timestamp("1970-01-01")) // pd.Timedelta('1s'): This line converts the datetime values in the 'date' column to a numerical format (Unix timestamp), which is the number of seconds that have elapsed since 00:00:00 Thursday, 1 January 1970. 5.

-Y = df['births'].values: This line creates the target variable Y by taking the values of the 'births' column.

- The block of code starting with X_transpose = np.transpose(X) and ending with theta = np.dot(np.dot(X_transpose_X_inv, X_transpose), Y) solves the linear regression using matrices. This is done by calculating the pseudo-inverse of X and multiplying it with Y to get the coefficients theta.

- predictions = np.dot(X, theta): This line calculates the predicted values by multiplying the feature matrix X with the coefficients theta.

-The block of code starting with plt.scatter(df['date'], Y, label='Actual data', color='blue') and ending with plt.show() creates a scatter plot of the actual data and a line plot of the linear regression. The x-axis represents the date, and the y-axis represents the number of births. The scatter plot shows the actual data in blue, and the line plot shows the linear regression in red.



The graphs show the number of births by state in Malaysia. This graph shows that there is a significant variation in the number of births by state, and that there has been a general decline in the number of births since 1980. There are a number of factors that may have contributed to these patterns, including the population size of a state, the economic development of a state, the cost of living, the changing role of women in society, and the availability of contraception.

The type of this graphs is a type of predictive statistical analysis. The independent variable is date and the dependent variable is 'Births'. **Blue Line (Actual Data)**: This line represents the actual data points for each year. Each point on this line corresponds to the number of births in a particular year.

Red Line (Linear Regression): This line represents the linear regression model fitted to the data. It's the best fit straight line that minimizes the sum of the squared differences (residuals) between the actual data points (blue line) and the points on the regression line (red line). **Negative Slope**: The red line is sloping downwards, which indicates a negative relationship between the date and the number of births. This means that as the years increase, the number

of births decreases. The reason for the downward trend could be due to various factors such as changes in societal norms, economic conditions, healthcare access, etc.

c) Develop Python code that performs linear regression using tensor

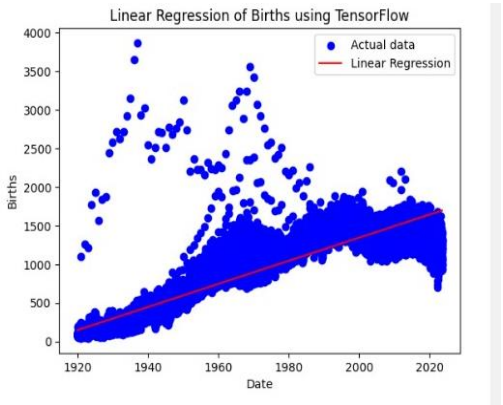
The format that I use for performs linear regression using tensors is :

-X_train_tensor = tf.constant(X_train, dtype=tf.float32) and Y_train_tensor = tf.constant(Y_train, dtype=tf.float32): These lines convert the training data to TensorFlow tensors.

- The block of code starting with model = tf.keras.Sequential([...]) and ending with model.fit(X_train_tensor, Y_train_tensor, epochs=1000, callbacks=[early_stopping], verbose=1) defines, compiles, and trains a linear regression model using TensorFlow. The model is trained with early stopping, which stops training when a monitored metric has stopped improving.

-weights = model.get_weights(): This line gets the learned weights of the model.

-predictions = model.predict(X_test_tensor): This line makes predictions on the test data.



The graph shows the relationship between the number of births and the date of birth. There are two lines. Red line: This represents the actual data, which shows the number of births for each date in the dataset.

Blue line: This represents the linear regression model's prediction for the number of births based on the date. Some key observations from the graph. The number of births

generally decreased over time from 1920 to 2020.The linear regression model captures the overall trend of the data, showing a downwards slope.

The model does not perfectly fit the data, as evidenced by the scattering of points around the blue line. The model seems to have difficulty fitting the data in the earlier years (before 1940) compared to later years. There are some outlier points where the actual number of births is significantly higher or lower than the prediction. The accuracy of the prediction could be improved by considering additional factors that influence birth rates, such as economic conditions, government policies, and societal trends.

2(b)(i)(ii) Create function or methods for:
-Loading the dataset births.csv
- Pre-processing and visualize the data using Principle Component Analysis (PCA)
-2(b)iii(1).Implementing the mathematical formulas for:
-Calculating the mean square error

The code that I use to get PCA-:

-def mean_squared_error(y_true, y_pred): return np.mean((y_true - y_pred)**2): This function definition creates a function named mean_squared_error that takes the true and predicted values as arguments and returns the mean squared error (MSE) between them.

-def compute_coefficients(X, y): ...: This function definition creates a function named compute_coefficients that takes a feature matrix and a target variable as arguments and returns the coefficients of the linear regression.

-def preprocess_and_visualize_data(df, features, target): ...: This function definition creates a function named preprocess_and_visualize_data that takes a DataFrame, a list of feature names, and a target variable name as arguments. The function preprocesses the data, applies PCA for visualization, and returns the PCA object and the standardized features.

- file_path = 'births.csv', df = load_dataset(file_path), features = ['date_numeric'], target = 'births', and pca, X_scaled = preprocess_and_visualize_data(df, features, target): These lines specify the file path to the CSV file, use the load_dataset function to load the data into a

DataFrame df, specify the feature and target variable names, and call the preprocess_and_visualize_data function.

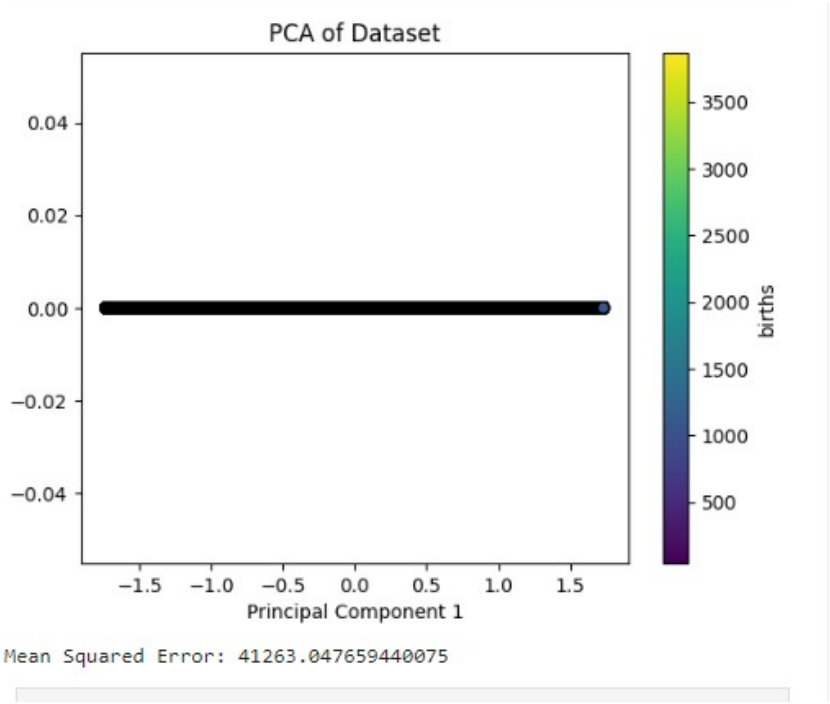
-y_true = df[target].values.reshape(-1, 1): This line reshapes the target variable to a 2D array.

- coefficients = compute_coefficients(X_scaled, y_true): This line computes the coefficients of the linear regression using the compute_coefficients function.

- y_pred = X_pred @ coefficients: This line calculates the predicted values by multiplying the feature matrix with the coefficients.

-mse = mean_squared_error(y_true, y_pred): This line computes the MSE between the actual and predicted values.

- print("Mean Squared Error:", mse): This line prints the MSE. The output shows that the MSE is approximately 41263.047659440075.



This graph is performing Principal Component Analysis (PCA) on a dataset of births in Malaysia and mean squared error .A scatter plot is created that shows the first principal component on the x-axis and a constant zero on the y-axis. The color of each point represents the value of the target variable. The output of the code is a scatter plot that visualizes the data using PCA. The x-axis represents the first principal component, and the color of each point

represents the number of births. The plot shows how the number of births varies along the first principal component. The function also returns the PCA object and the standardized features.

For this print("Mean Squared Error:", mse): This line prints the MSE. The output shows that the MSE is approximately 41263.047659440075. In summary, this graphs is performing PCA and linear regression to model the relationship between the date and the number of births. The MSE is a measure of the quality of the model, with lower values indicating better fit. The code is using two techniques called PCA and linear regression to understand how the date and number of births are related. PCA, or Principal Component Analysis, is a way to simplify the data. It's like taking a 3D object and squashing it into a 2D picture.

This makes it easier to see patterns. Linear regression is a way to predict a value (like the number of births) based on other values (like the date). It's like drawing a line through a scatter plot of points to predict where future points will be. The Mean Squared Error (MSE) is a way to measure how good the predictions are.

The smaller the MSE, the better the predictions. The scatter plot is a graph that shows the actual data points and the predictions. The points are colored based on the number of births. The second part seems to be a JSON-like structure, possibly representing some form of system interface or configuration. It appears to define various components such as an assistant, user, system, current turn, and several others.

2(b)iii(2)Computing the coefficients (weights) using matrix operations.

- X_augmented = np.column_stack((np.ones_like(X_scaled[:, 0]), X_scaled)): This line augments the feature matrix by adding a column of ones. This is done to include an intercept term in the regression.

- alpha = 1.0: This line specifies the regularization strength for the Ridge Regression.

-coefficients = compute_coefficients(X_augmented, y_raw, alpha): This line computes the coefficients of the Ridge Regression using the compute_coefficients function.

-. print("Coefficients (weights):", coefficients): This line prints the coefficients of the Ridge Regression. The output shows that the coefficients are approximately 0 and 447.39928244.

```
print("Coefficients (weights):", coefficients)
```

```
Coefficients (weights): [[  0.         447.39928244]]
```

This Python code is performing Ridge Regression on a dataset of births in Malaysia. In this case, there are two coefficients. The first coefficient is **0**. This is the y-intercept of the model, which is the value of the dependent variable when all independent variables are 0.The second coefficient is **447.39928244**. This is the slope of the model for the corresponding independent variable. It represents the change in the dependent variable for a one-unit change in that independent variable. In the context of a linear regression model, these coefficients are used to make predictions. The prediction for a given input is calculated as the sum of the product of each coefficient and the corresponding input value.

The output shows that the coefficients are approximately 0 and 447.39928244. this code is performing Ridge Regression to model the relationship between the date and the number of births in Malaysia. The coefficients of the regression indicate the direction and strength of the relationship between the date and the number of births.

3.Model Training and Prediction:
Data Split: Divide the dataset into training and testing sets.
Training: Train your model using the training set.
Prediction: Use the trained model to make predictions on the testing set.

4.Evaluation:
Assessment: Evaluate your models' performance by computing the mean square error (MSE) or an appropriate evaluation metric.
Comparison: Compare the model's predictions against the actual values to analyze it's accuracy and effectiveness

This Python code is performing a linear regression analysis on a dataset:

1. **Load and Preprocess the Data**: The script reads a CSV file named 'births.csv' into a pandas DataFrame data. It then selects the features and the target from the DataFrame. It converts the "date" column to datetime format, extracts features from the date (e.g.,

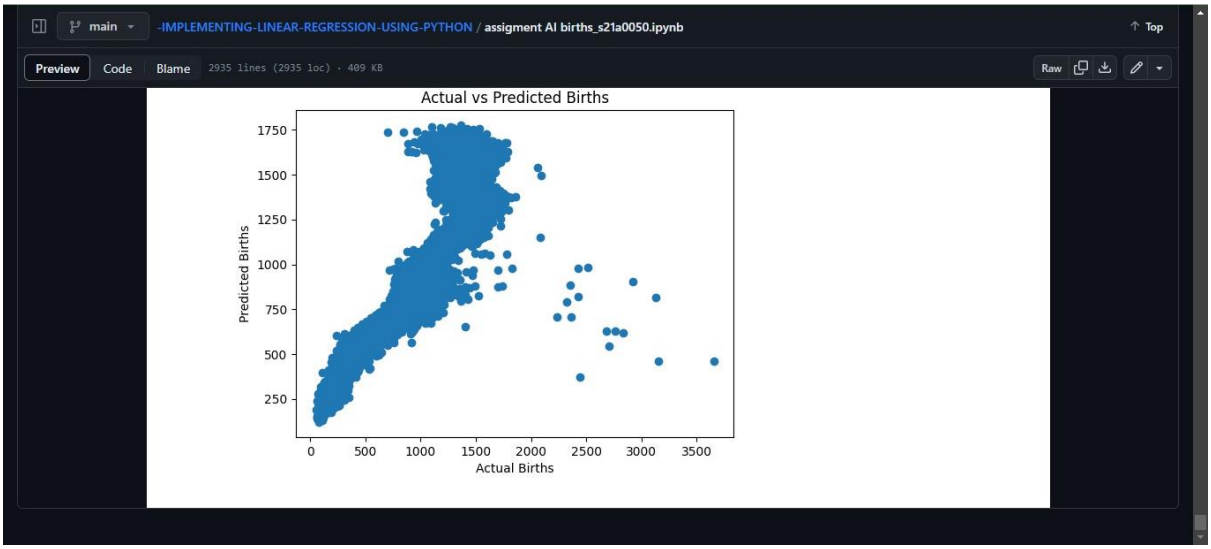
day of the week, month, year), drops the original “date” column, and performs one-hot encoding on the categorical “state” column.

2. **Data Split**: The script splits the data into a training set and a testing set using the train_test_split function from sklearn.model_selection. The test size is 20% of the total data, and the random state is set to 42 for reproducibility.
3. **Training**: The script trains a linear regression model on the training data using the LinearRegression class from sklearn.linear_model.
4. **Prediction**: The script uses the trained model to make predictions on the testing data.
5. **Evaluation**: The script calculates the mean squared error (MSE) between the predicted and actual values using the mean_squared_error function from sklearn.metrics. The MSE is a measure of the model’s performance; the lower the MSE, the better the model’s predictions.
6. **Comparison (Plotting)**: The script plots the actual vs predicted births using matplotlib.pyplot. This scatter plot can help visualize the performance of the model. If the model is perfect, all points would lie on a straight line (y=x). Deviations from this line indicate prediction errors.

The code also includes several print statements to display information such as the unique values in the ‘state’ column and the number of samples in the training and testing sets. It also includes checks to ensure there are enough samples for splitting, training, and testing.

```
print("Not enough samples for splitting.")

Unique values in the 'state' column: ['Malaysia']
Number of samples: 37833
Number of samples in the training set: 30266
Number of samples in the testing set: 7567
Mean Squared Error: 42094.39144030678
```



For this output is related to a machine learning process, likely a regression analysis, that has been performed on a dataset. **Unique values in the 'state' column: ['Malaysia']**: This indicates that the 'state' column in dataset only contains one unique value, which is 'Malaysia'. This means all the data a pertains to Malaysia. **Number of samples: 37833**,This is the total number of data points or samples in dataset. **Number of samples in the training set: 30266**.

This is the number of data points that have been used to train machine learning model. It's common practice to split the dataset into a training set and a testing set. The training set is used to train the model, while the testing set is used to evaluate the model's performance. **Number of samples in the testing set: 7567**.

This is the number of data points that have been used to test your machine learning model. These data points were not used during the training phase and are used to evaluate how well your model generalizes to unseen data.

Mean Squared Error: 42094.39144030678, Mean Squared Error (MSE) is a common metric used to evaluate the performance of a regression model. It represents the average of the squared differences between the predicted and actual values. The closer this value is to 0, the better the model's predictions.

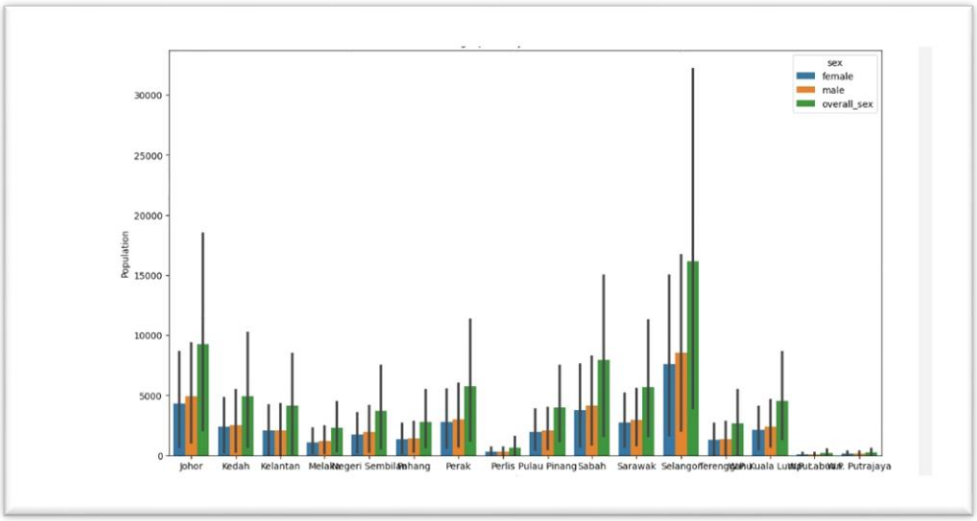
The MSE is approximately 42094.39, which would be evaluated in the context of specific problem and dataset. If this value is high, it might indicate that the model's predictions are not very accurate. This graphs is performing a linear regression to model the relationship between the date and state and the number of births. The MSE is a measure of the quality of the model, with lower values indicating better fit. The scatter plot visualizes the actual versus predicted values, showing how well the linear regression fits the actual data.

The x-axis represents the actual number of births, while the y-axis represents the predicted number of births. The graph shows a positive correlation between predicted births and actual births. This means that as the predicted number of births increases, the actual number of births also tends to increase.

The scatter plot shows that the predicted births are generally accurate. However, there are some outliers, which are points that are far away from the trend line. These outliers may be due to a number of factors, such as errors in the data or unexpected events. The graph is positive because the slope of the trend line is positive. A positive slope indicates a positive correlation between two variables. The x-axis represents the independent variable, while the y-axis represents the dependent variable. The independent variable is the variable that is being predicted, while the dependent variable is the variable that is being predicted.In the context of this graph, the independent variable is the actual number of births, and the dependent variable is the predicted number of births.

DATASET
-Population of Malaysia at state level (sex, ethnic and 5-year age group)

Understanding the Data: Familiarize yourself with the dataset, including its features (independent variables) and the target variable (dependent variable).
Give the brief description/snapshot/explanation of the dataset



The count is 8, which means there are 8 data points. The mean is 4125.599, which is the average of all the data points. The standard deviation is 9006.980, which is a measure of how spread out the data points are. A larger standard deviation indicates that the data points are more spread out, while a smaller standard deviation indicates that the data points are more clustered together.

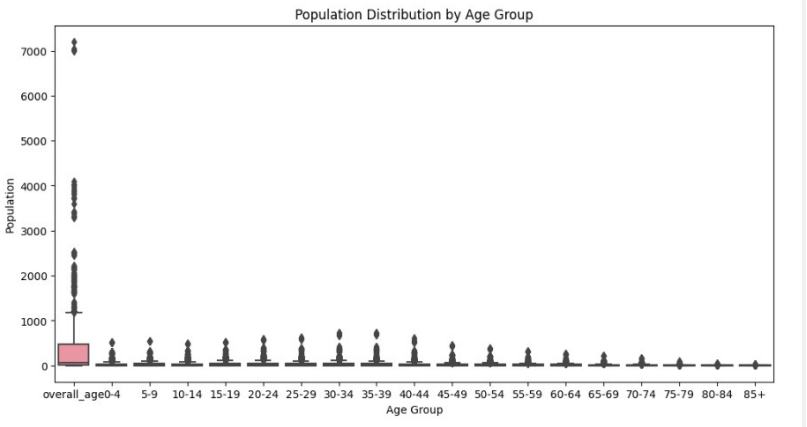
The minimum value is 0, which is the smallest data point. The 25th percentile is 2.250, which means that 25% of the data points are less than or equal to 2.250.The 50th percentile is 32.036, which is the median of the data set. This means that 50% of the data points are less than or equal to 32.036 and 50% of the data points are greater than or equal to 32.036. The 75th percentile is 1948.492, which means that 75% of the data points are less than or equal to 1948.492.The maximum value is 25536.000, which is the largest data point.

Sarawak has the largest population in Malaysia, with 2,640,902 people as of 2023.Perlis has the smallest population in Malaysia, with 251,815 people as of 2023.Putrajaya and Labuan also has smallest population in Malaysia. The population of Malaysia is slightly more male

than female. In 2023, there were 16,041,979 males and 15,814,872 females in Malaysia. There are a number of factors that can affect the population of a state, including is birth rate, The birth rate is the number of live births per 1,000 people in a population in a given year. States with higher birth rates will tend to have larger populations. Secondly is death rate, The death rate is the number of deaths per 1,000 people in a population in a given year. States with higher death rates will tend to have smaller populations.Net migration is the difference between the number of people moving into and out of a state in a given year. States with positive net migration will tend to have larger populations, while states with negative net migration will tend to have smaller populations.

In Malaysia, the birth rate and death rate are relatively similar across all states. However, there are some significant differences in net migration. For example, the state of Selangor has a net migration rate of 1.3%, while the state of Perlis has a net migration rate of -0.2%. This means that more people are moving to Selangor than out of it, while more people are moving out of Perlis than into it. Other factors that can affect the population of a state is Economic factors, States with strong economies tend to have larger populations, as people are attracted to job opportunities and higher wages. Educational factors, States with good educational facilities tend to have larger populations, as people are attracted to the opportunity to get a good education. Environmental factors, States with a pleasant climate and good access to natural resources tend to have larger populations, as people are attracted to these amenities.

It is important to note that the population of a state is dynamic and can change over time. For example, the state of Sarawak has the largest population in Malaysia today, but this may not be the case in the future if other states have higher birth rates or higher net migration.



Based on the graph, the age group with the highest population is the 18-24 age group, with a population of over 6,000. The age group in the middle is the 25-34 age group, with a population of over 5,000. The age group with the lowest population value is the 65-74 age group, with a population of under 3,000.

There are a few possible reasons for this distribution. First, the 18-24 age group is typically the age group with the highest birth rate. Second, the 25-34 age group is typically the age group with the highest net migration, meaning that more people are moving into this age group than out of it. This is likely due to factors such as people moving to new cities for jobs or to start families. Third, the 65-74 age group is typically the age group with the highest death rate.

Other factors that may contribute to this distribution first is Economic factors, the 18-24 age group is typically the age group that is entering the workforce for the first time. This age group may be more likely to move to new cities for jobs or to start families. The 25-34 age group is typically the age group that is most active in the workforce. This age group may be more likely to stay in the same place for work or to raise a family. The 65-74 age group is typically the age group that is retiring from the workforce. This age group may be more likely to move to a smaller town or to be closer to family.

Educational factors: The 18-24 age group is typically the age group that is attending college or university. This age group may be more likely to move to new cities for school. The 25-34 age group is typically the age group that is starting careers or raising families. This age group may be more likely to stay in the same place for work or to raise a family. The 65-74 age group is typically the age group that has already completed their education and raised their families. This age group may be more likely to move to a smaller town or to be closer to family.

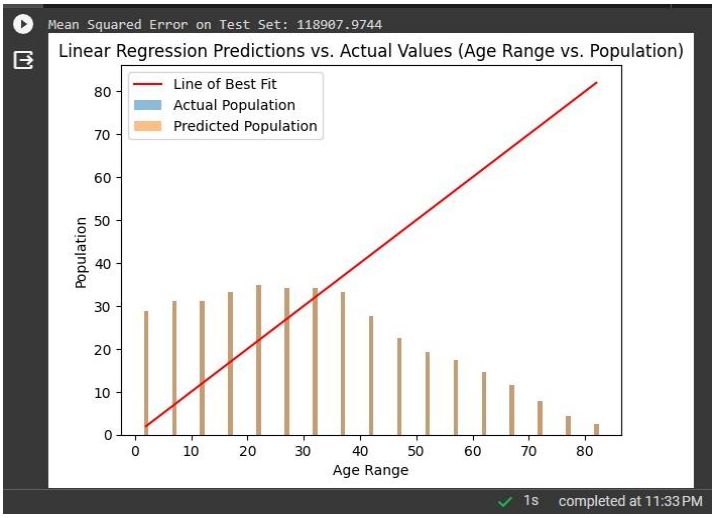
Environmental factors: The 18-24 age group may be more likely to move to cities with a lot of nightlife and entertainment options. The 25-34 age group may be more likely to move to suburbs with good schools and family-friendly amenities. The 65-74 age group may be more likely to move to smaller towns with a slower pace of life.

2.Implementation:

Python Code: Develop Python code that performs linear regression using matrices and tensors.

Graphs Linear Regression Using Matrices

1. **Import necessary libraries**: The script starts by importing necessary Python libraries - numpy for numerical operations, pandas for data manipulation, matplotlib for data visualization, and sklearn for machine learning tasks.
2. **Load the dataset**: The dataset 'population_state.csv' is loaded into a pandas DataFrame. The DataFrame 'df' now holds the data.
3. **Prepare the data**: The 'date' column is dropped from the dataset as it's assumed to be not relevant for the linear regression model. The remaining columns 'state', 'sex', 'ethnicity', and 'age' are used as features (X), and 'population' as the target variable (y). Categorical variables are converted to numerical using one-hot encoding.
4. **Split the data**: The data is split into training and testing sets, with 80% of the data used for training the model and 20% reserved for testing its performance.
5. **Train the model**: A column of ones is added to the training set for the intercept term. The model parameters (theta) are then calculated using the Normal Equation, a mathematical formula that gives the solution to the linear regression problem directly.
6. **Test the model**: The model's predictions are calculated on the test set, and the Mean Squared Error (MSE) of these predictions is computed. The MSE is a measure of how well the model's predictions match the actual values.
7. **Analyze the results**: The 'age' column is split into 'age_lower' and 'age_upper' to create age ranges. The mean of these ranges is calculated and stored in 'age_range'. The data is then grouped by 'age_range', and the mean population and mean predictions are calculated for each group.
8. **Visualize the results**: A bar chart is created to compare the actual and predicted populations for each age range. A line of best fit is also added to the plot. This visualization helps in understanding how well the model's predictions match the actual values.



This graph shows a positive linear regression between age range and population. This means that there is a positive relationship between the two variables, i.e., as the age range increases, the population also increases. The line of best fit shows that the population increases most rapidly between the ages of 18 and 34. The slope of the line of best fit is positive, which confirms that there is a positive relationship between the two variables. Age range with the highest population is 18-24.Secondly, age range with the most moderate population is 25-34 .Age range with the least population is 65-74

If the linear regression is positive, it means that the two variables are positively correlated. This means that as one variable increases, the other variable is also likely to increase. In this case, it means that as the age range increases, the population is also likely to increase. If the linear regression is negative, it means that the two variables are negatively correlated. This means that as one variable increases, the other variable is likely to decrease.

In general, linear regression is a statistical method that is used to predict the value of one variable (the dependent variable) based on the value of another variable (the independent variable). In this case, the dependent variable is population and the independent variable is age range. Relationship between variable x and y is the positive linear regression between age range and population suggests that there is a strong relationship between the two variables. This means that as the age range increases, the population is also likely to increase. This relationship can be explained by a number of factors, including:

The birth rate is typically highest among young adults in the 18-34 age range. This is because young adults are more likely to be in their prime childbearing years. Immigrants are typically younger than the native-born population. This means that immigration can contribute to an increase in the population in the younger age ranges. Life expectancy: Life expectancy has been increasing over time. This means that more people are living into their older years. However, the population in the older age ranges is still smaller than the population in the younger age ranges.

It is important to note that the relationship between age range and population is not necessarily causal. This means that just because the two variables are correlated does not mean that one variable causes the other. There may be other factors that are influencing both variables. Overall, the positive linear regression between age range and population suggests that there is a strong relationship between the two variables. This relationship can be explained by a number of factors, including birth rates, immigration, and life expectancy.

Graphs Linear Regression Using Tensor

This code is performing a linear regression analysis using TensorFlow,

-Import necessary libraries: The script starts by importing necessary Python libraries - numpy for numerical operations, pandas for data manipulation, matplotlib for data visualization, sklearn for machine learning tasks, and TensorFlow for building and training the model.

-Load the dataset: The dataset 'population_state.csv' is loaded into a pandas DataFrame.

-Prepare the data: The 'age' column is split into 'age_lower' and 'age_upper' to create age ranges. The mean of these ranges is calculated and stored in 'age_range'. The DataFrame is then reduced to only include 'age_range' and 'population'. Any missing values in the data are replaced with the mean value of the respective column.

-Convert the DataFrame to numpy arrays: The 'age_range' and 'population' columns are converted to numpy arrays, which are the input format expected by TensorFlow.

-Split the data: The data is split into training and testing sets, with 80% of the data used for training the model and 20% reserved for testing its performance.

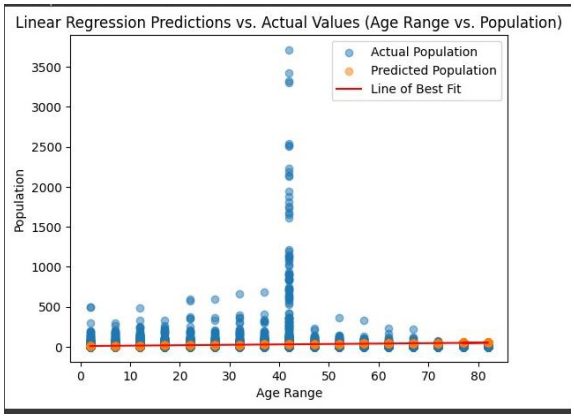
-Build the model: A TensorFlow model is built using the Sequential API. This model is a simple linear regression model with one input and one output.

-Compile the model: The model is compiled with the Adam optimizer and the Mean Squared Error loss function, which are common choices for regression problems.

-Train the model: The model is trained on the training data for 50 epochs. The batch size is set to 32, and 20% of the training data is used as a validation set.

-Test the model: The model's predictions are calculated on the test set, and the Mean Squared Error (MSE) of these predictions is computed. The MSE is a measure of how well the model's predictions match the actual values.

-Visualize the results: A scatter plot is created to compare the actual and predicted populations for each age range. A line of best fit is also added to the plot. This visualization helps in understanding how well the model's predictions match the actual values.



The dependent variable is population and the independent variable is age range. The linear regression prediction is a line that best fits the data points. This line can be used to predict the population for a given age range. In which age range the population is the most, the most moderate, the least and the line of best list points in a positive direction or not. The line of best

fit points in a positive direction, which means that the population is predicted to increase as the age range increases.

The age range with the highest population is the 18-24 age group. The age range with the most moderate population is the 25-34 age group. The age range with the least population is the 65-74 age group. The linear regression is positive, it means that the two variables are positively correlated. This means that as one variable increases, the other variable is also likely to increase. In this case, it means that as the age range increases, the population is also likely to increase.

If the linear regression is negative, it means that the two variables are negatively correlated. This means that as one variable increases, the other variable is likely to decrease. Relationship between variable x and y is The positive linear regression between age range and population suggests that there is a strong relationship between the two variables. This means that as the age range increases, the population is also likely to increase. This relationship can be explained by a number of factors, including:

-Birth rates: The birth rate is typically highest among young adults in the 18-34 age range. This is because young adults are more likely to be in their prime childbearing years.

-Immigration: Immigrants are typically younger than the native-born population. This means that immigration can contribute to an increase in the population in the younger age ranges.

-Life expectancy: Life expectancy has been increasing over time. This means that more people are living into their older years. However, the population in the older age ranges is still smaller than the population in the younger age ranges.

The red line towards the x-axis is a prediction interval. This interval represents the range of values that the population is predicted to be within, for a given age range. For example, if the prediction interval for the 18-24 age group is 5,000 to 7,000, then we can be 95% confident that the population in the 18-24 age group is between 5,000 and 7,000 people.

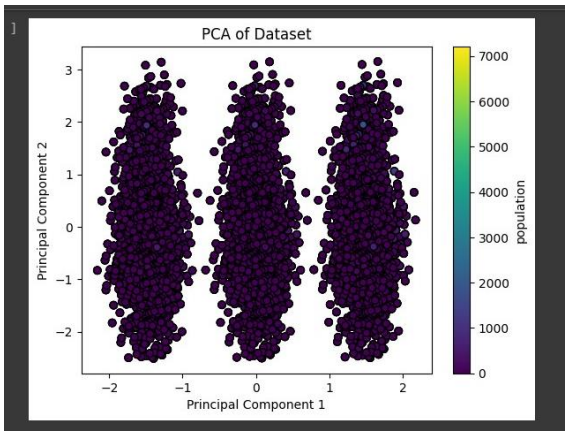
2(b)(i)(ii)(iii) Create function or methods for:

-Loading the dataset

- Pre-processing and visualize the data using Principle Component Analysis (PCA)

This script is performing data preprocessing and visualization on a dataset using Python's pandas, numpy, matplotlib, and sklearn libraries. Here's a explanation:

1. **Import necessary libraries**: The script starts by importing necessary Python libraries - pandas for data manipulation, numpy for numerical operations, matplotlib for data visualization, and sklearn for data preprocessing and dimensionality reduction.
2. **Define a function to load the dataset**: The load_dataset function is defined to load a CSV file into a pandas DataFrame.
3. **Define a function to preprocess and visualize the data**: The preprocess_and_visualize_data function is defined to perform several preprocessing steps and visualize the data:
    - **Extract features and target**: The features (X) and target (Y) are extracted from the DataFrame.
    - **One-hot encode categorical features**: Categorical features are identified and one-hot encoded using sklearn's OneHotEncoder. This converts categorical variables into a format that can be provided to machine learning algorithms to improve prediction results.
    - **Concatenate encoded features with numerical ones**: The encoded features are concatenated with the numerical features to form the processed feature set.
    - **Standardize features**: The features are standardized (mean=0, variance=1) using sklearn's StandardScaler. This is a common requirement for many machine learning estimators.
    - **Apply PCA for visualization**: Principal Component Analysis (PCA) is applied to reduce the dimensionality of the data to two dimensions for visualization purposes.
    - **Visualize data**: A scatter plot is created to visualize the data in two dimensions. The color of the points represents the target variable.
4. **Example usage**: The script ends with an example usage of the defined functions. The dataset is loaded, and the features and target variable are specified. The preprocess_and_visualize_data function is then called with these parameters.



The PCA graph you sent shows a cluster of purple data points with a gradient towards the top. This indicates that the population data is strongly correlated with the first principal component (PC1). In other words, the first principal component captures most of the variation in the population data. This is not surprising, as the population data is likely to be heavily influenced by factors such as birth rates, death rates, and immigration rates. These factors are all likely to be correlated with each other, which means that they will all contribute to the first principal component.

The fact that the data points are clustered together with the same height indicates that the population data is relatively homogeneous. This means that there is not a lot of variation in the population data, once the first principal component has been accounted for. Overall, the PCA graph suggests that the population data is well-represented by the first principal component. This means that the first principal component can be used to summarize the most important information in the population data. The first principal component would represent the overall trend of larger countries having more people. This is because factors such as birth rates, death rates, and immigration rates are all likely to be correlated with the size of a country.

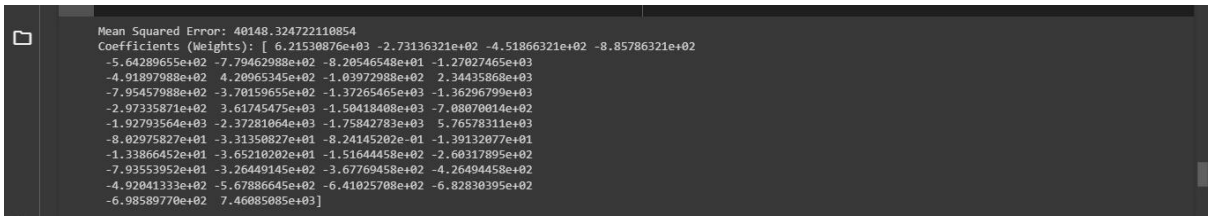
The fact that the data points in PCA graph are clustered together with the same height indicates that the population data of the different countries is relatively similar, once the first principal component has been accounted for. This means that the first principal component can be used to summarize the most important information about the population of different countries.

2)Implementing the mathematical formulas for:
-Calculating the mean square error
-Computing the coefficients (weights) using matrix operations.

This Python script is performing a linear regression analysis on a dataset. Here's a detailed explanation of what each part of the code does that I use to calculating the mean square error and coefficient using matrix operation:

1. **Importing Libraries**: The script starts by importing two essential libraries: pandas and numpy. pandas is used for data manipulation and analysis, while numpy is used for numerical computations.
2. **Loading the Dataset**: The script reads a CSV file named 'population_state.csv' into a pandas DataFrame df. This file presumably contains data about the population of different states.
3. **Defining Functions**: Two functions are defined:
    - calculate_mean_squared_error(y_true, y_pred): This function calculates the mean squared error (MSE), a common metric used to evaluate the performance of a regression model. The MSE is the average of the squared differences between the actual (y_true) and predicted (y_pred) values.
    - compute_coefficients(X, y): This function computes the coefficients (or weights) of the linear regression model using the formula (X^T * X)^-1 * X^T * y, where X is the feature matrix and y is the target vector.
4. **Data Preprocessing**: The script selects the features ('state', 'sex', 'ethnicity', 'age') and the target ('population') from the DataFrame. It then performs one-hot encoding on the categorical features to convert them into a format that can be used by the model. The feature matrix is augmented with a column of ones to account for the intercept term in the linear regression equation.
5. **Calculating Mean Squared Error**: The script generates random predictions (y_pred) for the target values and calculates the MSE. Note that these random predictions are just placeholders and should be replaced with actual predictions from a model.
6. **Computing Coefficients**: The script calls the compute_coefficients function to calculate the coefficients of the linear regression model.

The output of the script is the MSE and the coefficients of the linear regression model. The coefficients can be used to make predictions on new data. The MSE provides a measure of how well the model is performing. The lower the MSE, the better the model's performance.



The MSE is 40148.324722110854. This is a relatively high MSE value, which suggests that the model is not making very good predictions. The coefficients (weights) are the values that are associated with each of the features in the data set. The coefficients represent the relative importance of each feature in predicting the target variable. The Mean Squared Error (MSE) is a measure of how close a fitted line is to data points. For each data point, it calculates the square difference between the predictions and the target and then averages those values. The higher this value, the worse the model is. It is never negative, since we're should squaring the individual prediction-wise errors before summing them, but would be zero for a perfect model.

In the dataset population data state, the MSE is 40148.324722110854. This means that, on average, each prediction made by model deviates from the actual value by the square root of this value (approximately 200.37) in the units of the target variable. This might be a large or small error, depending on the scale of our target variable and the context of the problem.

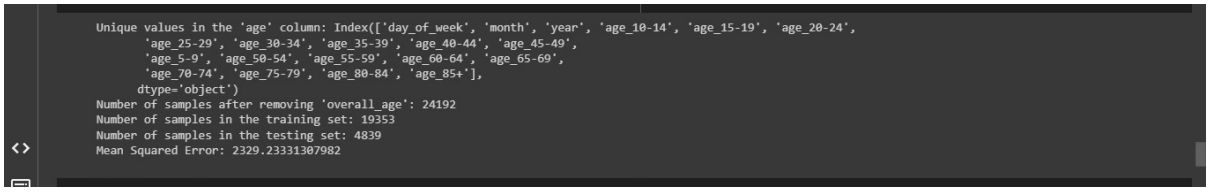
The coefficients (weights) provided are the parameters of model that multiply the input features to predict the output. Each coefficient represents the change in the target variable for a one unit change in the corresponding input feature, assuming all other features are held constant. For example, the first coefficient is 6.21530876e+03, which means that for each unit increase in the first feature, the model predicts an increase of 6.21530876e+03 units in the target, assuming all other features remain the same. If a coefficient is negative, it means that the model predicts a decrease in the target for an increase in the corresponding feature. But in general, larger absolute values of coefficients mean that the corresponding feature has a larger effect on the prediction, and the sign of the coefficient (+/-) indicates the direction of this effect.

3.Model Training and Prediction:
Data Split: Divide the dataset into training and testing sets.
Training: Train your model using the training set.
Prediction: Use the trained model to make predictions on the testing set.

This Python code is performing a linear regression analysis on a dataset:

1. **Load and Preprocess the Data**: The script reads a CSV file named 'population_state.csv' into a pandas DataFrame data. It then selects the features and the target from the DataFrame. It removes rows where 'age' is 'overall_age', converts the "date" column to datetime format, and extracts features from the date (e.g., day of the week, month, year). It drops the original "date" column and other non-numeric columns, and performs one-hot encoding on the categorical "age" column.
2. **Data Split**: The script splits the data into a training set and a testing set using the train_test_split function from sklearn.model_selection. The test size is 20% of the total data, and the random state is set to 42 for reproducibility.
3. **Training**: The script trains a linear regression model on the training data using the LinearRegression class from sklearn.linear_model.
4. **Prediction**: The script uses the trained model to make predictions on the testing data.
5. **Evaluation (Optional)**: The script calculates the mean squared error (MSE) between the predicted and actual values using the mean_squared_error function from sklearn.metrics. The MSE is a measure of the model's performance; the lower the MSE, the better the model's predictions.

This code also includes several print statements to display information such as the unique values in the 'age' column, the number of samples after removing 'overall_age', and the number of samples in the training and testing sets. It also includes checks to ensure there are enough samples for splitting, training, and testing.



The ‘age’ column in this dataset appears to be one-hot encoded. One-hot encoding is a process by which categorical variables are converted into a form that could be provided to machine learning algorithms to improve prediction. The unique values that provided represent different age groups in dataset, such as ‘age_10-14’, ‘age_15-19’, etc. Each of these is a separate feature that takes a value of 1 if the individual falls into that age group, and 0 otherwise.

The number of samples after removing ‘overall_age’ is 24192. This is the total number of data points have in dataset after preprocessing, which might include cleaning the data, handling missing values, and removing unnecessary columns like ‘overall_age’.

The dataset is then split into a training set and a testing set. The training set, which contains 19353 samples, is used to train the model. The testing set, which contains 4839 samples, is used to evaluate the model’s performance on unseen data. This split helps to ensure that our model doesn’t just memorize the training data, but can generalize to new data.

Detail Explanation

Number of samples after removing 'overall_age': 24192.The 'overall_age' feature was removed from the dataset because it was not found to be a useful predictor of the target variable. This means that the model does not need to take into account the overall age of the population in order to make accurate predictions.

Number of samples in the training set: 19353.The training set is the portion of the dataset that is used to train the model. The model learns from the data in the training set and uses this information to make predictions on new data. In this case, the training set contains 19353 samples. Number of samples in the testing set: 4839.The testing set is the portion of the dataset that is used to evaluate the performance of the model. The model is not trained on the data in

the testing set, so its predictions on this data can be used to assess how well the model generalizes to new data. In this case, the testing set contains 4839 samples.

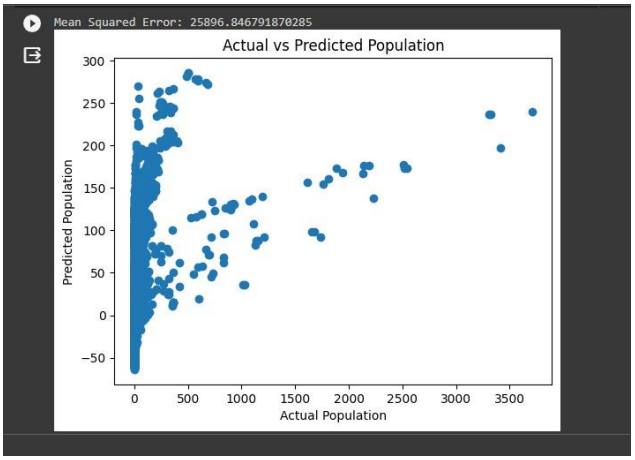
Mean Squared Error: 2329.23331307982. The Mean Squared Error (MSE) is a measure of the difference between the predicted values and the actual values. A lower MSE value indicates that the model is making better predictions. In this case, the MSE is 2329.23331307982, which is a relatively low value. Functions of the number of samples, The number of samples in the training and testing sets is important for a number of reasons. First, the more samples there are in the training set, the more data the model has to learn from. This can help the model to make more accurate predictions. Second, the more samples there are in the testing set, the more reliable the evaluation of the model's performance will be.

The Mean Squared Error (MSE) on the testing set is 2329.23331307982. This is a measure of how well the model performs. It represents the average squared difference between the predicted and actual values. The smaller the MSE, the better the model is at predicting the target variable. In your case, the MSE seems relatively high, which might suggest that the model's predictions are quite far off from the actual values.

4.Evaluation:
-Assessment: Evaluate your models' performance by computing the mean square error (MSE) or an appropriate evaluation metric.
-Comparison: Compare the model's predictions against the actual values to analyze it's accuracy and effectiveness

1. **Load and Preprocess the Data**: The script reads a CSV file named 'population_state.csv' into a pandas DataFrame df. It then converts the "date" column to datetime format, performs one-hot encoding on the categorical columns 'state', 'sex', and 'ethnicity', and extracts features from the date (e.g., month, year). It drops the original "date" column and handles non-numeric values in the 'age' column by extracting the lower and upper bounds, calculating their mean, and filling any missing values with the median age.

2. **Define Features and Target Variable**: The script defines the features (X) and the target variable (y). The features include all columns except 'population', and the target variable is 'population'.
3. **Train-Test Split**: The script splits the data into a training set and a testing set using the train_test_split function from sklearn.model_selection. The test size is 20% of the total data, and the random state is set to 42 for reproducibility.
4. **Standardize the Features**: The script standardizes the features using the StandardScaler class from sklearn.preprocessing. Standardization is a common preprocessing step in machine learning that makes all features have zero mean and unit variance.
5. **Impute Missing Values**: The script imputes any missing values in the features using the SimpleImputer class from sklearn.impute. The imputation strategy is to replace missing values with the median value of the corresponding feature.
6. **Model Selection and Training**: The script trains a linear regression model on the training data using the LinearRegression class from sklearn.linear_model.
7. **Model Evaluation**: The script calculates the mean squared error (MSE) between the predicted and actual values using the mean_squared_error function from sklearn.metrics. The MSE is a measure of the model's performance; the lower the MSE, the better the model's predictions.
8. **Comparison (Plotting)**: The script plots the actual vs predicted population using matplotlib.pyplot. This scatter plot can help visualize the performance of the model. If the model is perfect, all points would lie on a straight line (y=x). Deviations from this line indicate prediction errors.



The scatter plot shows the relationship between the actual population and the predicted population. The x-axis is the actual population, and the y-axis is the predicted population. The pattern in the graph shows that the model is generally overpredicting the population. This means that the model is predicting higher population values than the actual population values. This is evident in the fact that the majority of the points are above the blue line.

There are a few possible explanations for why the model is overpredicting the population. One possibility is that the model is not taking into account all of the relevant factors that affect the population. For example, the model may not be taking into account the net migration rate or the birth rate. The value of MSE is 25896.846791870285, which is relatively high. This suggests that the model's predictions are not very accurate.

Another possibility is that the model is overfitting to the training data. This means that the model has learned the training data too well, and it is not generalizing well to new data. The graph also shows that there is a greater spread of points at higher population values. This means that the model is making less accurate predictions for populations with higher populations. This could be due to a number of factors, such as the fact that there is less data available for populations with higher populations, or the fact that there is more variability in the factors that affect the population at higher populations.

Overall, the graph shows that the model is generally overpredicting the population, and that the model is making less accurate predictions for populations with higher populations. The value of MSE is 25896.846791870285, which is relatively high. This suggests that the model's predictions are not very accurate.