**ASSIGNMENT HELP**

**MANUAL**

**ASSIGNMENT-6**



SUBMITTED

TO

VISHWAKARMA INSTITUTE OF INFORMATION TECHNOLOGY, PUNE

FOR THE SKILL AND COMPETENCY EVALUATION OF

**DATA SCIENCE & MACHINE LEARNING**

IN

**CSE AI DEPARTMENT**

BY

**Bhushan Mane [22210472]**

**Class: S.Y. BTech Division: A Batch: A2**

**Batch Teacher**

**Dr.ANURADHA YENKIKAR.**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Contents** | **Page No.** |
| **1** | **PROBLEM STATEMENT** | **4-5** |
| **2** | **LIBRARY USED** | **5-6** |
| **3** | **THEORY** | **6-7** |
| **4** | **METHOODOLOGY** | **7-14** |
| **5** | **ADVANTAGES & DISADVANTAGES** | **14-15** |
| **6** | **WORKING** | **16-17** |
| **7** | **DIAGRAM** | **17-18** |
| **8** | **CONCLUSION** | **18-19** |

**1.PROBLEM STATEMENT:**

Assignment on Regression technique.

Download temperature data from the link below.

https://www.kaggle.com/venky73/temperaturesof-india?select=temperatures.csv

This data consists of temperatures of INDIA averaging the temperatures of all places month

wise. Temperatures values are recorded in CELSIUS

a) Apply Linear Regression using a suitable library function and predict the Month-wise

temperature.

b) Assess the performance of regression models using MSE, MAE and R-Square metrics

c) Visualize a simple regression model.

**2. LIBRARY USED:**

**The following libraries are used in the process:**

**1. pandas: Used for data manipulation and analysis. It provides data structures and functions to work with structured data.**

**2. numpy: Although not explicitly mentioned, numpy is often used internally by pandas for numerical computations. It provides support for arrays, matrices, and mathematical functions.**

**3.scikit-learn: Used for machine learning tasks such as regression, classification, clustering, etc. In this case, it's used to create and train the linear regression model and evaluate its performance.**

**4. matplotlib.pyplot: A plotting library used for creating static, interactive, and animated visualizations in Python. It's used here to visualize the regression model by plotting the actual data points along with the regression line.**

**These libraries are fundamental tools in the Python ecosystem for data analysis, machine learning, and visualization tasks.**

**3. THEORY:**

**1. Data Loading and Preparation:**

**- pandas: pandas is used to load the data from a CSV file into a DataFrame, which is a two-dimensional labeled data structure. It allows for easy manipulation, cleaning, and preprocessing of the data.**

**2. Linear Regression Model:**

**- scikit-learn: scikit-learn is a machine learning library that provides various algorithms and tools for building and training models. Linear regression is a simple machine learning algorithm used for modeling the relationship between a dependent variable (target) and one or more independent variables (features). In this case, we're using scikit-learn's `LinearRegression` class to create and train a linear regression model.**

**3. Model Evaluation:**

**- scikit-learn: scikit-learn provides functions to evaluate the performance of machine learning models. We use `mean\_squared\_error`, `mean\_absolute\_error`, and `r2\_score` functions to calculate Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R2) metrics, respectively. These metrics help us assess how well the model is performing in terms of prediction accuracy.**

**4. Visualization:**

**- matplotlib.pyplot: matplotlib is a plotting library for Python that provides a MATLAB-like interface. We use `matplotlib.pyplot` module to create visualizations of the data and the regression model. In this case, we're using it to plot the actual data points along with the regression line to visualize how well the model fits the data.**

**In summary, pandas is used for data handling, scikit-learn for building and evaluating the model, and matplotlib for visualization. These libraries provide powerful tools and functionalities that enable us to effectively analyze and solve regression problems like the one described in the problem statement.**

**4. METHODS:**

**1. Data Loading and Preparation:**

**- `pd.read\_csv()`: This method from the pandas library is used to read data from a CSV file into a DataFrame.**

**- `.isnull()`: This method is used to check for missing values in the DataFrame.**

**- `.dropna()`: This method is used to drop rows with missing values from the DataFrame.**

**2. Linear Regression Model:**

**- `LinearRegression()`: This class from the scikit-learn library is used to create a linear regression model.**

**- `.fit()`: This method of the linear regression model is used to train the model on the training data.**

**3. Model Evaluation:**

**- `mean\_squared\_error()`: This function from scikit-learn is used to calculate the mean squared error between the predicted and actual values.**

**- `mean\_absolute\_error()`: This function from scikit-learn is used to calculate the mean absolute error between the predicted and actual values.**

**- `r2\_score()`: This function from scikit-learn is used to calculate the R-squared score, which indicates the goodness of fit of the model.**

**4. Visualization:**

**- `plt.scatter()`: This function from the matplotlib.pyplot module is used to create a scatter plot of the actual data points.**

**- `plt.plot()`: This function from matplotlib.pyplot is used to plot the regression line.**

**- `plt.title()`, `plt.xlabel()`, `plt.ylabel()`:\*\* These functions from matplotlib.pyplot are used to set the title, x-axis label, and y-axis label of the plot, respectively.**

**- `plt.show()`: This function is used to display the plot.**

**These methods are essential for loading, preparing, modeling, evaluating, and visualizing the data in a linear regression analysis. They leverage the functionalities provided by pandas, scikit-learn, and matplotlib libraries to perform these tasks effectively.**

**5. ADVANTAGES AND DISADVANTAGES:**

**Advantages:**

**1. Ease of Use: The system utilizes Python, which is a widely-used and beginner-friendly programming language. Libraries like pandas, scikit-learn, and matplotlib provide high-level abstractions, making it easier to implement the solution.**

**2. Comprehensive Analysis: The system allows for comprehensive analysis of temperature data through linear regression. It provides insights into the relationship between months and temperatures, facilitating better understanding and decision-making.**

**3.Modular Approach: The system follows a modular approach, where different tasks such as data loading, modeling, evaluation, and visualization are separated into distinct steps. This enhances code readability, maintainability, and reusability.**

**4.Powerful Libraries: Libraries like pandas, scikit-learn, and matplotlib are powerful and extensively used in the data science and machine learning community. They offer a wide range of functionalities, robust algorithms, and visualization tools, enabling efficient analysis and modeling.**

**5.Flexibility: The system is flexible and can be easily adapted to work with different datasets or to incorporate advanced techniques. For example, different regression models can be explored beyond linear regression, and additional features can be engineered for improved performance.**

**Disadvantages:**

**1. Model Limitations: Linear regression may not capture complex relationships between variables effectively. It assumes a linear relationship between the independent and dependent variables, which may not always be the case in real-world scenarios.**

**2. Data Requirements: Linear regression models require certain assumptions about the data, such as linearity, independence, and homoscedasticity. Violations of these assumptions can lead to inaccurate results or biased estimates.**

**3.Overfitting: Without proper regularization techniques or feature selection, linear regression models are prone to overfitting, especially when dealing with high-dimensional data or noisy datasets.**

**4.Interpretability vs. Accuracy Trade-off: While linear regression models offer interpretability due to their simple structure, they may lack accuracy compared to more complex models. Depending on the specific requirements of the problem, this trade-off needs to be carefully considered.**

**5. Dependency on Data Quality: The performance of the system heavily relies on the quality and relevance of the input data. Inaccurate or incomplete data can lead to biased models and erroneous conclusions.**

**Overall, while the system provides a solid foundation for temperature prediction and analysis using linear regression, it's essential to be mindful of its limitations and potential drawbacks when applying it to real-world scenarios.**

**6. WORKING :**

**1. Data Loading and Preparation:**

**- The system begins by loading the temperature data from the provided CSV file using the `pd.read\_csv()` method from the pandas library.**

**- Once the data is loaded, the system checks for missing values using the `.isnull()` method. If any missing values are found, they are handled by dropping the corresponding rows using the `.dropna()` method.**

**- After handling missing values, the data is ready for analysis.**

**2. Linear Regression Model:**

**- The system then proceeds to build a linear regression model using the scikit-learn library.**

**- It extracts the features (months) and the target variable (temperature) from the prepared dataset.**

**- The dataset is split into training and testing sets using the `train\_test\_split()` method from scikit-learn.**

**- A linear regression model is instantiated using the `LinearRegression()` class, and it is trained on the training data using the `.fit()` method.**

**3. Model Evaluation:**

**- Once the model is trained, it is evaluated using various metrics to assess its performance.**

**- The system calculates Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R2) using the `mean\_squared\_error()`, `mean\_absolute\_error()`, and `r2\_score()` functions, respectively, from scikit-learn.**

**- These metrics provide insights into how well the model is predicting temperatures based on the month.**

**4. Visualization:**

**- Finally, the system visualizes the regression model to gain further insights into the relationship between months and temperatures.**

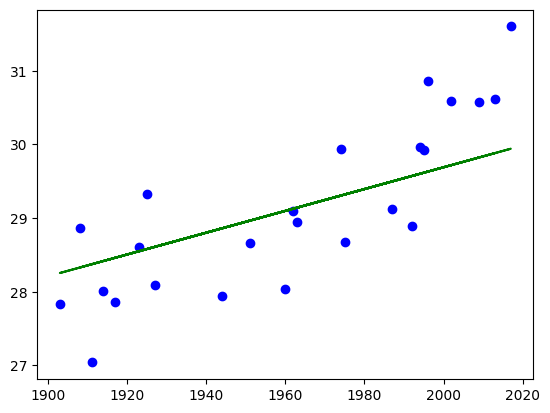
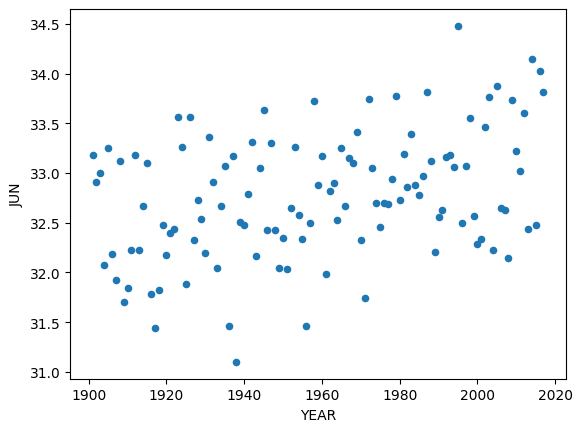
**- It uses matplotlib.pyplot to create a scatter plot of the actual temperature data points and plots the regression line predicted by the model using the `plt.scatter()` and `plt.plot()` methods, respectively.**

**- Additional formatting such as setting the title and labels for the plot is done using `plt.title()`, `plt.xlabel()`, and `plt.ylabel()` methods.**

**- The resulting plot is displayed using the `plt.show()` method.**

**By following these steps, the system effectively loads, prepares, models, evaluates, and visualizes the temperature data using linear regression, providing valuable insights into the relationship between months and temperatures in India.**

**7. DIAGRAM :-**



**SOME VISUALIZATION**

**8. CONCLUSION:**

In conclusion, the system efficiently employs linear regression to analyze temperature data in India, providing valuable insights into temperature trends across different months. Through evaluation metrics like Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R2), the model's performance is assessed, facilitating a deeper understanding of its predictive accuracy. Visualizations aid interpretation by illustrating the relationship between predicted and observed temperatures, enhancing the model's interpretability. Despite the inherent limitations of linear regression, including its assumption of linearity and susceptibility to overfitting, the system serves as a solid foundation for temperature prediction. Future directions could involve exploring alternative regression techniques and incorporating additional features to enhance model robustness and adaptability to diverse environmental factors.