Assignment 2

Title : Assignment on Regression technique

Aim: Download temperature data from below link. https://www.kaggle.com/venky73/temperatures- of-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 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 simple regression model

I] Theory

Regression analysis is a statistical method that helps us to analyze and understand the relationship between two or more variables of interest,
In regression, we normally have one dependent variable and one or more independent variables. Here we try to "regress" the value of dependent variable "Y" with the help of the independent variables. In other words, we are trying to understand, how does the value of 'Y' change w.r.t change in 'X'.

- **Dependent Variable:** This is the variable that we are trying to understand or forecast.
- **Independent Variable:** These are factors that influence the analysis or target variable and provide us with information regarding the relationship of the variables with the target variable.

General Uses of Regression Analysis

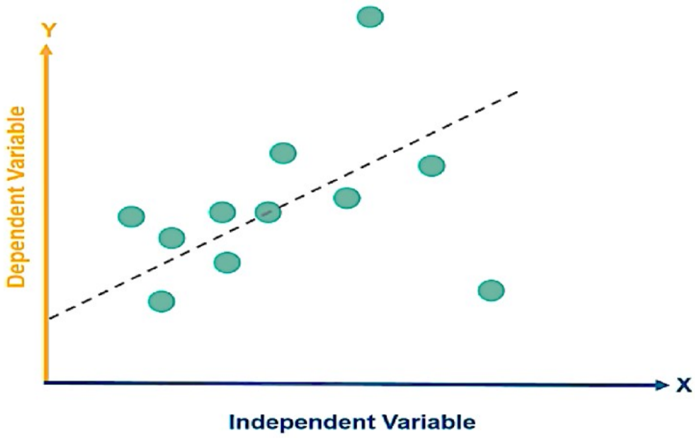
Regression analysis is used for prediction and forecasting. This has a substantial overlap to the field of machine learning. This statistical method is used across different industries such as,

- Financial Industry- Understand the trend in the stock prices, forecast the prices, evaluate risks in the insurance domain
- Marketing- Understand the effectiveness of market campaigns, forecast pricing and sales of the product.
- Manufacturing- Evaluate the relationship of variables that determine to define a better engine to provide better performance
- Medicine- Forecast the different combination of medicines to prepare generic medicines for diseases.

Types of Regression

1. Linear Regression

The simplest of all regression types is Linear Regression where it tries to establish relationships between Independent and Dependent variables. The Dependent variable considered here is always a continuous variable. Linear Regression is a predictive model used for finding the ***linear*** relationship between a dependent variable and one or more independent variables.



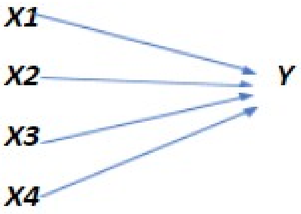
Here, ‘Y’ is our dependent variable, which is a continuous numerical and we are trying to understand how does ‘Y’ change with ‘X’. If the relationship with the dependent variable is in the form of single variables, then it is known as Simple Linear Regression

Simple Linear Regression

X* ——> *Y

If the relationship between Independent and dependent variables are multiple in number, then it is called Multiple Linear Regression

Multiple Linear Regression



Simple Linear Regression Model

As the model is used to predict the dependent variable, the relationship between the variables can be written in the below format.

$\boldsymbol{Y_i} = \beta_0 + \beta_1 X_i + \varepsilon_i$

Where,

$\boldsymbol{Y_i}$ – Dependent variable

β_0 — Intercept

β_1 – Slope Coefficient

X_i – Independent Variable

ε_i – Random Error Term

The main factor that is considered as part of Regression analysis is understanding the variance between the variables. For understanding the variance, we need to understand the measures of variation.

$$\underset{\text{Total Sum of Squares}}{SST} = \underset{\text{Regression Sum of Squares}}{SSR} + \underset{\text{Error Sum of Squares}}{SSE}$$

- **SST = total sum of squares (Total Variation)**
  - Measures the variation of the Y i values around their mean Y
- **SSR = regression sum of squares (Explained Variation)**
  - Variation attributable to the relationship between X and Y
- **SSE = error sum of squares (Unexplained Variation)**
  - Variation in Y attributable to factors other than X

With all these factors taken into consideration, before we start assessing if the model is doing good, we need to consider the assumptions of Linear Regression.

Assumptions:

Since Linear Regression assesses whether one or more predictor variables explain the dependent variable and hence it has 5 assumptions:

1. Linear Relationship
2. Normality
3. No or Little Multicollinearity
4. No Autocorrelation in errors
5. Homoscedasticity

With these assumptions considered while building the model, we can build the model and do our predictions for the dependent variable. For any type of machine learning model, we need to understand if the variables considered for the model are correct and have been analysed by a metric. In the case of Regression analysis, the statistical measure that evaluates the model is called the ***coefficient of determination which is represented as r^2.*** The coefficient of determination is the portion of the total variation in the dependent variable that is explained by variation in the independent variable. A higher value of ***r^2*** better is the model with the independent variables being considered for the model.

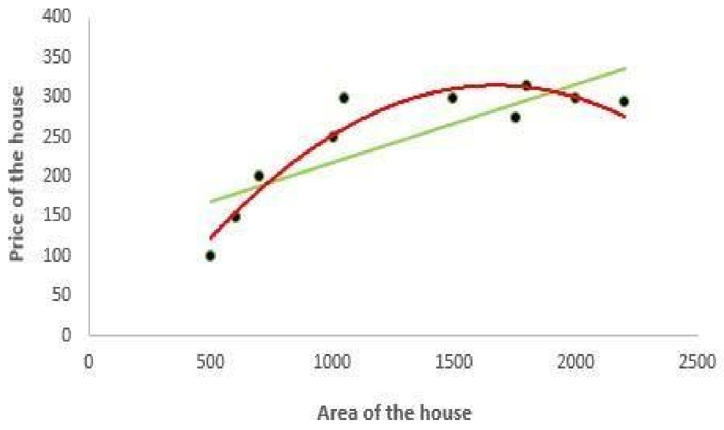
r^2 = SSR

SST

Note: The value of r^2 is the range of $0 \leq r^2 \leq 1$

2.Polynomial Regression

This type of regression technique is used to model nonlinear equations by taking polynomial functions of independent variables. In the figure given below, you can see the red curve fits the data better than the green curve. Hence in the situations where the relationship between the dependent and independent variable seems to be non-linear, we can deploy **Polynomial Regression Models**.



Thus a polynomial of degree k in one variable is written as:

$$y = \beta_0 + \beta_1 X + \beta_2 X^2 + \ldots + \beta_k X^k + \varepsilon$$

Here we can create new features like

$$X_1 = x, X_2 = x^2, \ldots , X_k = x^k$$

and can fit linear regression in a similar manner. In case of multiple variables say X1 and X2, we can create a third new feature (say X3) which is the product of X1 and X2 i.e.

$$X_3 = X_1 * X_2$$

The main drawback of this type of regression model is if we create unnecessary extra features or fitting polynomials of higher degree this may lead to overfitting of the model.

3.Logistic Regression

Logistic Regression is also known as Logit, Maximum-Entropy classifier is a supervised learning method for classification. It establishes a relation between dependent class variables and independent variables using regression. The dependent variable is categorical i.e. it can take only integral values representing different classes. The probabilities describing the possible outcomes of a query point are modelled using a logistic function. This model belongs to a family of discriminative classifiers. They rely on attributes which discriminate the classes well. This model is used when we have 2 classes of dependent variables. When there are more than 2 classes, then we have another regression method which helps us to predict the target variable better.

There are two broad categories of Logistic Regression algorithms

1. Binary Logistic Regression when the dependent variable is strictly binary
2. Multinomial Logistic Regression when the dependent variable has multiple categories.

There are two types of Multinomial Logistic Regression

1. Ordered Multinomial Logistic Regression (dependent variable has ordered values)
2. Nominal Multinomial Logistic Regression (dependent variable has unordered categories)

III] Python Libraries and functions required

numpy : NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy stands for Numerical Python. To import numpy use

```
import numpy as np
```

pandas: pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language. To import pandas use

```
import pandas as pd
```

sklearn : Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib. For importing train_test_ split, linear regression, metrics use following

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn import metrics as m
import matplotlib.pyplot as plt
```

matplotlib: Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

IV] Sample Code with comments

```
# Assignment 2: Linear Regression

import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn import metrics as m
import matplotlib.pyplot as plt

trainData = pd.read_csv(r"C:\Users\Dell 1\Desktop\TEIT-
ML\temperatures.csv")

cols = list(trainData.columns)

# Plot Subplots
fig, axs = plt.subplots(2, 6, )

for i in range(1, 13):
    X = trainData[[cols[0]]]
    Y = trainData[[cols[i]]]

    print('\n\nMONTH:', cols[i])

    X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.2, random_state=1)

    LR_model = LinearRegression()

    LR_model.fit(X_train, Y_train)

    r_sq = LR_model.score(X_train, Y_train)

    print("determination coefficient:", r_sq)

    print("INTERCEPT:", LR_model.intercept_)
    print('SLOPE\t:', LR_model.coef_)

    Y_pred = LR_model.predict(X_test)
    print('PREDICTION:', Y_pred, sep='\n')

    print('METRICS:')
    print('MSE: ', m.mean_squared_error(Y_test, Y_pred, squared=True))
    print('RMSE: ', m.mean_squared_error(Y_test, Y_pred,
squared=False))
    print('MAE: ', m.mean_absolute_error(Y_test, Y_pred))
    print('R-Squared Score: ', m.r2_score(Y_test, Y_pred))
    # Plot Graph for Temperature vs Year for Training Data
    # axs[0, 0].scatter(X_train, Y_train, color='black')
    # axs[0, 0].plot(X_train, LR_model.predict(X_train), color='blue', 
linewidth=3)
    # axs[0, 0].set_title("Temperature vs Year: Training Data")
    # axs[0, 0].set_xlabel("Year")
    # axs[0, 0].set_ylabel("Temperature")

    if i < 7:
        j = 0
        k = i - 1
    else:
        j = 1
        k = i - 7

    # Plot Graph for Temperature vs Year for Testing Data
    axs[j, k].scatter(X_test, Y_test, color='red')
    axs[j,  k].plot(X_test,  LR_model.predict(X_test),  color='black', 
linewidth=3)
    axs[j, k].set_title("Temp vs Year: {}".format(cols[i]))
    axs[j, k].set_xlabel("Year")
    axs[j, k].set_ylabel("Temperature")

plt.show()
```

V] Output of Code:

Note: Run the code and attach your output of the code here.