Assignment-2

Ginjala Lohith Reddy(210050054)

May 16, 2023

1 What I have learnt

In this week I have learnt a lot about datasets how to manage them and plot different types of graphs like **Boxplot**, **Heatmap**, **Q-Q Plots**, **Scatter Plots** an many more. I have learnt how to find column names from Dataframe, Datatypes of Columns, **Mean** and **variance** of each column. I have learnt how to compute **correlation matrix** and plot **Heatmap** with annotations using **Matplotlib** library. I have learnt about Q-Q Plots for dataset and how to analyse the plots. I have learnt about Box and whisker's plot, **Histogram** and **Scatter Plots** how to plot them using **Matplotlib**.

2 Mean and Variance Conclusions

Mean: The average value of the data in a column is known as the **mean**. It provides us with a sense of the column's data's core trend. A **high mean** indicates that the values of the data in that column are generally higher, whereas a **low mean** indicates that the values of the data are generally lower.
Variance: The spread of the data values vary from the **mean** is indicated by the **variance**. In contrast to a high variance, which indicates that the data values are more widely dispersed from the mean, a low variance indicates that the data values in that column are concentrated around the mean.
We can determine the correlation between two columns in a dataset by examining the relationship between their means and variances. For instance, we would anticipate that the means and variances of two columns with a high positive correlation would be comparable. A high variance can also suggest the presence of **outliers** in the data. **Outliers** are data points that are significantly different from the other data points in the column. They can affect the mean and increase the variance.

3 Choice of Number of Bins for histogram

I have selected number of bins for each column of Histogram for dataset **Iris.csv** near to as the **range of values of column** divided by the least error value for column. We have different number of bins to different columns. I have taken bins as this values because it is used to represent data with good accuracy and precision. We can see it in histograms and Iris Dataset.

4 Heatmap Conclusions

Strong Positive Correlation: A strong positive correlation between two variables indicates that the two tend to rise together as one rises. **Dark Red** colours on the heatmap denote strong positive associations. For instance, the Iris dataset's petal length and petal breadth columns show a significant positive association.

Strong Negative Correlation: If there is a significant negative correlation between two variables, it suggests that as one variable rises, the other tends to fall. **Dark Blue** colours on the heatmap represent strong negative relationships. There are no pronounced negative associations in the Iris dataset.

Weak Correlation: A weak or nonexistent linear link between two variables is shown if the correlation coefficient between them is close to zero. Weak correlations are represented by **light colours** (almost white and pale blue) in the heatmap. For instance, the Iris dataset shows a weak association between the sepal length and sepal breadth columns.

The heatmap's colour intensity gives an indication of how strong the correlation is. Stronger correlations are represented by darker (blue or red) colours, whilst weaker correlations are represented by lighter colours.

The **direction** of the correlation is indicated by the sign of the correlation coefficient **(+ or -)**. When two variables are positively correlated, they move in the same direction, and when they are negatively correlated, they move in the opposite way.

We may infer from the **heatmap** and **correlation matrix** that the columns for **PetalLengthCm** and **PetalWidthCm** have a high positive association, whereas the columns for **SepalLengthCm** and **SepalWidthCm** show a weak correlation. This implies that while the association between sepal length and width is somewhat weaker, there is a strong relationship between petal length and width. These results are consistent with what we already know about the **Iris** dataset, where it is known that **petal** measures have a stronger relationship with the species categorization than **sepal** measurements.

5 Q-Q Plots for Dataset

A **Q-Q plot**(Quantile-Quantile plot), is a graphical tool used to assess how well a given dataset aligns with a particular theoretical distribution, such as the normal distribution. It provides a visual comparison between the observed quantiles of the dataset and the expected **quantiles** of the **theoretical distribution**. To construct a Q-Q plot, the dataset is sorted in ascending order, and the **theoretical quantiles** are calculated based on the chosen distribution. The dataset's quantiles are then plotted against the theoretical quantiles on a scatter plot. The line of equality, representing a perfect match between the dataset and the theoretical distribution, is also included. By observing the positioning of the points in relation to the **line of equality**, deviations from the assumed distribution can be identified. If the points **closely** align with the line, it suggests a **good fit** to the chosen distribution, while deviations may indicate departures such as heavier tails or differing **peakedness**. **Q-Q plots** provide a visual assessment of distributional assumptions and can be useful in various statistical analyses to determine if the data meets the assumed distribution requirements.
SepalLengthCm: The points on the Q-Q plot for **SepalLengthCm** exhibit a **roughly** diagonal relationship and a somewhat linear relationship. This implies that the column of sepal length may roughly follow a normal distribution.
SepalWidthCm: We can observe some deviations from the diagonal line in the Q-Q plot for **SepalWidthCm**, especially near the tails. This suggests that there may be **skewness** or large tails in the sepal width column, which would suggest a deviation from the normal distribution.
PetalLengthCm: The points closely match the diagonal line in the Q-Q plot for **PetalLengthCm**, which clearly shows a significant deviation from linear relationship. This shows that a normal distribution is deviated probably in the case for the petal length column.
PetalWidthCm: The points on the Q-Q plot for petal width also follow in the lines of PetalLengthCm column which has a good amount deviation from centra line and probably deviated from normal distribution

6 Strategies to preprocess data

Load the data: First, load the data into a Pandas DataFrame using pd.read_csv() or other appropriate functions.
Check for data types: We determine the data types of the features in the data, use the Pandas **dtypes** property. You might need to convert the data to the proper kinds or apply alternative preprocessing techniques depending on the data types.
Explore the data: We the use **Matplotlib** library to study the distribution, correlations between features, and probable outliers of the data. To comprehend the data, you can use scatter plots, histograms, box plots, and other visualisation approaches.
Preprocess the data: We might need to execute extra preprocessing processes like **normalisation** or **scaling** depending on the precise needs of the data. Additionally, you might need to use the Pandas **get_dummies** function or other suitable conversion methods to change categorical data into numerical variables.
Extract Information: Now after preprocessing the data, we can apply machine learning models to retrieve insightful data. Depending on the type of data and the objectives of the research, we can employ a variety of models, including regression, classification, clustering, and dimensionality reduction.

These tactics might not always be successful. It could be challenging to fill in missing values or normalise the data, for instance, if there are too many missing values or the distribution is excessively skewed. Additional data cleaning or feature engineering approaches might be necessary in such circumstances. Additionally, if the data is too big, Matplotlib might not be able to visualise it effectively, and you might need to utilise different visualisation methods or libraries.