**Insights and Q&A**

This file would act as a tracking book of the Insights and challenges I acquired/ faced while building the **career-counsellor-ml** project. Along with the Insights, I would also keep the record of a few Questions and Answers which I think are important, and which really forced me to think about the concept deeply.

**I – Insights**

**Q – Questions**

**C – Challenges**

**I1 -** In the describe function, include = "all" parameter makes sure to include all the columns while drawing the statistics table.

**Q1 – Which type of graphs are used for which type of data?**Ans) Categorical variables 🡪 Bar plot (Matplotlib), Count plot (Seaborn)  
 Numerical variables 🡪 Histogram, Boxplot  
 Time-series 🡪 Line plot  
 Multiple Numeric variables 🡪 Heatmap (Just like in the project)

**Q2 – How to detect and handle missing values?**Ans) Missing values can be detected by “df.isnull( ).sum( )”. There are multiple ways to handle missing values – Dropping rows/ columns, Imputation methods (Replacing with Mean, Median and Mode, Interpolation methods, model-based imputation), Missingness as a feature.

**Q3 – Explain different Imputation methods for handling missing values.**Ans) Imputation methods are basically for filling missing values with estimations which helps in preserving data (dropping rows/ columns can lead to removing critical information).  
1) **Simple Imputation methods** 🡪  
**Mean Imputation** – Replace missing values with the mean of the non-missing values. Quite easy to implement but can lead to distortion in distributions if outliers are present. *Also, not suitable for categorical variables.*  
**Median Imputation** – Replacing the missing values with the median of the non-missing values. This is more robust to outliers as compared to mean. *Doesn’t consider relationship between variables and not suitable for categorical variables.*  
**Mode Imputation** – Replacing missing values with the mode of the non-missing values. Suitable for categorical variables but can reduce variability in data.

2) **Forward fill/ Backward fill** – Filling the values with the last (forward fill) /next (backward fill) observed non-missing value. *Useful for sequential data, when values are expected to be similar over time. But this assumes continuity, what if the data changes every second? This method would fail then*.

3) **Interpolation Techniques (Linear Interpolation)** – Estimates missing values based on values of neighbouring data points. *This method can capture the relationships in data more effectively. The only issue is it assumes linear relationship between data points*.

4) **Model based Imputation** -  
**Regression Imputation** – Missing values are predicted using a regression model based on other values in the dataset.   
**KNN Imputation –** Fills missing values by finding the K-nearest neighbours and observes their values to impute the missing data. (By maybe taking their mean, median or mode). *This method considers similarity and relationship between variables but can be computationally very expensive for large datasets.*

**Q4 – Explain missingness as feature and its types.**Ans) In some cases, the fact that the value is missing can be an information in itself, so a new binary variable can be created (0 for present, 1 for missing) for variables with missing values. Types of missing values -  
1) **MCAR (Missing Completely at Random) –** Standard deletion methods can be used for large dataset.  
2) **MAR (Missing at Random) –** Imputation methods are generally preferred.  
3) **MNAR (Missing Not at Random) –** Most challenging, requires complex modelling/ data collection efforts.

**Q5 – What are outliers and how can they be detected?**Ans) An outlier is a data point that significantly deviates from other data points in the dataset. Usually, extreme high or extreme low values. Presence of outliers can skew statistical analyses and lead to misleading conclusions.  
There are 3 major methods to detect outliers -  
1) **Box Plot** – Excellent for identifying outliers. Box plot displays the data quartiles, and points that fall outside the whiskers (typically 1.5 times the interquartile range above the third quartile or below the first quartile) are considered outliers.

2) **Z-Score** – The Z-score measures how many standard deviations a data point is from the mean. Generally, a data point is considered an outlier if the Z-score is greater than 3 or less than -3. *Most effective for data that is normally distributed*. *The formula for the Z-score of a data point x is****Z=(x−μ​)/σ***  *where µ is the mean and σ is the standard deviation of the dataset.*

3) **Interquartile Range (IQR)** – IQR is the range between the first quartile (Q1, the 25th percentile) and the third quartile (Q3, the 75th percentile). *A data point is identified as outlier if it falls* ***below Q1 – 1.5 \* IQR or above Q3 + 1.5 \* IQR***

**Q6 – How can we check for skewness and deal with it?**Ans) We can check for skewness by doing **df.skew()** or drawing out a histogram and checking the distribution. Skewness refers to symmetry in data distribution, a perfectly symmetrical distribution has zero skewness. There are a couple of ways to check for skewness like plotting a histogram and checking the skew coefficient (Pearson’s coefficient).  
A few ways to deal with skewness -  
1) **Data Transformations** – Log Transformation, square transformation  
2) **Outlier Detection/ Treatment** – Sometimes having outliers in the dataset can cause skewness.  
3) **Robust Statistical Methods** – These methods are designed to be less sensitive to outliers, not normal data or assumptions.

**Q7 – What is correlation? How to detect it?**Ans) Correlation is a statistical measure that quantifies the extent to which two variables are linearly related. Simpler terms – How much and in what direction two variables tend to move together.  
There are two key aspects of correlation, direction and strength :-  
**Strength** – The strength of the relationship determines how closely the variables move together. A strong correlation means the points on a scatter plot are very close to a straight line. A weak correlation means the points are scattered.  
**Direction** – Positive correlation *(As one variable increases, other tends to increase as well. E.g. height and weight)*, Negative correlation *(As one variable increases, other tends to decrease. E.g. hours spent studying and hours spent studying)* and No correlation *(No linear relationship between variables. Changes in one variable do not predict changes in other. E.g. shoe size and IQ)*

Correlation can be detected using both numerical and visual methods -  
1) **Visual Inspection (Scatter plots)** – Positive correlation (upwards, left to right), Negative Correlation (downwards, left to right), No correlation (Points randomly scattered).  
2) **Heatmap –** Positive correlation (Dark warm colors, closer to +1), Negative correlation (Dark cool colors, closer to -1), No correlation (Neutral colors, close to 0). While plotting heatmap, color gradients are already displayed to avoid confusion between warm and cool colors.  
3) **Calculating correlation coefficients –** Provide a numerical measure of the strength and direction of the linear relationship. *Pearson’s Product-Moment Correlation Coefficient (r), Spearman’s Rank Correlation Coefficient (ρ or rs) etc. are a few examples.*4) **Hypothesis Testing for correlation –** After calculating a correlation coefficient, we can perform a hypothesis test to determine if the observed correlation is statistically significant (not random). Null hypothesis states that there is no correlation (ρ = 0), Alternate hypothesis states that there is correlation (ρ is not 0). A p-value is calculated, and if it is below a chosen significant level, we reject the null hypothesis.

*Important: Write about the value error faced in step 6 soon after writing train\_test\_split function -* ***ValueError****: The least populated class in y has only 1 member, which is too few. The minimum number of groups for any class cannot be less than 2.*

*This is due to stratify = y not having enough members in the class to distribute evenly*