Date
Page

Data Visualization (plotting)

Is noise related to X_2

* ## Data Visualization

→ Data visualization deals with a visual representation of data and is part of data analysis.

→ It is the process of translating data into a chart, graph or other visual components.

* ## Variables (features)

→ Variables refer to characteristics, properties, or attributes that can be measured, observed, or recorded for a particular entity or unit within a dataset.

Variables → Dependent
↳ Independent

QUALITATIVE (categorical) | QUANTITATIVE (Numerical)

→ Describes the quality

* ## Univariate Analysis → checks central tendency, range
↳ only considering 1 feature at a time.

→ Used in statistics to describe a data type that contains only 1 attribute or characteristic.

classmate
Date
Page

Histogram: Frequency Distribution Graph

Box Plot: Compare the spread of the variables and get an insight into outlier.

* Bivariate Analysis: (Remember: If one var influences the change in the other variable, then you have an independent & dep var.
  - Mainly used to compare two sets of data to find a relationship b/w the two variables.
  - scatter plot, heat map, contour plot, pair plot
  - Scatter Plot: Captures the correlation b/w the two

* Multi-variate Analysis
  - Used to reveal the relationship among several variables simultaneously.
  - Assists in making informed decisions by considering multiple variables & their interactions.
  - Ex: Grouped Box Plot, Multi-variate Scatter Plot, 3D Scatter plot

* Visualization Techniques

- Distribution of data points: Box Plot, Histogram
- Comparison of data points: Multi-line chart, Bar plot, line chart
- Relationship
  ~~Correlation~~ of data points: Scatter Plot
- Composition of datapoints: Pie chart, stacked Area chart/Barchart

Page

Pair Plot

↳ preliminary idea

↳ Pair plot visualizes given data to find the relationship b/w them and plots pairwise relationships in a dataset.

↳ It is used for exploring the relationship b/w multiple variables at once

↳ Plots in a matrix format.
  → Diagonal subplots are the univariate histograms for each attribute.
  → off diagonal entries are the scatter plots.

* Joint Plot

↳ Joint plot combines univariate and bivariate plots to visualize relationship b/w 2 variables.

↳ It consists of a scatter plot for the bivariate relationship, with additional marginal plots for each variable.

↳ Helps understand correlation & distributions of two variables simultaneously.

* Heatmap

- color-coded representation of a 2D data, representing magnitude of individual values within a dataset.

- Colors are used to represent the magnitude, intensity with the color gradient scheme ranging from a lighter color (low values) to darker colors (high values)

- Displays the correlations or relationships in a correlation matrix

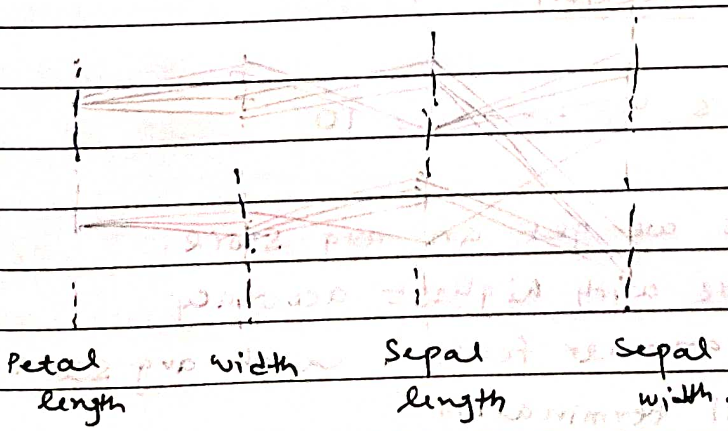
* Parallel co-ordinates

- Parallel co-ordinates allows for the comparison of multiple data records, by using parallel lines to connect points based on multiple numerical variables.

- Each vertical line is a dimension.

- A data item is connected by line segments.

- Large number of samples clutters the visualization.



* ## Dimensionality

↳ # of i/p variables or features for a dataset is referred to as its dimensionality.

$$y = w_1x_1 + w_2x_2 + \cdots\cdots + w_{30}x_{30}$$

↳ Difficulties related to training machine learning models due to high dimensional data
⇒ Curse of dimensionality

* ## Dimensionality Reduction

① Feature selection
→ Select the most relevant subset of features
→ Reducing the number of irrelevant features.

② Feature Extraction
→ Extracting / deriving information from the original features set to create a new features subspace.
→ compress data with the goal of maintaining most of the relevant info.

* ## Forward-Feature selection

Features : 1 2 3 4 10

Each feature set we get an avg score.
- Take feature with highest accuracy.
- Next with another feature check avg score.
- Repeat till termination.

FORWARD FEATURE

→ It iteratively selects one feature at a time, evaluating the model's performance after adding each feature & keeping/removing the best subset of features that maximizes/minimizes the chosen performance metrics.

10

5, 10

5, 8, 10

5, 7, 8, 10

→ computationally cheaper as it starts with no features, and only a few features are needed to reach optimal performance.

→ works well when number of features is very high, as it starts small & adds only informative features.

→ Since features are added one by one, its easier to track the contribution of each new feature.

→ only adds the "best" feature at each step without considering combination of features that might work well together later.

⇒ miss the best overall set of features.

→ If there are many features to evaluate, F.S can be slow, especially if the model is complex.

Backward Feature Selection

→ starts with all available features, iteratively removes one feature at a time, and evaluate the model's performance

→ If the perf improves, we keep the feature removed; otherwise, we add it back.

→ The final set of features that maximizes or minimizes the chosen performance metric is returned as the selected feature subset.

0 1 2 3 4 5 ~~6~~ 7 8 9 10

0 1 ~~2~~ 3 4 5 7 8 9 10

~~0~~ 1 3 4 5 6 7 8 9 10

1 3 4 5 6 7 8 9 10

→ starts with all features, so it naturally accounts for feature interactions that forward selection might miss, thereby leading to a more optimal set of features compared to F.S.

→ If the ~~in~~ initial set of features is small, backward elimination can quickly remove irrelevant ones and reach a good solution.

→ computationally expensive esp with large datasets or high dimensional feature spaces, as starting with all features means the model has to be fit with a large set of features initially.

→ starting with all features
⇒ overfitting early in the process
(if there are many irrelevant / redundant features

* Feature Extraction

- Aims to reduce # of features in a dataset by creating new features from the existing ones
(discards original ones)

feature extraction techniques:

- PCA: linear transformation techniques by finding orthogonal axes that capture the most variance.

- Isomap, t-SNE*: Non-linear dimensionality red^n technique that emphasizes the local structure of the data.

*t - distributed stochastic Neighbour Embedding.