Dimensionality Reduction & Visualization

Terminology -

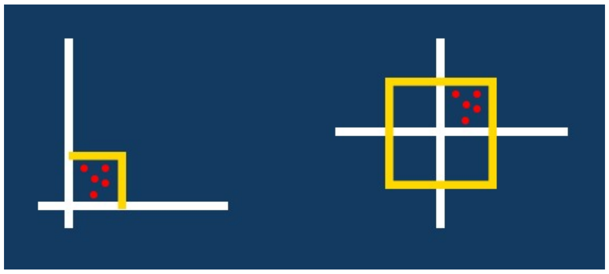
1. Normalization
2. Standardization
3. Co-variance matrix
4. Label Encoding
5. One-Hot Encoding

Data Preprocessing

- Pre-processing refers to the transformations applied to your data before feeding it to the algorithm.
- In this stage, we'll deal with outlier values, encode variables, impute missing values, and take every possible initiative which can remove inconsistencies from the data set.

Feature Scaling

- Feature scaling is the method to limit the range of variables so that they can be compared on common grounds. It is performed on continuous variables.
- It improves (significantly) the performance of some machine learning algorithms and does not work at all for others.
- We scale our data before employing a **Distance based algorithm** so that all the features contribute equally to the result.



Why to use Feature Scaling -

```
* We must have faced this issue in every project. For example, one featur
e is entirely in kilograms while the other is in grams, another one is li
ters, and so on.
* How can we use these features when they vary so vastly in terms of what
they’re presenting?
```

ML Algo which require Feature Scaling-

1. **Gradient Descent Based Algorithms-**

    - Machine learning algorithms like linear regression, logistic regression, neural network, etc. that use gradient descent as an optimization technique require data to be scaled.

2. **Distance-Based Algorithms-**

    - Distance algorithms like KNN, K-means, and SVM are most affected by the range of features.
    - This is because behind the scenes they are using distances between data points to determine their similarity.

3. **Tree-Based Algorithms-**

    - Tree-based algorithms, on the other hand, are fairly insensitive to the scale of the features.
    - A Decision Tree is only splitting a node based on a single feature. The decision tree splits a node on a feature that increases the homogeneity of the node. This split on a feature is not influenced by other features.

Feature Standardization

- Standardization (or Z-score normalization) is the process where the features are rescaled so that they'll have the properties of a **standard normal distribution** with μ=0 and σ=1, where μ is the mean (average) and σ is the standard deviation from the mean.
- Standard scores (also called z scores) of the samples are calculated as follows -

$$z = \frac{x - \mu}{\sigma}$$

Feature Normalization -

Normalization is a scaling technique in which values are shifted and rescaled so that they end up **ranging between 0 and 1**. It is also known as Min-Max scaling.

Here's the formula for normalization-

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Here, Xmax and Xmin are the maximum and the minimum values of the feature respectively.

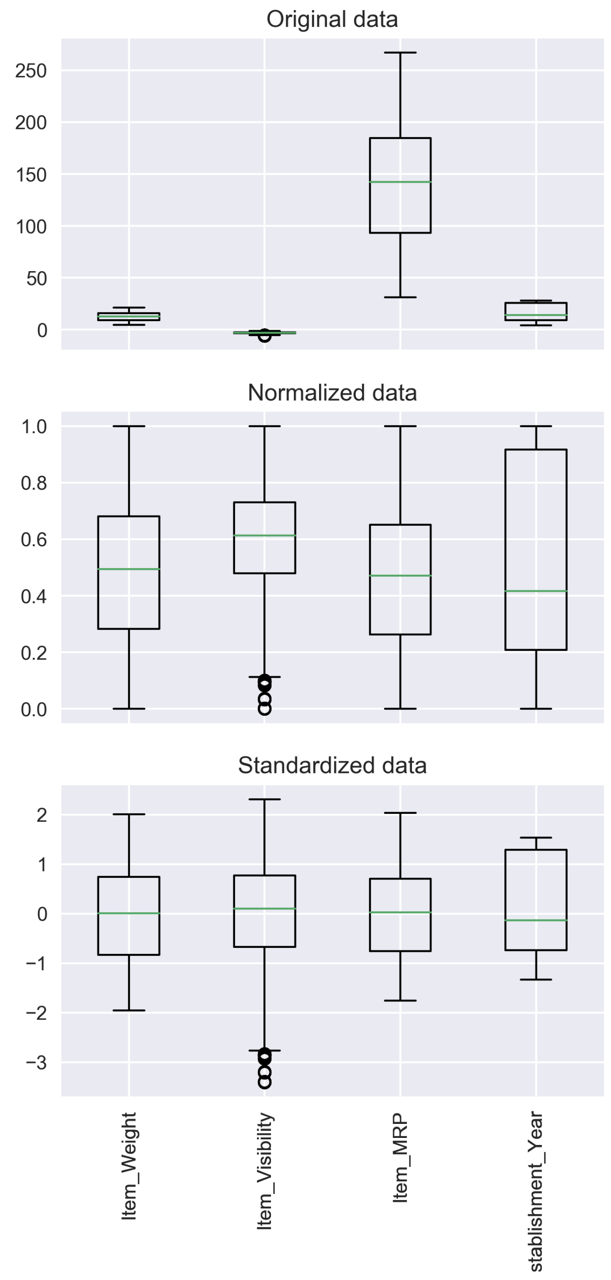
- When the value of X is the minimum value in the column, the numerator will be 0, and hence X' is 0
- On the other hand, when the value of X is the maximum value in the column, the numerator is equal to the denominator and thus the value of X' is 1
- If the value of X is between the minimum and the maximum value, then the value of X' is between 0 and 1

Which one to choose- Normalize or Standardize?

- Normalization is good to use when you know that the distribution of your data does not follow a Gaussian distribution. This can be useful in algorithms that do not assume any distribution of the data like K-Nearest Neighbors and Neural Networks.

- Standardization is helpful in cases where the data follows a **Gaussian distribution**. However, this does not have to be necessarily true. Also, unlike normalization, standardization does not have a bounding range.
  - So, even if you have outliers in your data, they will not be affected by standardization.

- Choosing Normalization or Standardization will depend on your problem and the machine learning algorithm we are using.
- There is no hard and fast rule to tell you when to normalize or standardize your data.
- We can always start by fitting your model to raw, normalized and standardized data and compare the performance for best results.



One-Hot Encoding vs Label Encoding using Scikit-Learn

Machines understand numbers, not text. We need to convert each text category to numbers in order for the machine to process them using mathematical equations.

This is where Label Encoding and One-Hot Encoding come into the picture.

Categorical Encoding-

- Categorical encoding is a process of converting categories to numbers.

- We need to convert categorical columns to numerical columns so that a machine learning algorithm understands it.

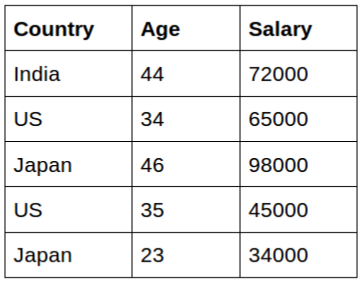
Types of Categorical Encoding-

1. Label Encoding
2. One-Hot Encoding

1. Label Encoding -

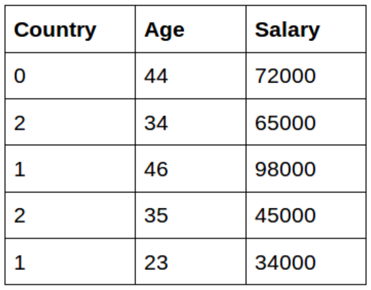
- Label Encoding is a popular encoding technique for handling categorical variables.
- In this technique, each label is assigned a unique integer based on alphabetical ordering.

The first column, Country, is the categorical feature as it is represented by the object data type and the rest of them are numerical features -



Country	Age	Salary
India	44	72000
US	34	65000
Japan	46	98000
US	35	45000
Japan	23	34000

After applying Label Encoding, Country column will look like -



Country	Age	Salary
0	44	72000
2	34	65000
1	46	98000
2	35	45000
1	23	34000

Drawback of Label Encoding - Country names do not have an order or rank. But, when label encoding is performed, the country names are ranked based on the alphabets. Due to this, there is a very high probability that the model captures the relationship between countries such as India < Japan < the US.

So how can we overcome this problem, we use One-Hot Encoding.

2. One-Hot Encoding -

- One-Hot Encoding is another popular technique for treating categorical variables.
- It simply creates additional features(column) based on the number of unique values in the categorical feature in Binary form. Every unique value in the category will be added as a feature.

The first column, Country, is the categorical feature as it is represented by the object data type and the rest of them are numerical features -

Country	Age	Salary
India	44	72000
US	34	65000
Japan	46	98000
US	35	45000
Japan	23	34000

After applying One-Hot Encoding, Country column will look like -

0	1	2	Age	Salary
1	0	0	44	72000
0	0	1	34	65000
0	1	0	46	98000
0	0	1	35	45000
0	1	0	23	34000

Drawback of One-Hot Encoding -

- Its not good for Higher no. of categorical value.
- It will lead to Dummy Variable Trap, Its a scenario in which variables are highly correlated to each other.
- Dummy Variable Trap leads to the problem of multicollinearity, Multicollinearity occurs where there is a dependency between the independent features.

When to use a Label Encoding vs. One Hot Encoding ?

We apply Label Encoding when-

```
1. The categorical feature is ordinal(have a sequence) (like Jr. kg, Sr.
 kg, Primary school, high school)
2. The number of categories is quite large as one-hot encoding can lead t
o high memory consumption
```

We apply One-Hot Encoding when-

```
1. The categorical feature is not ordinal (like the countries above)
2. The number of categorical features is less so one-hot encoding can be
 effectively applied
```

Co-variance matrix -

- If covariance two feature is Positive it means if one feature is large then other feature will also be larger.
- If covariance of two feature is Negative, it means if one feature is large then other feature is small.

Dimension Reduction Techniques -

Dimension Reduction refers to the process of converting a set of data having vast dimensions into data with lesser dimensions ensuring that it conveys similar information concisely.

Benefits of Dimension Reduction -

1. It helps in data compressing and reducing the storage space required
2. It fastens the time required for performing same computations. Less dimensions leads to less computing, also less dimensions can allow usage of algorithms unfit for a large number of dimensions
3. It takes care of multi-collinearity that improves the model performance. It removes redundant features. For example: there is no point in storing a value in two different units (meters and inches).

Common Methods to perform Dimension Reduction -

1. Missing Values
2. Low Variance
3. Decision Trees
4. Random Forest
5. High Correlation
6. Backward Feature Elimination
7. Principal Component Analysis (PCA)
8. Factor Analysis

1. Missing Values- While exploring data, if we encounter missing values, then our first step should be to identify the reason then impute missing values/ drop variables using appropriate methods.

- What if we have too many missing values? Should we impute missing values or drop the variables?
- We should prefer drop the variables, because it would not have lot more details about data set.

2. Low Variance- If we have a variable which have same value in all observations in our data set. In case of high number of dimensions, we should drop variables having low variance compared to others because these variables will not explain the variation in target variables.

3. Decision Trees- It can be used as a ultimate solution to tackle multiple challenges like missing values, outliers and identifying significant variables. It worked well in our Data Hackathon also. Several data scientists used decision tree and it worked well for them.

4. Random Forest- Similar to decision tree is Random Forest. We would also recommend using the in-built feature importance provided by random forests to select a smaller subset of input features.
Just be careful that random forests have a tendency to bias towards variables that have more no. of distinct values i.e. favor numeric variables over binary/categorical values.

5. High Correlation- Dimensions exhibiting higher correlation can lower down the performance of model. Moreover, it is not good to have multiple variables of similar information or variation also known as “Multicollinearity”.

We can use Pearson (continuous variables) or Polychoric (discrete variables) correlation matrix to identify the variables with high correlation and select one of them using VIF (Variance Inflation Factor). Variables having higher value (VIF > 5) can be dropped.

6. Backward Feature Elimination- In this method, we start with all n dimensions. Compute the sum of square of error (SSR) after eliminating each variable (n times). Then, identifying variables whose removal has produced the smallest increase in the SSR and removing it finally, leaving us with n-1 input features.

Repeat this process until no other variables can be dropped.
Reverse to this, we can use **Forward Feature Selection** method. In this method, we select one variable and analyse the performance of model by adding another variable. Here, selection of variable is based on higher improvement in model performance.

7. Principal Component Analysis (PCA)- In this technique, variables are transformed into a new set of variables, which are linear combination of original variables. These new set of variables are known as **principle components.**

- Principle Components are obtained in such a way that first principle component responsible for most of the possible variation of original data after which each succeeding component has the highest possible variance.
- The second principal component must be orthogonal to the first principal component. In other words, it does its best to capture the variance in the data that is not captured by the first principal component.
- The principal components are sensitive to the scale of measurement, now to fix this issue we should always standardize variables before applying PCA. Applying PCA to your data set loses its meaning. If interpretability of the results is important for your analysis, PCA is not the right technique for your project.

8. Factor Analysis- Let's say some variables are highly correlated. These variables can be grouped by their correlations i.e. all variables in a particular group can be highly correlated among themselves but have low correlation with variables of other group(s).
Here each group represents a single underlying construct or factor. These factors are small in number as compared to large number of dimensions. However, these factors are difficult to observe. There are basically two methods of performing factor analysis:

- EFA (Exploratory Factor Analysis)
- CFA (Confirmatory Factor Analysis)

Refer -

1. Feature Scaling with Code- https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/ (https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/)
2. Dimension Reduction Methods - https://www.analyticsvidhya.com/blog/2015/07/dimension-reduction-methods/ (https://www.analyticsvidhya.com/blog/2015/07/dimension-reduction-methods/)

3. Cleaning & Preprocessing - https://medium.com/analytics-vidhya/data-cleaning-and-preprocessing-a4b751f4066f (https://medium.com/analytics-vidhya/data-cleaning-and-preprocessing-a4b751f4066f)
4. How to Speed up Data Preprocessing - https://www.analyticsvidhya.com/blog/2020/09/pandas-speed-up-preprocessing/ (https://www.analyticsvidhya.com/blog/2020/09/pandas-speed-up-preprocessing/)