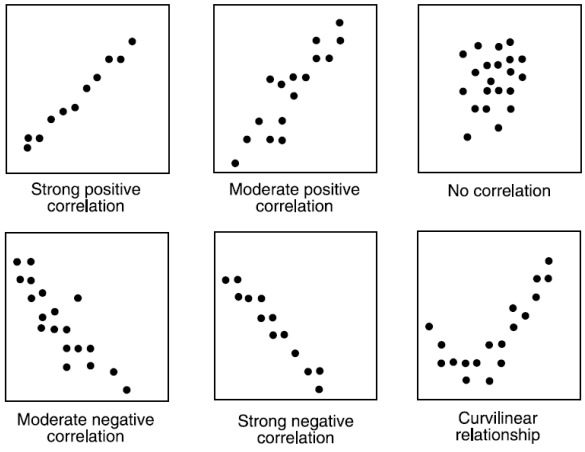
Data Analysis & Data Cleaning: Orange, Rapidminer

1. **Variable Identification:** First, identify **Predictor** (Input) and **Target** (output) variables. Next, identify the data type and category of the variables.
2. **Univariate Analysis:** Method to perform uni-variate analysis will depend on whether the variable type is categorical or continuous.
    a. **Continuous Variables:** In case of continuous variables, we need to understand the central tendency and spread of the variable. These are measured using various statistical metrics visualization methods
        i. **Central Tendency:** Min, Max, Mean, Median, Mode
    b. **Categorical Variables:-** For categorical variables, we'll use frequency table to understand distribution of each category

3. **Bi-variate Analysis:** Bi-variate Analysis finds out the relationship between two variables.
    a. Continuous & Continuous: While doing bi-variate analysis between two continuous variables, we should look at scatter plot. It is a nifty way to find out the relationship between two variables. The pattern of scatter plot indicates the relationship between variables. The relationship can be linear or non-linear.



Scatter plot shows the relationship between two variable but does not indicates the strength of relationship amongst them. To find the strength of the relationship, we use Correlation. Correlation varies between -1 and +1.

i. -1: perfect negative linear correlation
ii. +1:perfect positive linear correlation
iii. 0: No correlation

b. Categorical & Categorical:

i. Chi-Square Test: This test is used to derive the statistical significance of relationship between the variables. Also, it tests whether the evidence in the sample is strong enough to generalize that the relationship for a larger

population as well. Chi-square is based on the difference between the expected and observed frequencies in one or more categories in the two-way table. It returns probability for the computed chi-square distribution with the degree of freedom.

- Probability of 0: It indicates that both categorical variable are dependent
- Probability of 1: It shows that both variables are independent.
- Probability less than 0.05: It indicates that the relationship between the variables is significant at 95% confidence.

ii. Statistical Measures used to analyze the power of relationship are:

- Cramer's V for Nominal Categorical Variable
- Mantel-Haenszed Chi-Square for ordinal categorical variable.

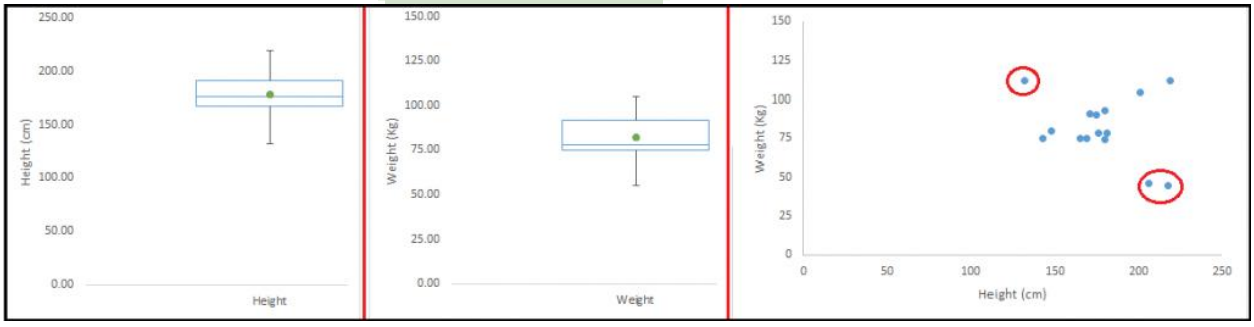
c. Categorical & Continuous:

i. Z-Test/ T-Test: Either test assess whether mean of two groups are statistically different from each other or not. If the probability of Z is small then the difference of two averages is more significant. The T-test is very similar to Z-test but it is used when number of observation for both categories is less than 30.

ii. ANOVA:- It assesses whether the average of more than two groups is statistically different.

4. Outliers: Outlier can be of two types: **Univariate** and **Multivariate.** These outliers can be found when we look at distribution of a single variable. Multi-variate outliers are outliers in an n-dimensional space. In order to find them, you have to look at distributions in multi-dimensions.

Let us understand this with an example. Let us say we understand the relationship between height and weight. Below, we have univariate and bivariate distribution for Height, Weight. Take a look at the box plot. We do not have any outlier (above and below 1.5*IQR, most common method). Now look at the scatter plot. Here, we have two values below and one above the average in a specific segment of weight and height.



a. Types of outliers:

- **Data Entry Errors:-** Human errors such as errors caused during data collection, recording, or entry can cause outliers in data. For example: Annual income of a customer is $100,000. Accidentally, the data entry operator puts an additional zero

in the figure. Now the income becomes $1,000,000 which is 10 times higher. Evidently, this will be the outlier value when compared with rest of the population.

- **Measurement Error:** It is the most common source of outliers. This is caused when the measurement instrument used turns out to be faulty. For example: There are 10 weighing machines. 9 of them are correct, 1 is faulty. Weight measured by people on the faulty machine will be higher / lower than the rest of people in the group. The weights measured on faulty machine can lead to outliers.
- **Experimental Error:** Another cause of outliers is experimental error. For example: In a 100m sprint of 7 runners, one runner missed out on concentrating on the 'Go' call which caused him to start late. Hence, this caused the runner's run time to be more than other runners. His total run time can be an outlier.
- **Intentional Outlier*:*** This is commonly found in self-reported measures that involves sensitive data. For example: Teens would typically under report the amount of alcohol that they consume. Only a fraction of them would report actual value. Here actual values might look like outliers because rest of the teens are under reporting the consumption.
- **Data Processing Error:** Whenever we perform data mining, we extract data from multiple sources. It is possible that some manipulation or extraction errors may lead to outliers in the dataset.
- **Sampling error:** For instance, we have to measure the height of athletes. By mistake, we include a few basketball players in the sample. This inclusion is likely to cause outliers in the dataset.
- **Natural Outlier:** When an outlier is not artificial (due to error), it is a natural outlier. For instance: In my last assignment with one of the renowned insurance company, I noticed that the performance of top 50 financial advisors was far higher than rest of the population. Surprisingly, it was not due to any error. Hence, whenever we perform any data mining activity with advisors, we used to treat this segment separately.

b. **What is the impact of Outliers on a dataset?** Outliers can drastically change the results of the data analysis and statistical modeling. There are numerous unfavorable impacts of outliers in the data set:

- It increases the error variance and reduces the power of statistical tests
- If the outliers are non-randomly distributed, they can decrease normality
- They can bias or influence estimates that may be of substantive interest
- They can also impact the basic assumption of Regression, ANOVA and other statistical model assumptions.

c. Outlier Detection

i. **Visualization method**: Most commonly used method to detect outliers is visualization. We use various visualization methods, like **Box-plot**, **Histogram**, **Scatter Plot**

- Any value, which is beyond the range of -1.5 x IQR to 1.5 x IQR
- Use capping methods. Any value which out of range of 5th and 95th percentile can be considered as outlier
- Data points, three or more standard deviation away from mean are considered outlier
- Outlier detection is merely a special case of the examination of data for influential data points and it also depends on the business understanding
- Bivariate and multivariate outliers are typically measured using either an index of influence or leverage, or distance. Popular indices such as Mahalanobis' distance and Cook's *D* are frequently used to detect outliers.

i. **Z-Score :** The z-score or standard score of an observation is a metric that indicates how many standard deviations a data point is from the sample's mean, assuming a gaussian distribution. This makes z-score a parametric method. Works well in low dimensional feature space.

ii. **Dbscan (Density Based Spatial Clustering of Applications with Noise)**: Dbscan is a density based clustering algorithm. The complexity of dbscan is of O(*n* log *n*), it and effective method with medium sized data sets. Outliers (noise) will be assigned to the -1 cluster. After tagging those instances, they can be removed or analyzed. Works well if the feature space for searching outliers is multidimensional (ie. 3 or more dimensions). The values in the feature space need to be scaled accordingly

```
from sklearn.cluster import DBSCAN
seed(1)
random_data = np.random.randn(50000,2)  * 20 + 20

outlier_detection = DBSCAN(min_samples = 2, eps = 3)
clusters = outlier_detection.fit_predict(random_data)
list(clusters).count(-1)
```

iii. **Isolation Forests:** It is a relatively novel method based on binary decision trees. There is no need of scaling the values in the feature space. If not correctly optimized, training time can be very long and computationally expensive. Below code will output the predictions for each data point in an array. If the result is -1, it means that this specific data point is an outlier. If the result is 1, then it means that the data point is not an outlier

```
import numpy as np

np.random.seed(1)

random_data = np.random.randn(50000,2) * 20 + 20

clf = IsolationForest( behaviour = 'new', max_samples=100,
random_state = 1, contamination= 'auto')

preds = clf.fit_predict(random_data)
```

iv. **Projection Methods:** Use projection methods to summarize your data to two dimensions (such as PCA, SOM or Sammon's mapping). Visualize the mapping and identify outliers by hand

v. **Local Outlier Factor:** Another efficient way to perform outlier detection on moderately high dimensional datasets is to use the Local Outlier Factor (LOF) algorithm.The neighbors.LocalOutlierFactor (LOF) algorithm computes a score (called local outlier factor) reflecting the degree of abnormality of the observations. It measures the local density deviation of a given data point with respect to its neighbors. The idea is to detect the samples that have a substantially lower density than their neighbors. In practice the local density is obtained from the k-nearest neighbors. https://scikit-learn.org/stable/modules/outlier_detection.html

vi. **One-class SVM for novelty detection: Does not work well for outlier detection.** One-class SVM is an unsupervised algorithm that learns a decision function for **novelty detection**: classifying new data as similar or different to the training set. https://scikit-learn.org/stable/auto_examples/svm/plot_oneclass.html

vii. **Optional : Python Outlier Detection** toolbox (**PyOD**)

Github: https://github.com/yzhao062/Pyod
PyPI: https://pypi.org/project/pyod/
Documentation: https://pyod.readthedocs.io
Interactive Jupyter Notebooks: https://mybinder.org/v2/gh/yzhao062/Pyod/master

https://www.analyticsvidhya.com/blog/2019/02/outlier-detection-python-pyod/

d. Outlier Handling

i. **Robust Algorithm:** Explore your data both with and without outliers. If you decide that you need them for your ML model then select a method that will be robust enough to handle them. Tree-based models are generally not as affected by outliers, while regression-based models are. If you're performing a statistical test, try a non-parametric test instead of a parametric one.

ii. **Use a more robust error metric**: Switching from mean squared error to **mean absolute difference** (or something **like Huber Loss**) reduces the influence of outliers. Huber Loss suggests that loss function should be quadratic (Squared Error) for smaller deviations, and should be linear (absolute error) for large deviations (outliers).

iii. **Delete Outliers:** If you find that those outliers are really out there and not useful for getting the big picture information and modeling of your data, then it's best to just drop them

iv. **Transforming and binning values:** If outliers are inseparable part of data. Transforming variables can also eliminate outliers. Natural log of a value

reduces the variation caused by extreme values. Binning is also a form of variable transformation. Decision Tree algorithm allows to deal with outliers well due to binning of variable. We can also use the process of assigning weights to different observations.

v. **Winsorize your data.** Artificially cap your data at some threshold
vi. **Treat separately:** If there is significant number of outliers, we should treat them separately in the statistical model. One of the approaches is to treat both groups as two different groups and build individual model for both groups and then combine the output.

5. Removing unwanted columns

a. Multi Collinear Columns
b. Unrelated columns: Not all features are equal. Some things you might not even need! For example, maybe you're looking at a dataset of books purchased from Amazon over the past year where one of the feature variables is called "font-type" indicating the type of font used in the book. This is pretty irrelevant to predicting the sales of a book! You can probably safely drop this feature all together.
c. Redundant information (one column already defined or included in other column)
d. Has only one value
e. Has almost unique values for all records

6. Clubbing two or more columns

7. Check for invalid data in columns

a. **Bad data** means any data points or values that shouldn't be there or are just plain wrong. For example, suppose one of your feature variables is called "gender" where most of the values are "male" or "female". But then as you're skimming through your dataset you notice that there are a couple of data points that have the value 67.3 for gender! Clearly 67.3 doesn't mean anything in the context of that variable. Moreover, if you try to convert the "gender" feature variable to categorical floats: male = 0.0 and female = 1.0, you'a have an extra one: 67.3 = 2.0!
b. Standardization: All of your data in each feature variable should be in the same standardized format. It'll make the technical aspects of your data exploration and modeling much easier. For example, let's take the example of the "gender" variable again with values "male" or "female". If the data was collected by a human you might get many different values which you did not expect:
    i. male, female (this one's good)
    ii. MALE, FEMALE (entered with caps lock on)
    iii. Male, Female (some people will capitalise)
    iv. Make, Femall (typos!)
c. Get rid of extra space in column values
d. Check formatting of data i.e. Number values are represented as text

8. **Handling missing values in columns:** Missing data in the training data set can reduce the power / fit of a model or can lead to a biased model because we have not analysed the behavior and relationship with other variables correctly. It can lead to wrong prediction or classification.
    a. Reason for missing values:
        i. **Data Extraction:** It is possible that there are problems with extraction process.
        ii. **Data collection**:

- **Missing completely at random:** This is a case when the probability of missing variable is same for all observations. For example: respondents of data collection process decide that they will declare their earning after tossing a fair coin. If an head occurs, respondent declares his / her earnings & vice versa. Here each observation has equal chance of missing value.
- **Missing at random:** This is a case when variable is missing at random and missing ratio varies for different values / level of other input variables. For example: We are collecting data for age and female has higher missing value compare to male.
- **Missing that depends on unobserved predictors:** This is a case when the missing values are not random and are related to the unobserved input variable. For example: In a medical study, if a particular diagnostic causes discomfort, then there is higher chance of drop out from the study. This missing value is not at random unless we have included "discomfort" as an input variable for all patients.
- **Missing that depends on the missing value itself:** This is a case when the probability of missing value is directly correlated with missing value itself. For example: People with higher or lower income are likely to provide non-response to their earning.

1.

    b. Categorical
    c. Numerical
    d. Interpolation \ Extrapolation
    e. Drop
        i. **Drop a feature:** `df.drop('feature_variable_name', axis=1)` If you find that a certain feature variables has > 90% of NaN values in the dataset, it makes sense to just drop the whole thing from your data.
    f. Impute
        i. mean, median (Mean is most useful when the original data is not skewed, while the median is more robust, not sensitive to outliers, and thus used when data is skewed.)
        ii. **linear regression**. Based on the existing data, one can calculate the best fit line between two variables, say, house price vs. size m²

iii. **Hot-deck (**Forward filling \ Backward filling): Copying values from other similar records. This is only useful if you have enough available data. And, it can be applied to numerical and categorical data.

iv. Model based imputation —

1. Linear regression:
2. k-nearest neighbors :Which classifies similar records and put them together, can also be utilized. A missing value is then filled out by finding first the k records closest to the record with missing values. Next, a value is chosen from (or computed out of) the k nearest neighbours

- **Advantages:**
  - k-nearest neighbour can predict both qualitative & quantitative attributes
  - Creation of predictive model for each attribute with missing data is not required
  - Attributes with multiple missing values can be easily treated
  - Correlation structure of the data is taken into consideration
- **Disadvantage:**
  - KNN algorithm is very time-consuming in analyzing large database. It searches through all the dataset looking for the most similar instances.
  - Choice of k-value is very critical. Higher value of k would include attributes which are significantly different from what we need whereas lower value of k implies missing out of significant attributes.

3.

g. Flag :

i. Filling in the missing values leads to a loss in information, no matter what imputation method we used. That's because saying that the data is missing is informative in itself, and the algorithm should know about it.

ii. Otherwise, we're just reinforcing the pattern already exist by other features. This is particularly important when the missing data doesn't happen at random. Take for example a conducted survey where most people from a specific race refuse to answer a certain question.

iii. Missing numeric data can be filled in with say, 0, but has these zeros must be ignored when calculating any statistical value or plotting the distribution.

iv. While categorical data can be filled in with say, "Missing": A new category which tells that this piece of data is missing.

9. Augmentation of data : If less data available then how to augment the data (Computer vision)

Feature Engineering: Feature engineering is the science (and art) of extracting more information from existing data. You are not adding any new data here, but you are actually making the data you already have more useful.

For example, let's say you are trying to predict foot fall in a shopping mall based on dates. If you try and use the dates directly, you may not be able to extract meaningful insights from the data. This is because the foot fall is less affected by the day of the month than it is by the day of the week. Now this information about day of week is implicit in your data. You need to bring it out to make your model better.

Feature engineering itself can be divided in 2 steps:

- Variable transformation.
- Variable / Feature creation.

1. **Variable transformation:**
    a. **Feature Scaling**
        - **Standardization:**
            - Standardization is a popular feature scaling method, which gives data the property of a standard normal distribution (also known as Gaussian distribution).
            - We don't need to do this process manually as sklearn provides a function called **StandardScaler.**
            - In simple terms we just calculate the mean and standard deviation of the values and then for each data point we just subtract the mean and divide it by standard deviation.
        - **Normalization**
            - Normalization refers to the rescaling of data features between 0 and 1, which is a special case of Min-Max scaling
            - The ML library scikit-learn has a MinMaxScaler class for normalization.
        - The following table shows the difference between standardization and normalization for a sample dataset with values from 1 to 5

input	standardized	normalized
0.0	-1.336306	0.0
1.0	-0.801784	0.2
2.0	-0.267261	0.4
3.0	0.267261	0.6
4.0	0.801784	0.8
5.0	1.336306	1.0

- **Logarithm:** Log of a variable is a common transformation method used to change the shape of distribution of the variable on a distribution plot**. It is generally used for reducing right skewness of variables**. Though, It can't be applied to zero or negative values as well.
- **Box-Cox transformation:** The Box-Cox transformation is a generalized "power transformation" that transforms data to make the distribution more normal. For example, when its lambda parameter is 0, it's equivalent to the log-transformation. It's used to stabilize the variance (eliminate heteroskedasticity) and normalize the distribution.
- **Square / Cube root:** The square and cube root of a variable has a sound effect on variable distribution. However, it is not as significant as logarithmic transformation. Cube root has its own advantage. It can be applied to negative values including zero. Square root can be applied to positive values including zero.
- **Binning:** It is used to categorize variables. It is performed on original values, percentile or frequency. Decision of categorization technique is based on business understanding. For example, we can categorize income in three categories, namely: High, Average and Low. We can also perform co-variate binning which depends on the value of more than one variables.

b. **Feature Selection:** It's best for your data collection pipeline to start with only one or two features. This will help you confirm that the ML model works as intended. Also, when you build a baseline from a couple of features, you'll feel like you're making progress!

c. **Feature Extraction**

- **Decomposition**: There may be features that represent a complex concept that may be more useful to a machine learning method when split into the constituent parts. An example is a date that may have day and time components

that in turn could be split out further. Perhaps only the hour of day is relevant to the problem being solved. consider what feature decompositions you can perform… For example, values for the gender groups that stored as 'm014', 'f014', 'm1528', 'f1528'. You cannot tell it's completely wrong, but it would be better to split these values into 'gender' and 'age' columns.

- **Aggregation**: There may be features that can be aggregated into a single feature that would be more meaningful to the problem you are trying to solve. For example, there may be a data instances for each time a customer logged into a system that could be aggregated into a count for the number of logins allowing the additional instances to be discarded. Consider what type of feature aggregations could perform.
- For example, you're still casually doing your job at the travel agency, **They asked you to create a model to prioritize your clients using clustering.**
  You have 3 features in your dataset:
  - Ticket request date: The date a person requests their ticket(s).
  - Departure date: The date a person wants to travel on.
  - Return date: The date of returning.

  You know you can't feed your model non-numerical data, still, you want to include them.
  Some of the insights the data scientist in you can observe:
  Stay duration: Difference between departure day and return date. That's the duration they're staying.
  Request duration: Difference between the arrival date and the request date.
  Okay, that's good.

2. ## Feature Addition
   a. **Creating dummy variables:** One of the most common application of dummy variable is to convert categorical variable into numerical variables. Dummy variables are also called Indicator Variables. It is useful to take categorical variable as a predictor in statistical models.
3. ## Feature Filtering
4. ## Feature Selection
5. ## Handling Categorical Columns
   a. Ordinal Values
   b. Nominal values
6. ## Skewed data or Imbalanced data set
   a. **Random Downsampling**: An effective way to handle imbalanced data is to downsample and upweight the majority class. Let's start by defining those two new terms:

- **Downsampling** (in this context) means training on a disproportionately low subset of the majority class examples.
- **Upweighting** means adding an example weight to the downsampled class equal to the factor by which you downsampled (**"Balanced" class_weight hyper-parameter in Logistic regression)**

b. **Random Over-Sampling**
c. **Cluster-Based Over Sampling**
d. **Informed Over Sampling: Synthetic Minority Over-sampling Technique (SMOTE):** This technique is followed to avoid overfitting which occurs when exact replicas of minority instances are added to the main dataset. A subset of data is taken from the minority class as an example and then new synthetic similar instances are created
e. **Modified synthetic minority oversampling technique (MSMOTE):** It is a modified version of SMOTE. SMOTE does not consider the underlying distribution of the minority class and latent noises in the dataset. To improve the performance of SMOTE a modified method MSMOTE is used.
f. **EnSemble with Bagging**
g. **Gradient Boosting/ADA Boost/ ADASYN/XG Boost/**
h. For better results, one can use synthetic sampling methods like SMOTE and MSMOTE along with advanced boosting methods like Gradient boosting and XG Boost.
i. **Reference link from Analytics Vidya**
j. **https://imbalanced-learn.readthedocs.io/en/stable/**

7. **Dimensionality Reduction**
   a. **PCA**

Splitting the data into training & testing sets

Selecting Algorithm

1. **Regression**

   a. Linear Regression
   b. Support Vector Regression
   c. Decision Tress/Random Forest
   d. Ensemble Methods

2. **Classification**

a. K-Nearest Neighbor
b. Naive Bayes
c. Decision Trees/Random Forest
d. Support Vector Machine
e. Logistic Regression

Randomforest works very well on a data set which contains both numerical and categorical variables. This is because it is a supervised nonlinear classification ML algorithm. Data transformation is not usually required in Randomforest since it is nonlinear, it does not give weightage to a single feature, and treats every feature equally. On the other hand, SVM is a linear ML algorithm. It takes into the consideration the distance between the points at a particular space and time. This method fluctuates with the change in values — the higher the value the higher is distance — which is why data scaling is necessary here. In both the algorithms, conversion of the categorical values to numerical features is required because the machine cannot understand how to read different categories.

3. **Unsupervised Learning:**

a. Gaussian mixtures
b. K-Means Clustering
c. Boosting
d. Hierarchical Clustering
e. K-Means Clustering
f. Spectral Clustering

Training

Cross Validation

Testing

Evaluation/Accuracy of Model

Improving ML Model

Prediction