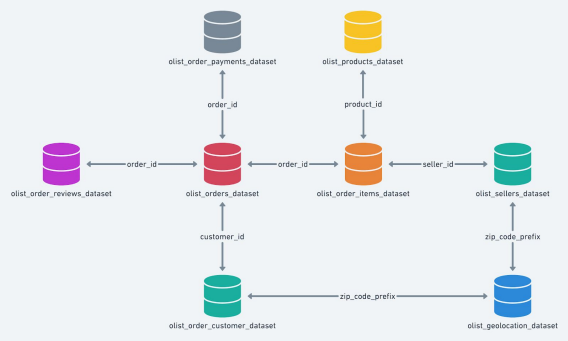
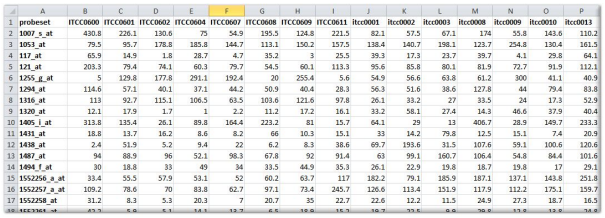
Data Science Project - Flow graph

1. Data preparation

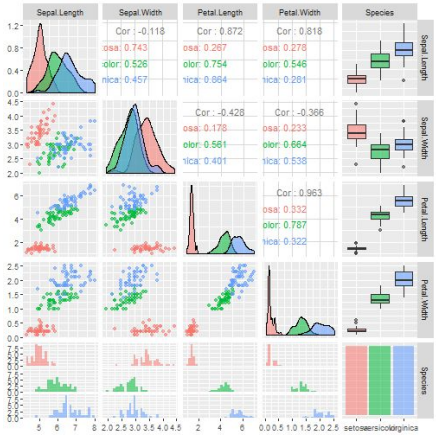


2. Flat-file generation

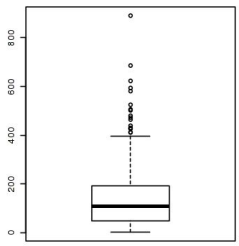
Data enrichment and transformation



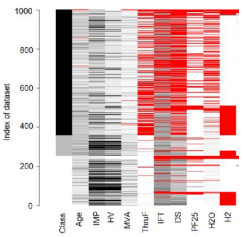
3. **Exploratory Data Analysis (EDA)**



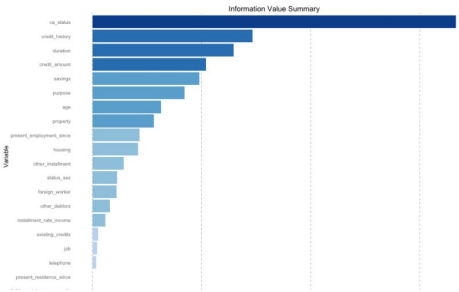
4. Outlier detection and treatment



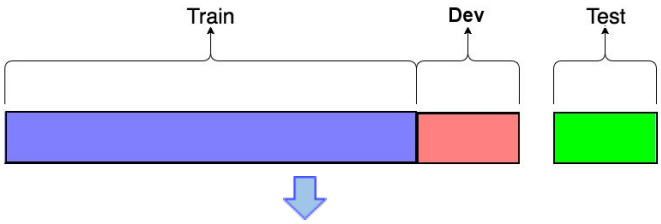
5. Missing values Imputation



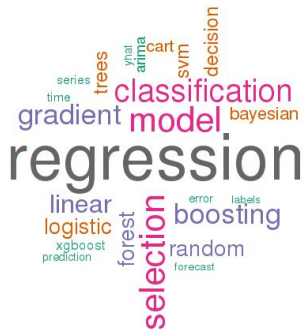
6. **Variable Selection**



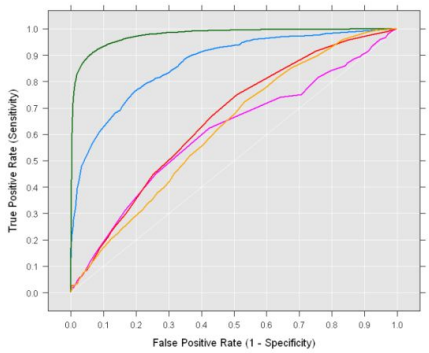
7. **Train-Dev-Test preparation**



8.Model Selection



9. **Model Fine-tuning**



Data Science Project - Flow graph

I. Data enrichment and transformation

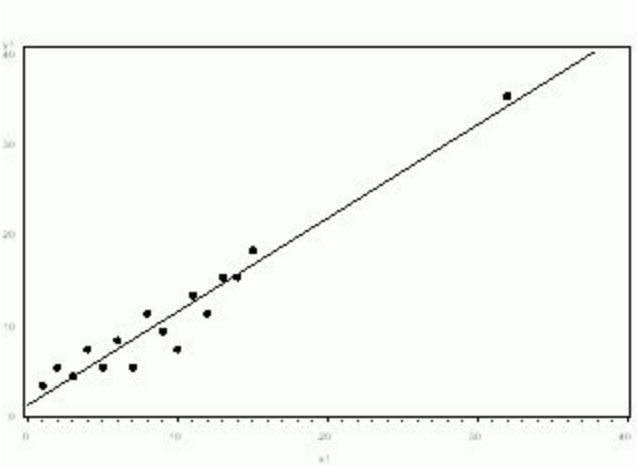
1. Combination of two or more variables: sum/difference/multiplication,division
2. Variable modification: Polynomial, Logaritm, Square root, exponencial
3. Enrich with cluster analysis
4. Enrich with external data

II. Exploratory Data Analysis (EDA)

1. Statistical Analysis:
    - Mean and Standard Deviation
    - Median and IQR (25%-75%)
    - Minimum, Maximum, Count
2. Table One
3. Graphics:
    - Categorical variables: barplot
    - Numeric Variables: histogram (distplot)
    - Numeric vs Categorical: boxplot
    - Numeric vs Numeric: scatter plot
    - Pairs

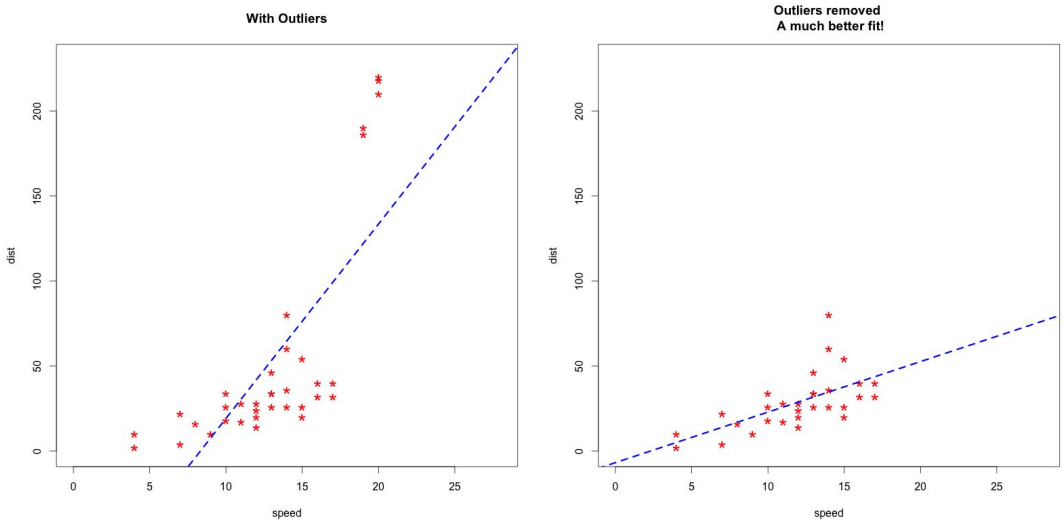
III. Data cleansing

1. Outlier detection - univariate:
    - Normal distribution: Standard Deviation (z-scores)
    - Non-normal distribution: IQR
2. Outlier detection - multivariate:
    - DBSCAN
3. Outlier treatment:
    - Obviously incorrect: convert value to NA
        - Example: age 450, body temperature 2°C, size -1.2
    - Does change the distribution but don't change relationships: convert value to NA and **report**.

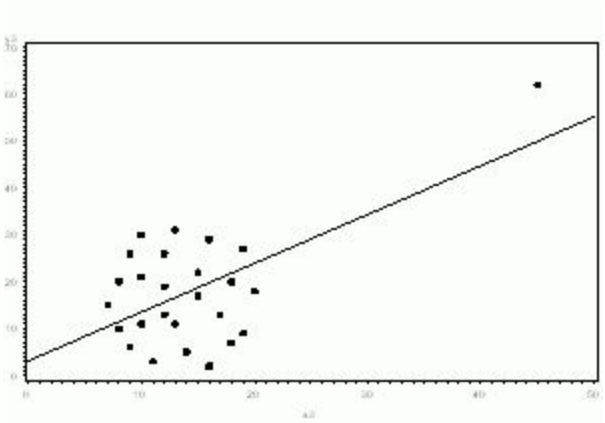


Data Science Project - Flow graph

- Does change both the distribution and the relationships: make analysis of the effect or influence in the presence and absence of the outliers. Report the findings and explain your decision of leaving or substituting the outliers with NAs.



- Makes a correlation where there is no correlation: convert value to NA



- Multivariate outliers: DBSCAN returns a list of rows containing mv-outliers. Add an variable indicator for outliers (0/1). Check if this indicator has a relation with the dependent variable (y or outcome). You can use correlation analysis or check the distribution of y between outlier and non outlier cases. You can drop it if there is no correlation or if the distribution is similar. If not, leave the outlier indicator in the dataset.

4. Missing values: Determinate the missingness mechanism.
    - *Missing Completely at Random (MCAR)*: missing values are generated randomly and there is no possible way to explain why those values are missing.
    - *Missing at Random (MAR)*: There is a variable that influence in the generation of missing values, but within subgroups of such a variable the missingness is completely at random. For example, women tend less to give write their age or weight than men. However, if we check the mechanism by separating women from men, there generative mechanism is MCAR.

- *Missing not at Random (MNAR):* There is a clear cause for the generation of missing values. An example is the visits to gynecologist (women's doctors). No men will have a visit in the dataset. Another example is a question made only to individuals with age 60 and more. No individual with 55 years old will have the question fulfilled.

5. Missingness treatment: If the missing mechanism is MCAR or MAR, we can use imputation techniques, otherwise we have to decide if dropping the rows or dropping the columns.

- Detecting Missingness Mechanism
    - For each variable (feature) having missing values, generate a dummy variable indicating for each value if it is full (0) or is missing (1).
    - Using only the variables which have no missing data, run a logistic regression (in R: glm; in Python: OLS) where the X is the dataset with only those variables and the outcome (y) is the missing dummy variable.
    - If all the p-values for the coefficients (betas) are non-significant (p>=0.05), we can assume that the mechanism is MCAR.
    - If there is one variable that was significant (p-value < 0.05) then we have to inspect the relationship between this variable and the missing indicator. We can check it using a boxplot and determinate a value which can divide the dataset into two subsets. Each subset will be analyzed separately as we did before. If in both groups we can demonstrate that the mechanism is MCAR, we say that for this variable original mechanism is MAR.
    - In the case we can't demonstrate MCAR or MAR, we say that the mechanism is MNAR.
- Imputation techniques:
    - Statistical imputation: using mean, median or mode (depending the data scale). **Not recommended**!
    - Model based imputation: we can use a predictive model to impute the missing values
        - KNN: this algorithm is very popular because it easily and fast impute data. We divide the dataset into complete data (including the variable that we want to impute) and train the model. Then we use the predict function to complete the data in the incomplete subset.
        - Random Forest (use as we explained with KNN)
        - Decisión trees
    - Multiple imputation: The most common method is the Multiple Imputation by Chained Equations (MICE) which uses all the dataset for imputation. It begins with the variable with less missing values and impute it, Then uses the second with less missings and impute it, and so on. MICE generate a number of imputed datasets (default or most common is 5), that has different values for each imputed variables. The mean, median, standard deviation and IRQ for each variable on each imputed dataset are constant. Any further analysis must be made using all the imputed datasets (5 different analysis each time, the final result is the calculated as the mean of the outcome (y) of all the models.
    - When having lot of data (>20,000 rows), a simple model is recommended. With smaller datasets, multiple imputation is recommended.

Data Science Project - Flow graph

IV. Feature Selection

1. Univariate Analysis: Each variable on the dataset is analyzed by comparing its relationship with the dependent variable (outcome or y). The analysis depends on the independent (x) and dependent (y) data types:

- If x is nominal and y is nominal or ordinal: use chi-square
- If x is nominal and y is continuous: use spearman correlation
- If x is continuous and y is binomial (0/1, yes/no, true/false): use independent t-test
- If x is continuous and y is multinomial (more than two categories) or ordinal: use anova
- If both x and y are continuous, use pearson or spearman correlation

In case that y is nominal or ordinal with less than 6 categories, you can use tableone (R and Python) setting the y as the grouping category. Tableone makes the whole analysis automatically. You will want to include in your analysis those variables with a significant p-value (< 0.05).

2. Multivariate analysis: Using the whole dataset and running predictive models that are able to return a list recommended features by defining their influence in the model. For this step you don't need to have to partition the dataset into train, dev and test.

- *LASSO* (L1 penalization): Lasso penalizes the growing of the values of the coefficients (betas) and can get them down to zero. Those variables with a zeroed coefficients are excluded from the analysis, giving to this algorithm the capacity of feature selection.
- Random Forest: this algorithm is able to generate a list of the most influential variables and their lift.
- Gradient Descent: same as Random forest.
- Support Vector Machine (SVM): It can be used for feature selection by using L1 penalization.
- Principal Component Analysis (PCA): can be used for the selection of variables comparing the variables with the highest correlation with the principal components that catch the much of the variability (80% cumulative variance).

3. Selection based on voting: using many of the techniques (univariate and multivariate), we make a table with all the variables on the dataset and indicate the recommended variables for each technique, then we select a threshold for the total votings and on this basis we select the variables that will be used to train our models.

Variable	Univarable	Lasso	RandomForest	GradientBoost	SVM	Sum
Fz_N100	1	1	1	1	1	5
FCz_N100	1	0	1	0	1	3
Cz_N100	1	1	1	0	1	4
FC4_N100	1	1	1	1	1	5
FC3_B0	1	0	0	0	0	1
FC4_B0	1	0	0	0	0	1
C3_B0	0	0	0	0	0	0
C4_B0	0	0	0	0	0	0
FC4_B1	0	1	0	0	1	2
FC3_B1	1	1	0	1	1	4