Credit Risk Model – PD based on NN

**Remarks:**

* Segmentation of clients? E.g. working and non-working (income variable)
* Add a field by calculating using existing?
* Reality check
* Index, Risk Attributes, Dependend variable
* Use defaults in 2020 as outliers because pandemic extraordinary  
  🡪 try with and without and look at results  
  🡪 or weight other years more strongly  
  🡪 in practice you have to include it bcs regulations and etc.
* ~~Before replacing~~ **~~nan~~** ~~look at value distributions and for each risk attribute decides separately~~
* **~~Non-numeric~~** ~~data: replace with max?~~
* ~~For each factor: distribution plot & boxplot for~~ **~~outlier detection~~** ~~(5% & 95%)~~
* ~~Reduce sensitivity of model to outliers by~~ **~~removing or winsorizating outliers for factors~~**
* **~~Standardize data or binning approach~~**
* Risk factor values over distribution quantiles & Prob of default over different distribution quantiles
* Sensitivity analysis

**Presentation:**

* Ppt: show **correlation matrix** & explain why logic the high correlations
* Limitations and challenges
  + Highly parameterized 🡪 no generalization

## Data Analysis

* Summary / Descriptive Statistics
* Plotting (Distribution)

## Data Cleaning

* Missing and wrong values:
  + Paper: often in NN deleted 🡪 loss of info
  + Paper: replaced by:
    - Mean (if value is missing)
    - Max (if computation error bcs e.g. 1/0 -> )
    - Try different methods and look at effect of results
    - Discuss implications and assumptions at the end \*
* Erasing useless fields:
  + Paper: If train/test split 70/30  
    🡪 don’t use factors having >= 30% missing values
  + Drop: title (purpose is the categorized title)
* Outliers:
  + Try with and without removing outliers (e.g. those being above a percentile)
  + 5% & 95%
  + Removing or winsorizing outliers to reduce model sensitivity
* Data Normalization: \*
  + Paper: logarithmic formula to [0,1]  
    (not Min-Max transformation to [0,1] bcs outliers)
  + Normalization needed bcs values differ storngly

## Correlation Analysis

* Remove most strongly correlated factors using:
  + Correlation matrix?
  + PCA (Normalization & Standardization?)

## Train/Test Split

* Want to have balanced train/test split
* Paper: reserves default/non-default ration in both
* P.143: down- & up-sampling (bootstrap)

## Neuronal Network

p.146

pipe = Pipeline(scaler, pca, NN)  
param\_grid = {different parameters   
 and multiple values to test}  
grid = GridSearchCV()  
grid.fit()

* Develop model
* Parameter tuning \*  
  🡪 p.144 hyperparameter fine tuning
* Try often and select model with best results

## Model Evaluation

* Model accuracy (MSE and score: p.45)
* Confusion matrix
* Performance metrics:
  + Score
  + Error Rate
  + Specificity
  + Sensitivity (Recall)
  + Precision
  + PRC Curve (Precision/Recall)
  + ROC Curve (Sensitivity/Specificity)

## Robustness Check

* Network behavior w.r.t. different train/test partitions
* Prediction on different data sets
* Comparison to simpler regression models: linear regression
  + Show coefficients & ROC curves