Anjir Team

Credit Default Risk

Predicting how likely each applicant is of repaying a loan?

Goal, identify if a new client shows a high risk for loan default.

How can this help?

Reduce Uncertainty

Proportional Disbursement

Risk Reduction

Doesn’t leave business on the table!

Cleaning steps

Since the ML model can't inherently deal with text, the data must be converted to appropriate numbers. Any significant distortion/noise in the model must be removed as much as possible.

- Converting string categorical columns into numerical – Label encoding.
- Converting string categorical columns into numerical and adding new columns to indicate the presence of categorical variables – One hot encoding.
- Replacing illogical outliers with empty values (NAN values).
- Imputing empty cells with the median of the values. In some cases, imputation is approached with a certain grouping.
- Dealing with a few anomalies.
- Changing invalid entries into valid entries.

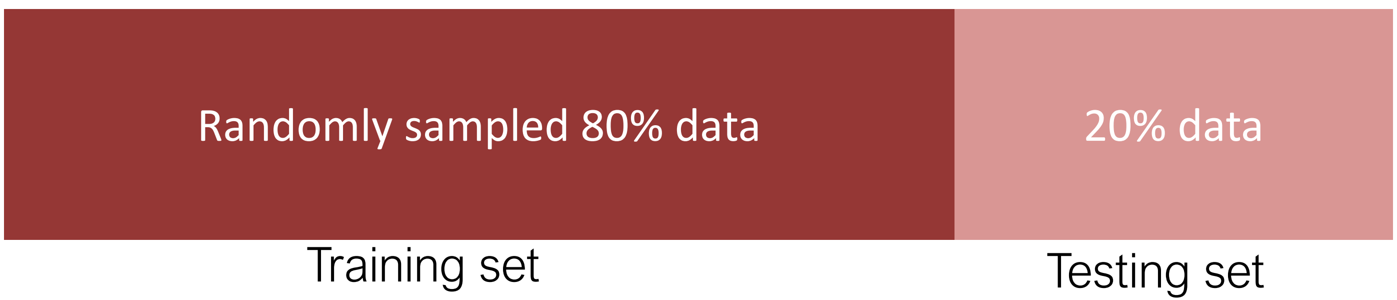
Predictive Modeling – Outcome of the model is expected to identify the potential that someone will default on a loan.

Expected Target Outcome: 0 or 1, 0 – Not a defaulter, 1 – potential defaulter.

Perfomance Metrics used : Accuracy.

Models currently used : Logistic regression, Random forest, XGBoost.

Training and Testing datasets were subjected to the same feature engineering to evaluate the model.



- Out of the main training dataset, a certain percentage is kept untrained to test the model’s performance.
- Training set and validation set are split in following percentages: 80% : 20%.
- On the testing set, the target labels are hidden, until the performance is evaluated.

Currently there are 3 models used.

Model Name	Hyperparameters	Accuracy
LogisticRegression	(C=2, class_weight=None, dual=False, fit_intercept=True, intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1, penalty='l2', random_state=None, solver='liblinear', tol=0.0001, verbose=0, warm_start=False)	74%
RandomForestClassifier	(bootstrap=True, class_weight=None, criterion='gini', max_depth=None, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, n_estimators=500, n_jobs=-1, oob_score=False, random_state=50, verbose=1, warm_start=False)	94%
XGBClassifier	(base_score=0.5, colsample_bylevel=1, colsample_bytree=1, gamma=0, learning_rate=0.1, max_delta_step=0, max_depth=5, min_child_weight=1, missing=None, n_estimators=250, nthread=-1, objective='binary:logistic', reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=0, silent=True, subsample=1)	90%

More Explanation in Notebook.

Random Forest is currently the best chosen model, Until further work.

- More feature engineering could be done, such as mathematical transformation, possibly: certain columns to log and such.
- Advanced techniques like SMOTE could be deployed to handle the class imbalance problems.
- DNN’s can be used.
- Long short term memory networks could be used to incorporate time series data.
- Advanced forms of stacking can be used apart from voting.
- Recursive feature selection can be used to reduce the features.