
# Home Credit Default Risk

Predicting how capable each applicant is of repaying a loan?

# Agenda

- Business problem
- Approach
- Data / Data wrangling
- Exploratory data analysis
- Predictive modeling
- Conclusion
- Future work

# Many people struggle to get loans due to insufficient or non-existent credit histories.

**This population is often taken advantage by untrustworthy lenders.**

**Home credit tries to include the unbanked population to support their economic needs.**

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

## How can this help?

Reduce Uncertainty

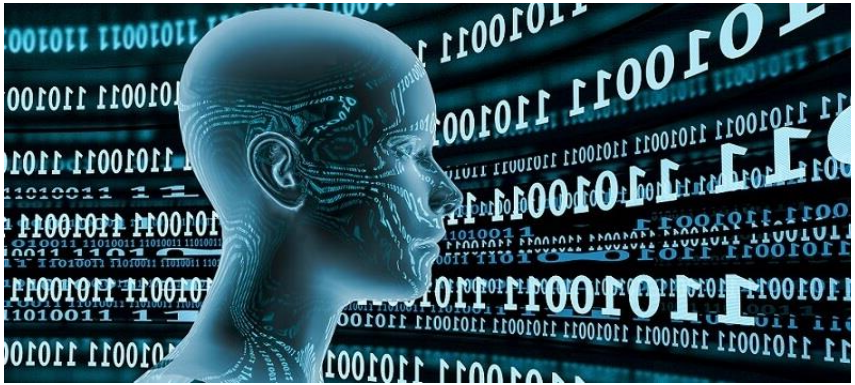
Proportional Disbursement

Risk Reduction

**Doesn't leave business on the table!**

# Uses supervised machine learning to classify one of two categories.

Supervised machine learning is a phenomenon where the AI learns from the data without anyone writing explicit code logic.



## Target Labels

**0**

**Target label of 0 indicates that there was no difficulty in repaying the loan on time.**

**1**

Target label of 1 indicates that there was difficulty in repaying the loan on time.

Image courtesy : https://www.expertsystem.com/machine-learning-definition/

# The data is taken from Kaggle's competition, publicly available.

## application_train.csv

- 307511 Records, 122 Columns
- Imbalanced Target Labels.
- Source for training machine Learning models.
- Target Labels :
- 0's – 282686, 1's - 24825

## application_test.csv

- 48744 Records, 121 Columns
- No Target Labels.
- Source for testing the performance of machine Learning models.
- Target Labels :
- None – need to predict.

# Cleaning steps can be many, relative to the approach taken & judgement calls.

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.

# Assumptions and choices were made relative the approach taken.

Sometimes there is no certainty as to why something occurred or what something really means. A helpful guide is the description of the column names, however there isn't enough certainty with just names too.

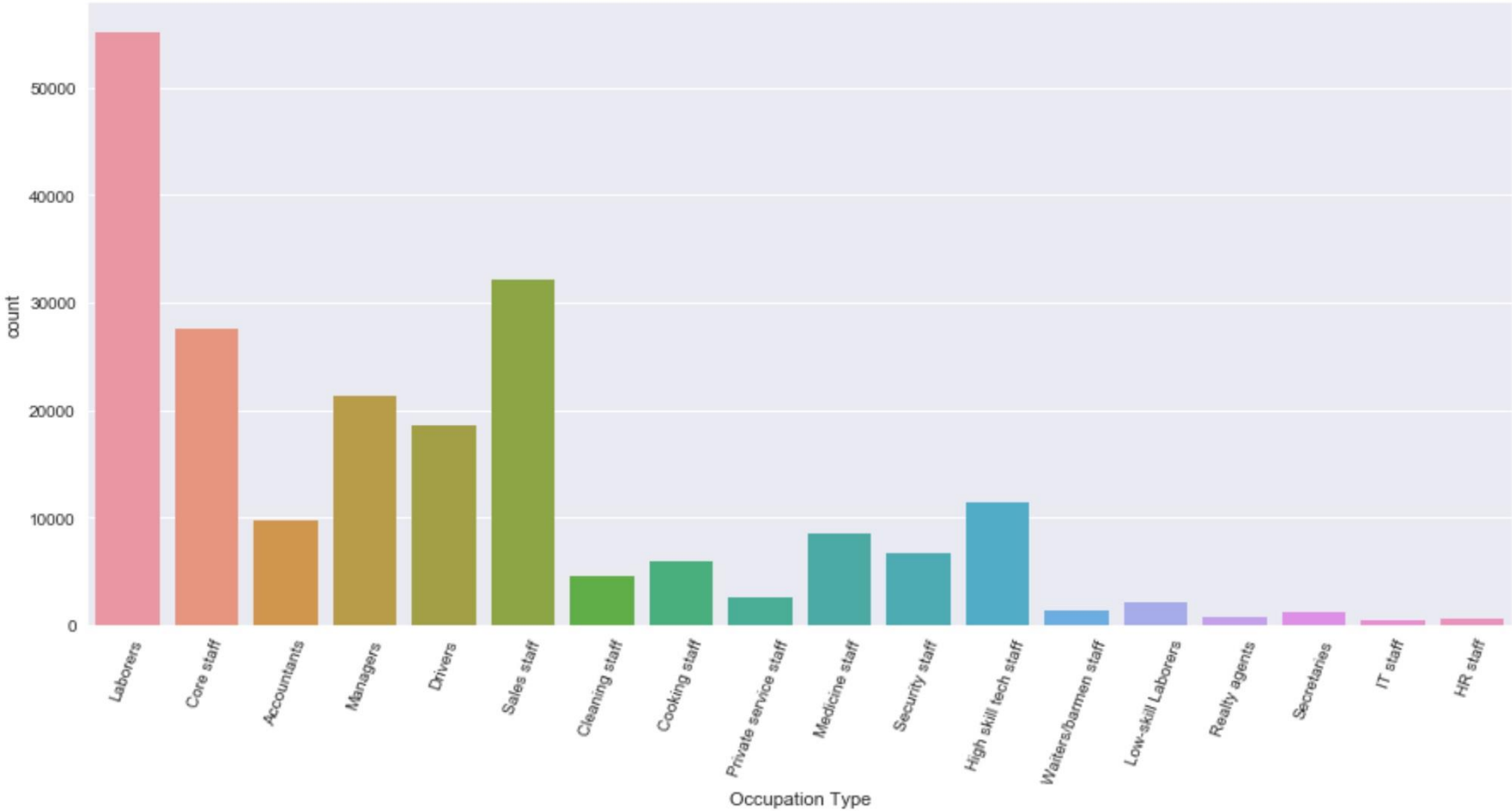
**Choices:**

- Used the same approach similar to target class balanced problem.
- Retained all the columns for processing.
- Generated interaction variables.

**Assumptions**:

- Feature relations are linear.
- Highest linearly correlated variables have a larger hand in making derived features.
- Domain engineered features are powerful.
- Data is noisy.

# Laborers - occupation type were the most borrowers.




Most of the clients are laborers and the least of the clients are IT Staff.

More EDA in Notebook.

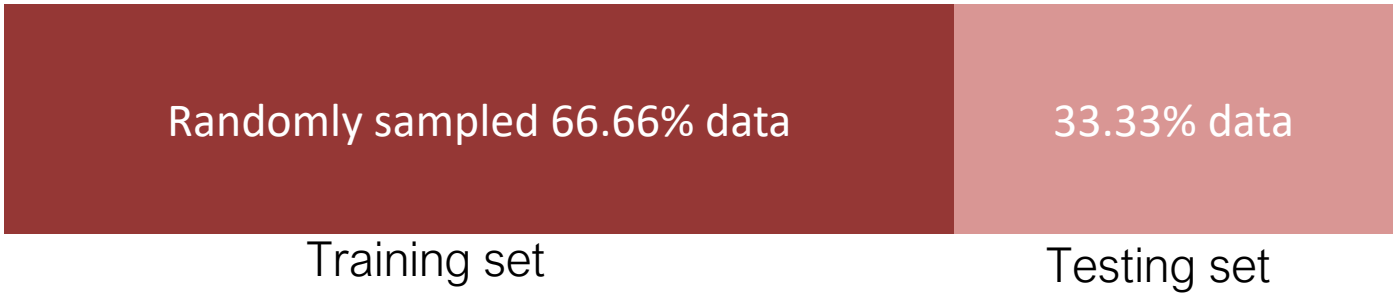
# Predictive Modeling – Outcome of the model is expected to identify the potential that someone will default on a loan.


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, LightGBM, Naïve bayes, Model ensemble.

# 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: 66.66% : 33.33%.
- On the testing set, the target labels are hidden, until the performance is evaluated.

# Currently there are 6 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) | 92.01% |
| 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) | 91.98% |
| 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) | 92.06% |
| LightGBM | (boosting_type='gbdt', class_weight=None, colsample_bytree=1.0, learning_rate=0.1, max_depth=-1, min_child_samples=20, min_child_weight=0.001, min_split_gain=0.0, n_estimators=100, n_jobs=-1, num_leaves=31, objective=None, random_state=None, reg_alpha=0.0, reg_lambda=0.0, silent=True, subsample=1.0, subsample_for_bin=200000, subsample_freq=0) | 92.04% |
| NaïveBayesClassifier | (priors=None) | 32.63% |
| All Stacked | Same as above, but voted. | 92.04% |

More Explanation in Notebook.

# XGBoost 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.