Luddy School of Informatics, Computing and Engineering

Home Credit Default Risk

Team Members:

1. Akash Gangadharan (akganga@iu.edu
2. Laya Harwin (lharwin@iu.edu)
3. Dhairya Shah (shahds@iu.edu)
4. Shobhit Sinha(shosinha@iu.edu)

Agenda For Phase 3

01 **Feature Engineering**

Creating new features

02 **Hyper Parameter Tuning**

Tuning hyper parameters of the model

03 **Modelling Pipelines**

Updating the new modelled pipeline

04 **Kaggle Submission**

Submitting data to get the scores

05 **Future Scope**

Future work to be done

Project Timeline

3 Phase 3 — Feature Engineering and Hyper Parameter Tuning

2 Phase 2 — Performed EDA and created Machine learning pipeline with Base line models

1 Phase 1 — Understood the Data and created Project plan.

Feature Engineering

Treating Highest proportion of zero values and Dividing main data into categorical and numerical features

Treating missing values and correlation with respect to target variables

Adding 17 new features

Performed One Hot Encoding.

Hyper Parameter Tuning

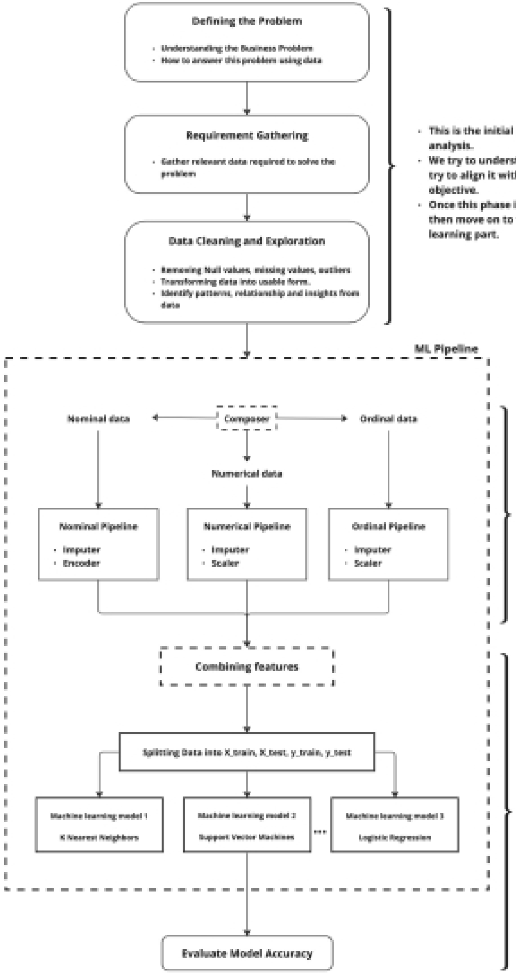
Grid Search CV was used to find the best parameters on the following models

Logistic Regression

Decision Tree

Random Forest

AdaBoost



- This is the initial phase of the data analysis.
- We try to understand the data and try to align it with the business objective.
- Once this phase is completed, we then move on to the Machine learning part.

- The data is segregated according to the type of features.
- Each data is then imputed based on the type of the feature.
- These features are then combined together and send to further step in the ML pipeline.

- The data is split into X_train, y_train, X_test, y_test.
- These data is then used to train various ML models and then validated for the best fit.
- The models are then compared with each other based on their evaluation metrics.

Experiment 2: Using Selected Features on Entire Dataset

Experiments	Pipeline	Parameters	TrainAcc	ValidAcc	TestAcc	Train Time(s)	Test Time(s)	Train AUC
2	Baseline Pipeline(steps=[('rf', RandomForestClassifier())]) with 100 inputs	{'rf__n_estimators': [20], 'rf__criterion': ['gini', 'log_loss']}	99.39%	91.92%	91.56%	1.503872	0.915629	0.999999
	Baseline Pipeline(steps=[('dt', DecisionTreeClassifier())]) with 100 inputs	{'dt__max_depth': [5, 10]}	91.98%	91.94%	91.60%	0.055064	0.916038	0.709815
	Baseline Pipeline(steps=[('lr', LogisticRegression())]) with 100 inputs	{'lr__C': [0.01], 'lr__penalty': ['l1', 'l2']}	91.97%	91.96%	91.60%	0.032885	0.916038	0.736215

Experiment 3: Using Feature Engineering without new features

Experiments	Pipeline	Parameters	TrainAcc	ValidAcc	TestAcc	Train Time(s)	Test Time(s)	Train AUC	Valid AUC	Test AUC	Best Params
3	Baseline Multinomial NB with 219 inputs	{'clf__alpha': (1, 0.1, 0.01, 0.001, 0.0001, 1	91.93%	92.02%	91.83%	0.031847	0.009847	0.623854	0.627888	0.628230	{'memory': None, 'steps': [('scaler', MinMaxSc
	Baseline Logistic Regression with 219 inputs	{'clf__solver': ['lbfgs', 'liblinear', 'newton	91.93%	92.02%	91.85%	0.030177	0.009847	0.755586	0.756720	0.742818	{'memory': None, 'steps': [('scaler', Standard
	Baseline AdaBoostClassifier with 219 inputs	{'clf__n_estimators': [1, 2]}	91.93%	92.02%	91.83%	0.031300	0.009847	0.586557	0.590228	0.587076	{'memory': None, 'steps': [('scaler', Standard
	Baseline Random Forest with 219 inputs	{'rf__n_estimators': [1, 10]	98.55%	91.96%	91.74%	0.034335	0.009847	0.999818	0.642760	0.632850	{'memory': None, 'steps': [('scaler', MinMaxSc
	Baseline KNN with 219 inputs	{'rf__n_estimators': [1, 10], 'rf__min_samples.	92.97%	91.96%	91.74%	0.020069	0.009847	0.939106	0.551639	0.550551	{'memory': None, 'steps': [('scaler', Standard

Experiment 4: Using Feature Engineering with new features

Experiments	Pipeline	Parameters	TrainAcc	ValidAcc	TestAcc	Train Time(s)	Test Time(s)	Train AUC	Valid AUC	Test AUC	Best Params
4	Baseline Multinomial NB with 17 inputs	{'clf__alpha': (1, 0.1, 0.01, 0.001, 0.0001, 1	91.93%	92.02%	91.83%	0.056123	0.010725	0.652961	0.655879	0.659160	{'memory': None, 'steps': [('scaler', MinMaxSc
	Baseline Logistic Regression with 17 inputs	{'clf__solver': ['lbfgs', 'liblinear', 'newton	91.91%	92.01%	91.82%	0.025727	0.010725	0.731534	0.734448	0.728382	{'memory': None, 'steps': [('scaler', Standard
	Baseline AdaBoostClassifier with 17 inputs	{'clf__n_estimators': [1, 2]}	91.93%	92.02%	91.83%	0.019794	0.010725	0.645804	0.644956	0.639363	{'memory': None, 'steps': [('scaler', Standard
	Baseline Random Forest with 17 inputs	{'rf__n_estimators': [1, 10], 'rf__min_samples	98.56%	91.88%	91.57%	0.021411	0.010725	0.999718	0.646205	0.642093	{'memory': None, 'steps': [('scaler', MinMaxSc
	Baseline KNN with 17 inputs	{'rf__n_estimators': [1, 10], 'rf__min_samples	93.05%	91.88%	91.57%	0.019411	0.010725	0.967684	0.556668	0.565649	{'memory': None, 'steps': [('scaler', Standard
	Baseline Ensemble with 17 inputs	{'clf__lr__C': [0.01], 'clf__lr__penalty': ['l	91.97%	91.88%	91.57%	0.017676	0.010725	0.773465	0.556668	0.565649	{'memory': None, 'steps': [('scaler', Standard

Featured Prediction Competition

Home Credit Default Risk

Can you predict how capable each applicant is of repaying a loan?

$70,000
Prize Money

Home Credit Group · 7,176 teams · 5 years ago

Overview Data Code Discussion Leaderboard Rules Team Submissions **Late Submission** ···

Submissions

You selected 0 of 2 submissions to be evaluated for your final leaderboard score. Since you selected less than 2 submission, Kaggle auto-selected up to 2 submissions from among your public best-scoring unselected submissions for evaluation. The evaluated submission with the best Private Score is used for your final score.

0/2

Submissions evaluated for final score

All Successful Selected Errors

Recent

Submission and Description	Private Score	Public Score	Selected
submission.csv Complete (after deadline) · now · Group15 AML Logistic Regression	0.73315	0.73762	☐
submission.csv Complete (after deadline) · 7h ago · group 15 AML	0.66457	0.65961	☐
submission.csv Complete (after deadline) · 7h ago · Submission from Group 15 AML	0.65139	0.66622	☐

Key Results/Findings

Our results and finding are that with feature engineering and adding newly generated features, AdaBoost and MultinomialNB models achieved 91.83% accuracy.

While on the other hand, Logistic Regression achieved 91.85% accuracy with feature engineering (in this case we did not consider the newly generated features).

To conclude, in this phase we did feature engineering and hyper parameter tuning using GridSearch CV. We also created ensemble models to get the maximum score as compared to the base line models scores.

Further we are planning to implement neural network and improve the scores using neural network models.

INDIANA UNIVERSITY BLOOMINGTON