An Explainable Cost-Sensitive Credit Risk Assessment Model

Bhawna Gupta
Student ID : 980432

Introduction

Credit Risk Assessment

- Credit risk is probability of default of loan that a lending firm may not receive the owed principal and interest amount from the borrower.

Impact of Loan default

- Loss in profitability
- Interruption in cash flows
- Increase in collection cost

- Credit risk assessment model will predict the default risk associated with the new loan application before granting the loan.

Introduction

- Credit assessment model should be accurate and interpretable.
- Feature of the credit assessment model
  - Cost-sensitive
  - Explainable

Benefits

Cost Sensitive

- Low misclassification cost
- Better model performance

Explainability

- Transparent credit assessment
- Gain the customer trust in the model

Objectives

- **Challenges with machine learning model**
  - **Imbalanced Dataset**
  - **Black-box Nature**

- **Research Questions:**

1. **Will oversampling technique improve the performance of the model in case of data imbalance?**
2. **Will the result of the black-box model be interpretable?**

Objectives

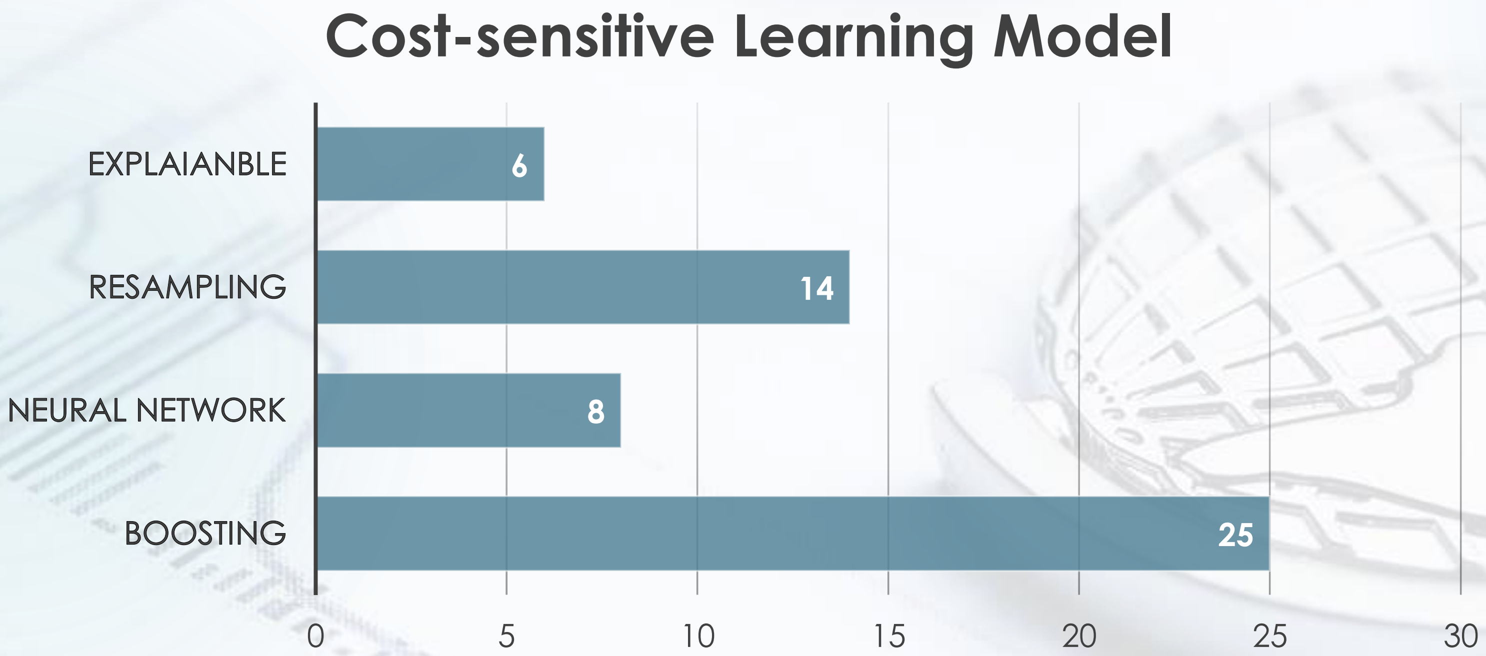
- **To analyze performance of ensemble classifiers.**
- **To find an optimal resampling technique**
- **Evaluate model's performance using misclassification cost metrics**
- **To suggest an explainable AI method**

Literature Review

- 37 studies were reviewed from 2018-2021 on credit assessment model.
- Out of 37, twenty five studies implement their cost-sensitive model using boosting classifiers.
- 8 studies were implemented using cost-sensitive neural network. But some of them were facing the overfitting issue.
- Ensemble of Neural network with boosting classifiers were performed better.

Literature Review

- To handle imbalance problem 14 studies used resampling techniques.
- Only six studies were based on explainable AI to interpret the result.
- And combination of both is none.



Methodology

- **Dataset Description**
  - ✓ A eleven year of data (2007-2018) of the Lending club dataset was being used.
  - ✓ Loan_status feature was the target feature.
  - ✓ Dataset was divided into individual and joint applicant type.
- **Data Cleaning**
  - ✓ Columns that had more than 50% null values were removed.
  - ✓ Remaining null values were imputed with mean and mode values.
  - ✓ Outliers were capped with suitable percentile values to reduce the data variance.
  - ✓ Columns with more than 80% skewness were removed.
  - ✓ Redundant features were removed from the dataset.

Methodology

- **Model Preprocessing**
  - ✓ Transforming categorical features to numerical feature using label encoding.
  - ✓ Split of dependent and independent features.
  - ✓ Normalization of data using standard scaler.
  - ✓ Split of train and test set.
- **Model Implementation**
  - ✓ Three boosting classifiers XGBoost, LightGBM and CatBoost were trained on dataset.
  - ✓ Stacking model of all three boosting classifiers was trained on train data.
  - ✓ Hyperparameter tuning was done using RandomizedSearchCV.

Methodology

- **Resampling Technique**
  - ✓ Performance of two oversampling technique SMOTE and ADASYN were compared.

- **Cost-sensitive Evaluation Metrics**
  - ✓ AUC-score, F1-score, G-mean, precision and recall were used to evaluate the performance.
  - ✓ Type-I error and type –II error metrics was used to calculate the misclassification cost.

- **Model Explainability**
  - ✓ SHAP explainable AI was used to calculate the features contribution in the prediction result.

Result & Discussion

Model Name	ACC	Confusion Matrix TP	FN	FP	TN	AUC	Precision	Recall	F1-score	G-mean	Type-I Error	Type-II Error
XGBoost	99.90%	316890	19	222	78733	99.856%	99.930%	99.994%	99.962%	99.856%	0.281%	0.006%
XGBoost-Tunned	99.90%	316892	17	455	78500	99.709%	99.857%	99.995%	99.926%	99.709%	0.576%	0.005%
XGBoost-SMOTE	99.90%	316885	24	220	78735	99.857%	99.931%	99.992%	99.962%	99.857%	0.279%	0.008%
XGBoost-ADASYN	99.90%	316884	25	238	78717	99.845%	99.925%	99.992%	99.959%	99.845%	0.301%	0.008%
Lightgbm	99.80%	316870	129	467	78488	99.684%	99.853%	99.959%	99.906%	99.684%	0.591%	0.041%
LightGBM-Tunned	99.90%	316860	49	460	78495	99.701%	99.855%	99.985%	99.920%	99.701%	0.583%	0.015%
LightGBM-SMOTE	99.90%	316844	65	490	78465	99.679%	99.846%	99.979%	99.912%	99.679%	0.621%	0.021%
LightGBM-ADASYN	99.90%	316857	52	489	78466	99.682%	99.846%	99.984%	99.915%	99.682%	0.619%	0.016%
Catboost	99.90%	316884	25	302	78653	99.805%	99.905%	99.992%	99.948%	99.805%	0.382%	0.008%
Catboost-Tunned	99.90%	316873	36	421	78534	99.728%	99.867%	99.989%	99.928%	99.727%	0.533%	0.011%
CatBoost-SMOTE	99.90%	316853	56	433	78522	99.717%	99.864%	99.982%	99.923%	99.717%	0.548%	0.018%
CatBoost-ADASYN	99.90%	316856	53	444	78511	99.710%	99.860%	99.983%	99.922%	99.710%	0.562%	0.017%
Stacking	99.90%	316890	19	222	78733	99.856%	99.930%	99.994%	99.962%	99.856%	0.281%	0.006%
Stacking-SMOTE	99.90%	316885	24	220	78735	99.857%	99.931%	99.992%	99.962%	99.857%	0.279%	0.008%
Stacking-ADASYN	99.90%	316884	25	238	78717	99.845%	99.925%	99.992%	99.959%	99.845%	0.301%	0.008%

Cost-sensitive metrics result for the individual test set

- **XGBoost and Stacking Classifier with SMOTE achieved the lowest misclassification cost (Type-I and Type-II error).**

Result & Discussion

Model Name	ACC	Confusion Matrix TP	FN	FP	TN	AUC	Precision	Recall	F1-score	G-mean	Type-I Error	Type-II Error
XGBoost	99.80%	5803	2	11	1926	99.699%	99.811%	99.966%	99.888%	99.698%	0.568%	0.034%
XGBoost-Tunned	99.80%	5802	3	12	1925	99.664%	99.794%	99.948%	99.871%	99.664%	0.620%	0.052%
XGBoost-SMOTE	99.80%	5803	2	11	1926	99.699%	99.811%	99.966%	99.888%	99.698%	0.568%	0.034%
XGBoost-ADASYN	99.90%	5805	0	12	1925	99.690%	99.794%	100.000%	99.897%	99.690%	0.620%	0.000%
Lightgbm	99.90%	5804	1	9	1928	99.759%	99.845%	99.983%	99.914%	99.759%	0.465%	0.017%
LightGBM-Tunned	99.80%	5803	2	10	1927	99.725%	99.828%	99.966%	99.897%	99.724%	0.516%	0.034%
LightGBM-SMOTE	99.90%	5804	1	9	1928	99.759%	99.845%	99.983%	99.914%	99.759%	0.465%	0.017%
LightGBM-ADASYN	99.90%	5805	0	11	1926	99.716%	99.811%	100.000%	99.905%	99.716%	0.568%	0.000%
Catboost	99.80%	5803	2	10	1927	99.725%	99.828%	99.966%	99.897%	99.724%	0.516%	0.034%
Catboost-Tunned	99.40%	5780	25	20	1917	99.268%	99.655%	99.569%	99.612%	99.268%	1.033%	0.431%
CatBoost-SMOTE	99.90%	5802	3	7	1930	99.793%	99.879%	99.948%	99.914%	99.793%	0.361%	0.052%
CatBoost-ADASYN	99.80%	5805	0	13	1924	99.664%	99.777%	100.000%	99.888%	99.664%	0.671%	0.000%
Stacking	99.90%	5804	1	9	1928	99.759%	99.845%	99.983%	99.914%	99.759%	0.465%	0.017%
Stacking-SMOTE	99.90%	5804	1	8	1929	99.785%	99.862%	99.983%	99.923%	99.785%	0.413%	0.017%
Stacking-ADASYN	99.80%	5805	0	13	1924	99.664%	99.777%	100.000%	99.888%	99.664%	0.671%	0.000%

Cost-sensitive metrics result for the Joint test set

- **For low volume dataset only CatBoost with SMOTE has lowest misclassification cost.**

Result & Discussion

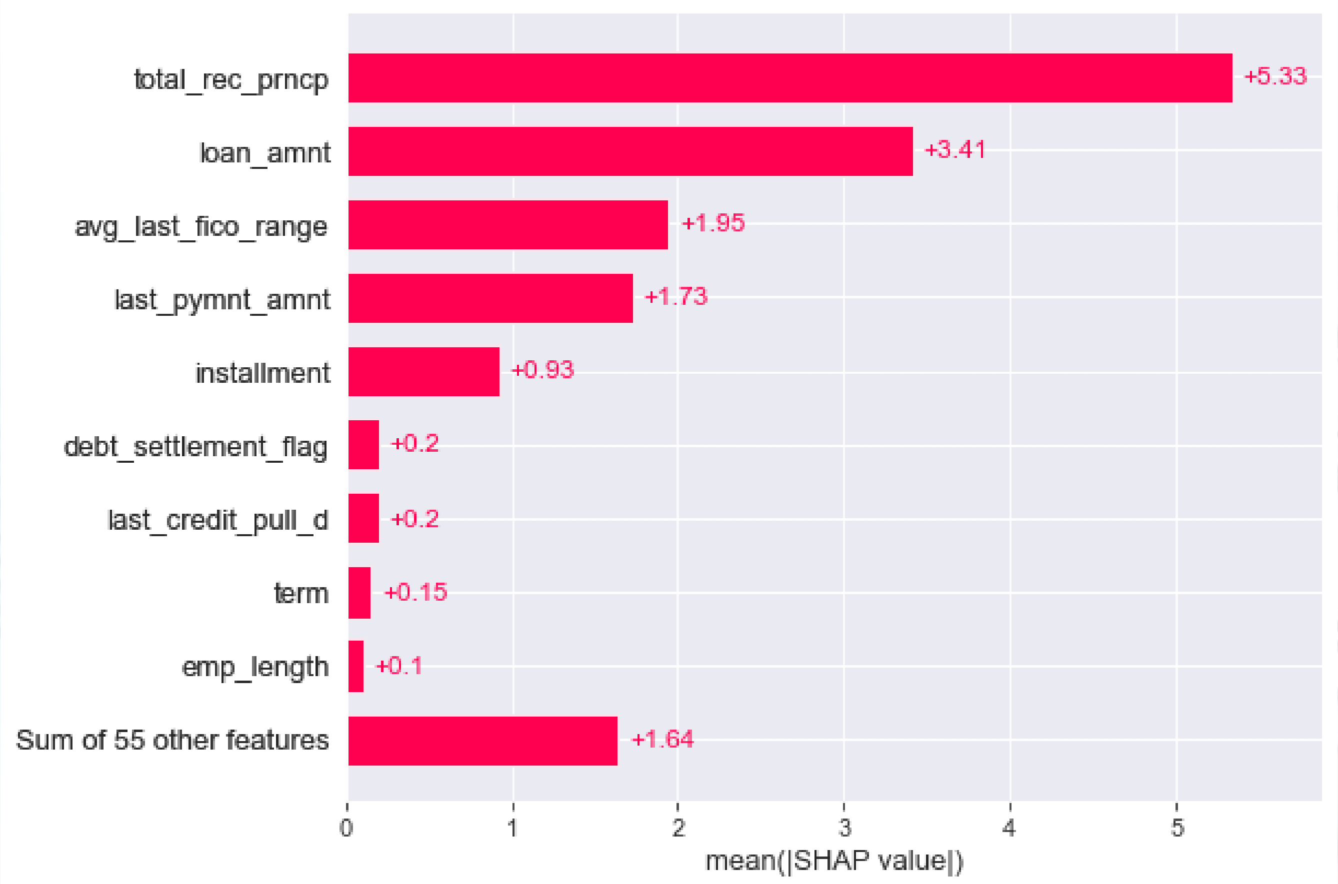
Comparative performance analysis with previous studies

Paper	Model	AUC	F1-Score	G-mean	Sensitivity	Specificity
Best performing model in the previous study						
(Yotsawat et al., 2021a)	CS-NNE	70.82	-	65.00	62.69	67.41
	XGBoost	69.69	-	30.87	9.79	98.21
(He et al., 2018)	EBCA	73.06	99.38	1.638	-	-
	XGBoost	71.45	99.35	0.00	-	-
(Chengeta and Mabika, 2021a)	CNN	99.74	95.39	-	92.30	-
	XGB	99.44	86.23	-	94.30	-
	LightGBM	99.76	89.54	-	97.04	-
	CatBoost	99.55	87.19	-	99.00	-
(Kun et al., 2020a)	Stacking	98.11	98.32	-	98.68	-
	XGBoost	97.69	97.95	-	98.40	-
Best performing model in this study						
Individual Dataset	**XGBoost-SMOTE**	**99.86**	**99.96**	**99.86**	**99.92**	**99.72**
	Stacking-SMOTE	**99.86**	**99.96**	**99.86**	**99.92**	**99.72**
Joint Dataset	**CatBoost-SMOTE**	**99.79**	**99.91**	**99.79**	**99.95**	**99.64**

Result & Discussion

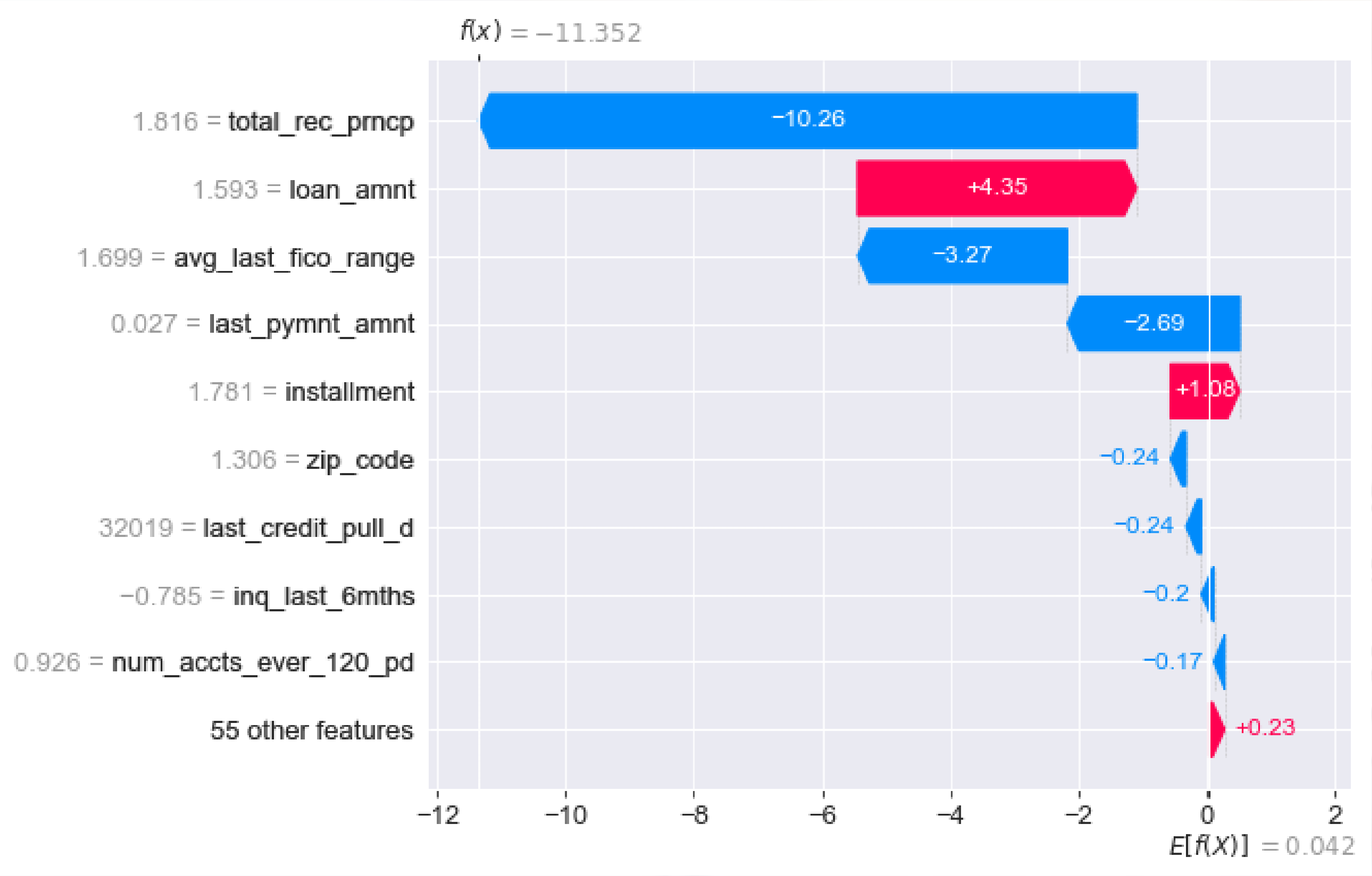
Interpretation of the result

Top 10 features contribution in the prediction result



Result & Discussion

Top 10 features contribution in the prediction result of first row



Conclusion

Major Achievements:

- Achieved the lowest misclassification cost
  - ✓ Type-I error (misclassification of default as non-default) 0.279%.
  - ✓ Type-II error (misclassification of non-default as default) 0.005%.
- Black-box nature of classifiers were interpreted by SHAP explainable AI.
- Boosting classifier XGBoost with oversampling technique SMOTE achieved the highest evaluation score for cost-sensitive learning.

Other Achievement:

- Performance of the stacking model of the three boosting classifiers with SMOTE achieved the same result as XGBoost-SMOTE.

Contribution & Future Scope

- **Contribution of the study**
  - ✓ **Combination of boosting classifier (XGBoost) with SMOTE provide the lowest misclassification error.**
  - ✓ **Using SHAP explainable AI make the cost-sensitive learning process transparent.**

- **Future scope**
  - ✓ **Stacking model of boosting classifier and neural network with oversampling will be interesting to be explored.**
  - ✓ **Performance of boosting classifier with undersampling can be compare with the performance of this study.**

Thank You