DATA MINING &
WEB ALGORITHMS
DR. ARCHANA PURWAR

CREDIT RISK PREDICTION REPORT

SUBMITTED BY:

VATSAL GUPTA (17104060)
MANISHA RATHORE (17104033)
AKSHARA NIGAM (17104018)

MOTIVATION

Nowadays, creditworthiness is very important for everyone since it is regarded as an indicator of how dependable an individual is. In various situations, service suppliers need to first evaluate the customers' credit history and then decide whether or not they will provide the service. However, it is time-consuming to check the entire personal portfolios and generate a credit report manually. Thus, the credit score is developed and applied for this purpose because it is time-saving and easily comprehensible.

The process of generating the credit score is called credit scoring. It is widely applied in many industries, especially in banking. The banks usually use it to determine who should get credit, how much credit they should receive, and which operational strategy can be taken to reduce the credit risk. Thus, this project aims to build a model that can predict a person's creditworthiness.

NOVELTY

The project uses various classification algorithms along with hyperparameter tuning in order to build a final model with high accuracy and F1 score. This is hence new, as people have always used algorithms like SVM (Support Vector Machine), Naive Bayes or various Deep Learning Models to solve this issue, but here our Catboost model is used, and a comparison based on accuracy is made with the rest (Random Forest, Decision Tree, XGBoost, LightGBM).

Algorithms Used

Decision Trees

A decision tree is a decision support tool that uses a tree-like graph to model possible consequences of each decision, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements. Tree-based learning algorithms are considered to be one of the best and mostly used supervised learning methods. Tree-based methods empower predictive models with high accuracy, stability, and ease of interpretation. Unlike linear models, they map nonlinear relationships quite well.

Parameters to tune a decision tree classifier:

- max_depth
- max_feature

They are adaptable at solving any kind of problem at hand (classification or regression).

A. **Root Node:** It represents the entire population or sample, and this further gets divided into two or more homogeneous sets.
B. **Splitting:** It is a process of dividing a node into two or more sub-nodes.
C. **Decision Node:** When a sub-node splits into further sub-nodes, then it is called the decision node.
D. **Leaf/ Terminal Node:** Nodes that do not split are called Leaf or Terminal nodes.
E. **Pruning:** When we remove sub-nodes of a decision node, this process is called pruning.

Random Forest

Random forests or random decision forests are an ensemble learning method (bagging) for classification, regression, and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction of the individual.

Random Forests are composed of multiple decision trees that take a random sample of the features to form their prediction, and then decide the final classification by consensus vote from all the trees.

The advantage of a Random Forest model over a simple Decision Tree is that Decision Trees are prone to overfitting. Decision Trees, especially ones that are deep, will form detailed feature branches that fit the training data but do not generalize well.

Parameters to tune Random Forests:

- bootstrap
- max_depth
- max_features
- min_sample_leaf
- min_sample_split

Boosting-Based Algorithms

Boosting algorithms perform subsequent training by placing weight on data that is hard to classify and less weight on data that is easy to classify. It uses a loss function to measure error and correct for it in the next iteration. Boosted-tree algorithms also penalize models for complexity. The prediction of the final boosted-tree model is the weighted sum of the predictions made by the individual models.

XGBoost

A popular implementation of gradient boosted trees designed for speed and performance. A disadvantage of XGBoost is that it requires data to be stored in memory when run. Since then, newer algorithms have been developed that do not have to process data in memory.

Parameters to tune a XGboost classifier:

- silent
- scale_pos_weight
- learning_rate
- colsample_bytree
- subsample
- objective
- n_estimators
- reg_alpha
- max_depth
- gamma

Catboost

The newest of the three algorithms, CatBoost, is designed to handle categorical features better. Instead of one-hot encoding features, which causes the curse of dimensionality,

CatBoost transforms categorical features into values based on a statistical calculation of its relationship with the target variable. Catboost also divides a given dataset into random permutations and applies ordered boosting on those random permutations.

Parameters to tune a Catboost classifier:

- depth
- learning_rate
- iterations

LightGBM

Designed to be a faster implementation of XGBoost with similar accuracy. This algorithm inspects the most informative samples and skips non-informative samples. It also bins sparse features, reducing complexity.

Parameters to tune a LightGBM classifier:

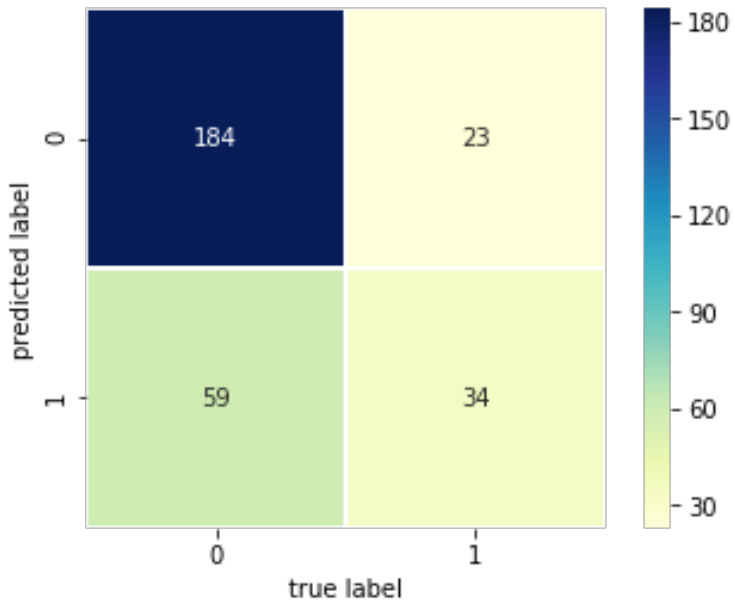
- max_depth
- learning_rate
- num_leaves
- n_estimators

Hyperparameter Tuning & Results

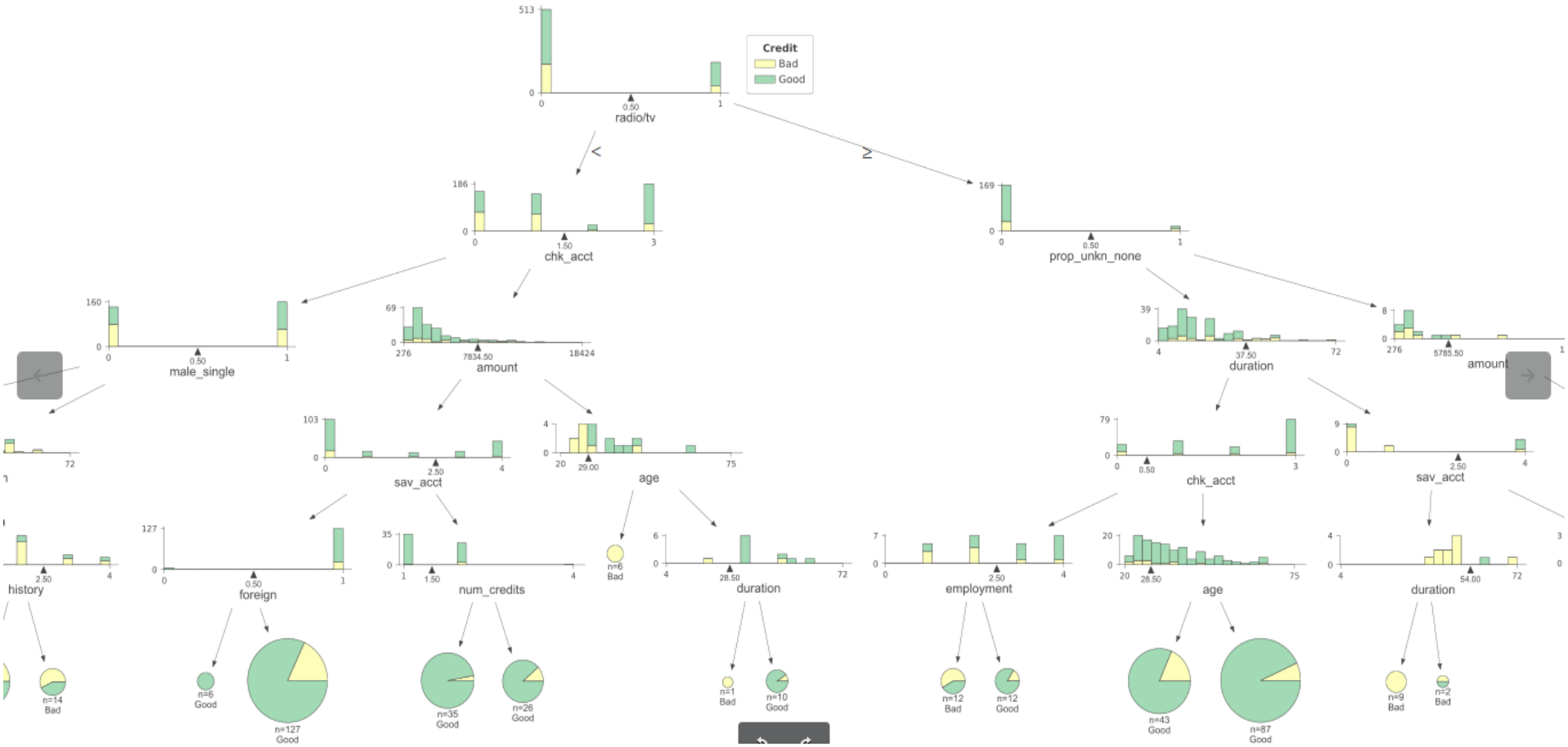
Decision Trees

The Grid Search found the following best parameters:

max_depth	:	3
max_feature	:	8



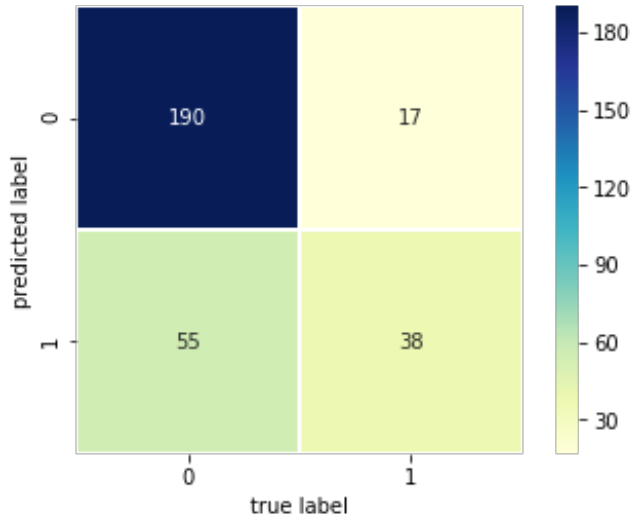
ACCURACY ON TEST SET: 0.727


Credit
Bad
Good
radio/tv
chk_acct
male_single
amount
prop_unkn_none
duration
amount
sav_acct
age
chk_acct
sav_acct
history
foreign
num_credits
duration
employment
age
duration
n=14 Bad
n=6 Good
n=127 Good
n=35 Good
n=26 Good
n=6 Bad
n=1 Bad
n=10 Good
n=12 Bad
n=12 Good
n=43 Good
n=87 Good
n=9 Bad
n=2 Bad

Random Forest

The Grid Search found the following best parameters:

bootstrap	:	true
max_depth	:	None
max_features	:	Auto
min_sample_leaf	:	1
min_sample_split	:	2
n_estimators	:	200



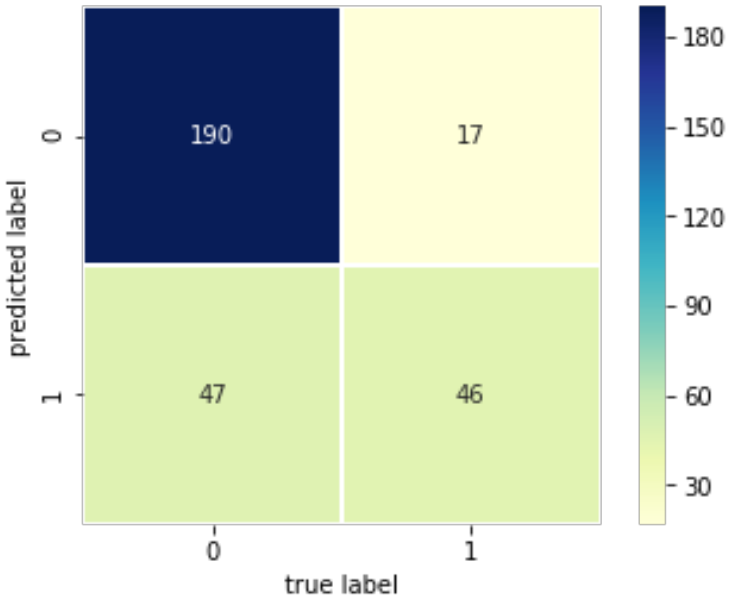
ACCURACY ON TEST SET: 0.760

By using an ensemble method, the accuracy has improved 4% over the simple Decision Tree model.

CatBoost

The best parameters:

depth	:	[6,8,10]
learning_rate	:	[0.01, 0.05, 0.1]
iterations	:	[30,50,100]



ACCURACY ON TEST SET: 0.786667

LightGBM

The best parameters:

max_depth	:	[25,50,75]
learning_rate	:	[0.01,0.05,0.1]
num_leaves	:	[300,900,1200]
n_estimators	:	200

ACCURACY ON TEST SET: 0.7500

XGBoost

The best parameters:

silent	:	*False*
scale_pos_weight	:	1
learning_rate	:	0.01
colsample_bytree	:	0.4

subsample	:	0.8
objective	:	'binary:logistic'
n_estimators	:	1000
reg_alpha	:	0.3
max_depth	:	4
gamma	:	10

ACCURACY ON TEST SET: 0.76667

CONCLUSION

As a part of this project, we have implemented and compared several data mining algorithms on a credit risk prediction dataset. There was a jump in the accuracy from the Decision Tree model to the ensemble models, which makes sense since ensemble models are designed to perform better through consensus prediction. Furthermore, the gradient boosted algorithms also performed somewhat better than the traditional decision tree. The Catboost classifier, in particular, performed very well. It outperformed the other models with the highest accuracy of 78.7%. This dataset contained many categorical features suited to using the CatBoost algorithm.