WHITE PAPER

Evaluation of Feature Selection Methods

Farhan Mohammad and Samira Golsefid, Ph.D.

PayPal

PayPal

Contents

Overview

In machine learning, one of the key challenges is to select the right set of features as inputs to a model. The features that are used to train a model will have a huge influence on the achieved performance.

Irrelevant or partially relevant features can negatively impact the performance of a model, and this becomes a pain point for a data scientist. Feature selection algorithms help overcome this problem by identifying relevant features from the original features without losing a lot of information.

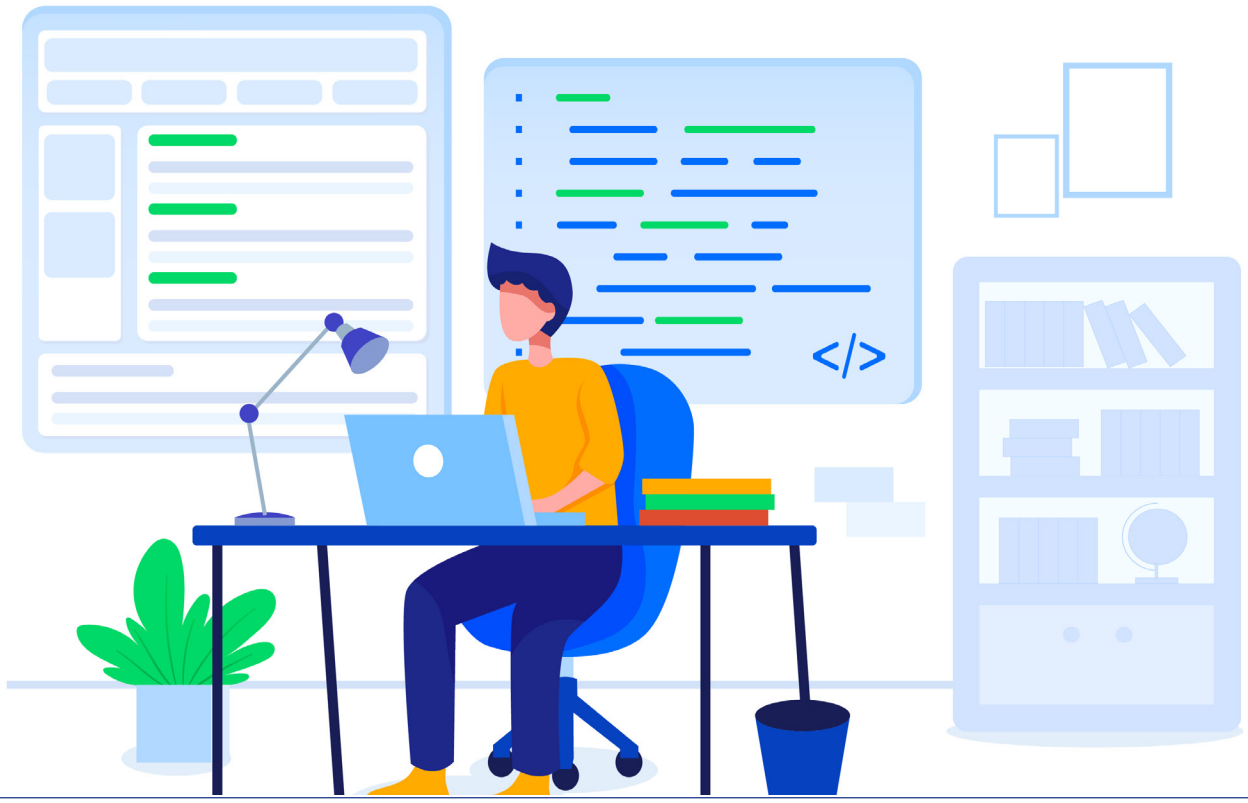
In this paper, we present an extensive overview and performance analysis of different feature selection methods that can be applied to a wide array of machine learning problems. We have taken a sample dataset to demonstrate the application of feature selection methods. We focus on filter, wrapper and embedded methods and additionally propose two new approaches to select features: union and voting selection.

The results of this paper demonstrate how selecting features direcly impacts a model's performance, which is especially important in a risk-sensitive application such as fraud detection.

What is feature selection?

Feature selection is the process of selecting a subset of relevant features used to train a machine learning model.

Raw data is the basic building block of ML algorithms. But on its own it can't be used to accurately train models. Instead, it must be refined to "features" – variables or attributes that can be used for analysis. The features we use to train a ML model are crucial to its performance. This means selecting the most relevant ones possible is absolutely vital.



PayPal

Why is feature selection important?

Feature selection has been proven to be effective and efficient in handling high dimensional data. Here are some of the reasons why feature selection is extremely important:

Enhanced generalization by reducing overfitting: excess variables in the data can add noise in a model, which leads to overfitting. By eliminating the noisy features in the data, we can substantially improve the generalization capability of a machine learning model.

Reduces training times: reducing the number of variables to build a machine learning model will reduce the computational cost and thus speed up model building.

Increase model interpretability: we sometimes lose explainability in a machine learning model when we have many features. By reducing the number of features, the model becomes simpler and easier to interpret. A model with 50 features has better explainability than a model with 200 features.

Variable redundancy: features within data are often highly correlated, making them redundan
t. By removing these correlated features, the model will be less prone to make noise-based predictions.

Reduces prediction time: reducing the number of features reduces the computation cost - simpler models tend to have faster prediction times.

Overview of feature selection methods

Feature selection methods are commonly categorized into three different types: **filter methods**, **wrapper methods**, and **embedded methods**. In addition to this, we propose two new approaches to feature selection: **union** and **voting selector**.

Filter Method

In this method, the selection of features is done independently of a machine learning algorithm. Instead, this method relies on the characteristics of the data to filter features based on a given metric.

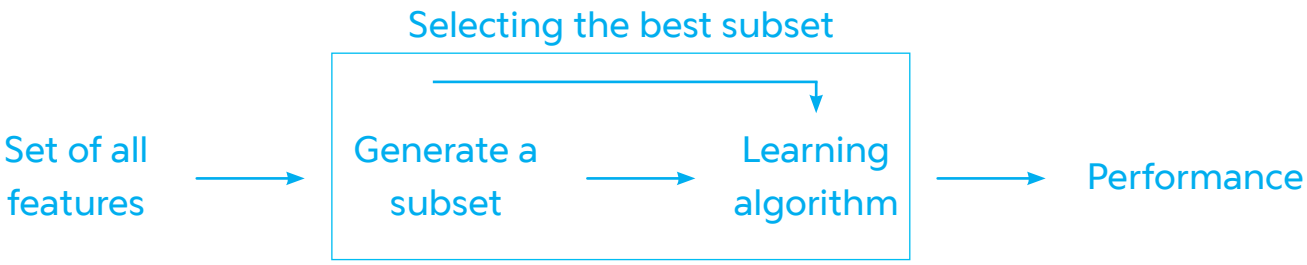
Chi-square Test

Chi-square test is a statistical test of independence to determine the dependency of independent variable and target using their frequency distribution.

- Advantages
  - Computationally very fast and provides a quick way to screen the features
- Disadvantages
  - Does not contemplate feature redundancy and feature interactions and does not handle multicollinearity

Wrapper Method

In this method, feature selection is based on a search criteria where a model is initially trained on a subset of features. Based on the inferences drawn from the previous model, we decide to either add or remove features.



One common example of the wrapper method used in our evaluation is **recursive feature elimination**.

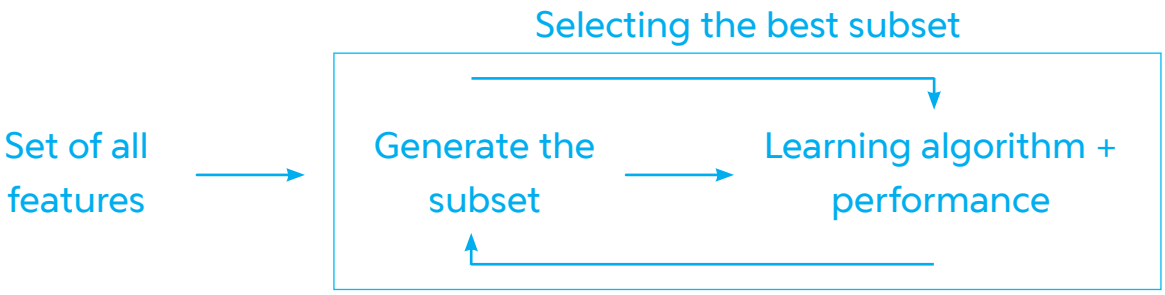
Recursive Feature Elimination

Recursive feature elimination is a greedy optimization algorithm which aims to find the best subset of features by building a model and recursively selecting a smaller and smaller set of features. It then constructs another model based on the remaining features. This process is repeated until all the features are evaluated by the model. Low importance features are eliminated. The final feature ranking is obtained based on the order of their elimination.

- Advantages
  - More accurate than filtering methods
  - Unlike filter methods, they can detect feature interactions
- Disadvantages
  - Computationally very expensive as the feature space grows

Embedded Methods

Embedded methods use the qualities of both the filter and wrapper methods. With this method, feature selection is embedded within the machine learning algorithm. The embedded methods can be further classified into two categories:



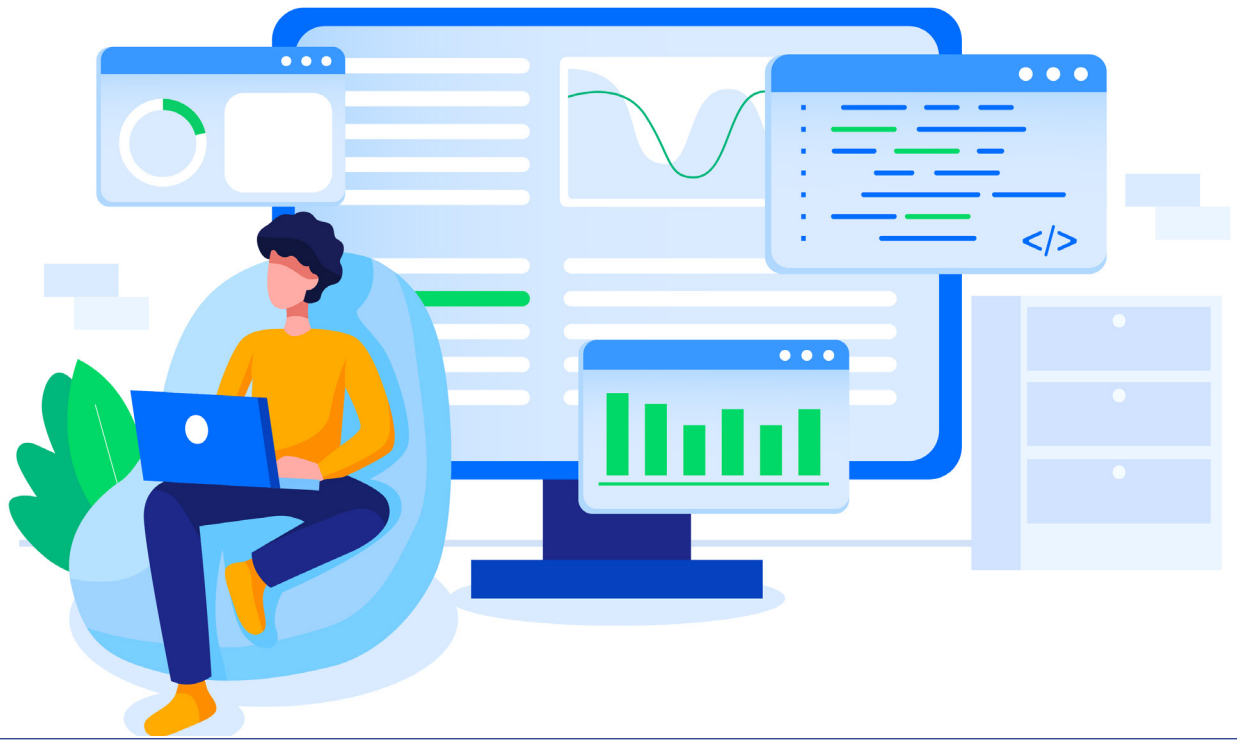
Lasso regularization

Lasso uses l1 regularization that has a property to shrink some parameters or feature coefficients to zero. It uses logistic regression to train a model with l1 penalty term to evaluate the coefficients of different variables and remove those variables whose coefficients are zero.

Tree based random rorest:

Random forest is an ensemble method that uses bagged decision trees with random feature subsets chosen at each split point. It calculates feature importance using node impurities in each decision tree. The final feature importance is obtained by taking an average of all decision tree feature importances.

- Advantages
  - Provides a more reliable feature estimate than a single decision tree algorithm
  - Reduces overfitting
  - No feature scaling required
  - Robust to outliers
- Disadvantages
  - Less interpretable than an individual decision tree
  - High computational cost and memory consumption
  - Does not handle categorical features directly



XGBoost

XGBoost is an optimized gradient boosting algorithm that supports parallel processing, tree-pruning, handling missing values, and regularization to avoid overfitting/bias. It automatically calculates feature importance for all features and the final feature importance scores are available in the feature_importances_ attribute of the trained model.

- Advantages
  - Parallelization
  - Built-in regularization
  - Effective tree-pruning
- Disadvantages
  - Prone to overfitting on small datasets
  - Does not handle categorical features directly
  - Less interpretable

LightGBM

LightGBM is a powerful implementation of boosting method that is similar to XGBoost but varies in a fewspecific ways, specifically in how it creates the tree or base learners. Unlike other ensemble techniques, LGBM grows trees leaf-wise, which can reduce loss during the sequential boosting process.

This usually results in higher accuracy than other boosting algorithms. Similar to XGBoost, the importance of each feature can be obtained from the feature_importances_ attribute embedded in the algorithm.

- Advantages
  - Faster training speed and higher efficiency

- Better accuracy than other boosting algorithms
- Works well with categorical features

- Disadvantages
  - Prone to overfitting on small datasets
  - Less interpretable

Catboost

Catboost is a gradient boosting algorithm that implements symmetric trees in order to reduce the model prediction time. It is well known for efficiently handling categorical features for large datasets. The feature importance ranking can be obtained from get_feature_importance attribute based on the model's loss function.

- Advantages
  - Handles categorical features automatically
  - It is robust as it does not require extensive hyper-parameter tuning
- Disadvantages
  - Prone to overfitting on small datasets
  - Less interpretable

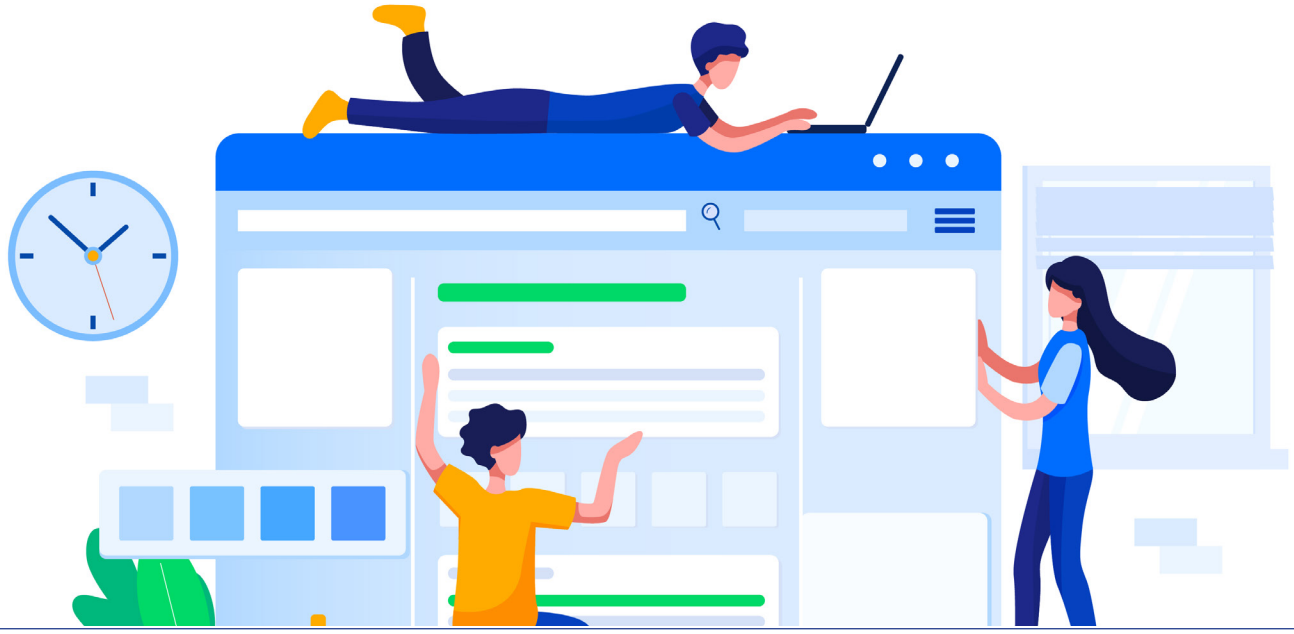
Proposed Methods

The idea behind these proposed methods is to combine a different subset of features chosen from each feature selection method, based on certain criteria in order to obtain more representative features, resulting in effective model performance. We discuss two strategies:

Voting Selector

In voting selector, we apply a variety of feature selection methods to pick the top variables and assign a vote for each variable chosen. It then calculates the total votes for each variable chosen and then chooses the best features based on majority voting.

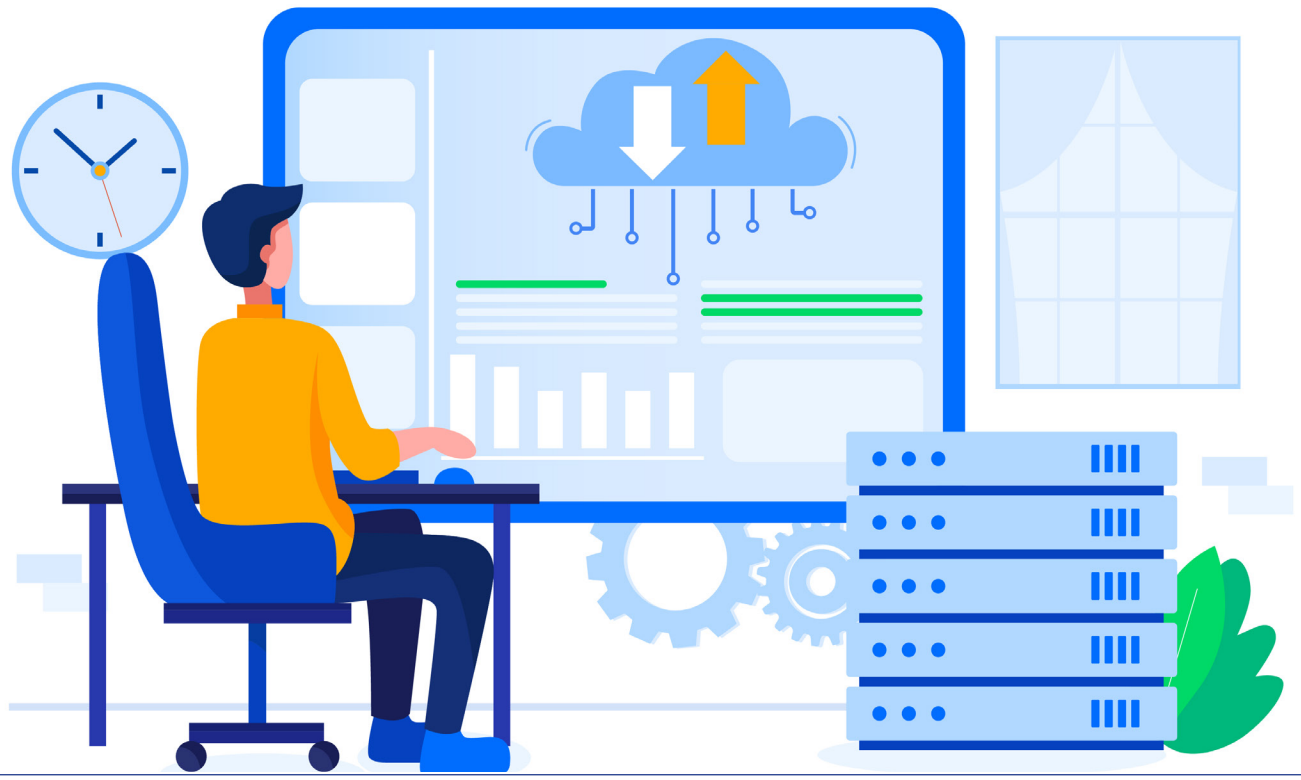
- Advantages
  - It provides an easy way to pick the best set of features based on top features identified from each method.
- Disadvantages
  - It becomes computationally costly with higher number of feature selection methods.



Union Selector

In union selector, we apply a number of feature selection methods to pick the top features and combine them together by taking a union of each feature subset. The final output is a collection of distinct features obtained from the combination of features.

- Advantages
  - Results in higher performance as it takes into account every feature that was chosen by a feature selection method
- Disadvantages
  - The feature space can grow exponentially if a high number of features are used in the selection process.



Experiment

The performance of each feature selection method is evaluated by training multiple machine learning models using the best subset of features selected from each method. The train/validation/test sets, range of hyperparameters, and the top K features used in the model building process were tested against the same benchmark; except in the case of union selection, the number of features may be greater than K (where K is the total number of features selected to train a machine learning model).

Below is the benchmark information used in the model evaluation:

- Model: to train each model using top features identified from all the feature selection methods discussed, we use GBM (Gradient Boosting Machines) from the H2O modeling framework
- Loss Function: logloss
- Number of models trained: 10 (equivalent to the number of feature selection methods implemented)
- Number of features used (K): 80 (equivalent to the number of features selected from each method)
- Hyper-parameter Tuning: grid search with same set of hyper-parameters

Results and discussion

In this experiment, we implement a total of 10 feature selection methods and evaluate the performance of each method using GBM model scores. To compare the model performance, we compute the precision and recall scores.

Precision is the number of correctly predicted frauds divided by the total number of predicted frauds, and recall is the number of correctly predicted frauds divided by the total number of actual frauds.

The precision and recall scores are evaluated at a **5%**, **15%**, and **30%** threshold based on the percentile analysis:

Feature Selection Method	Computation time (seconds)	Threshold = 30%		Threshold = 15%		Threshold = 5%	
		Precision	Recall	Precision	Recall	Precision	Recall
Union	757	16.12	79.25	24.06	59.14	39.42	32.29
Voting	680	16.09	79.08	24.77	60.88	40.77	33.39
Catboost	77.41	15.92	78.25	23.86	58.64	39.73	32.54
LGBM	18.34	15.85	77.92	24.15	59.36	40.94	33.53
RFE	177	15.81	77.73	24.38	59.91	40.7	33.34
XGBoost	21.4	15.81	77.07	24.03	59.06	38.11	31.21
Random Forest	22.98	14.92	73.34	21.02	51.65	33.79	27.67
Lasso	340	14.89	73.17	21.54	52.92	34.09	27.92
Chi-square	9	13.9	68.31	17.18	42.23	27.08	22.18
Union (Catboost, XGBoost, and LGBM)	149	16.07	78.97	24.93	61.27	41.21	33.75

Based on internal PayPal data, 2020

At a 30% threshold, **union selector** is the best performing method with a precision of **16.12%** and recall of **79.25%**.

At a 15% threshold, **voting selector** has the best performance with a precision of **24.77%** and recall of **60.88%**. **Catboost** is the third best performing method at all three thresholds with a precision of **15.92%** and recall of **78.25%**. It is also very efficient in terms of time complexity, compared to the union and voting methods.

Finally, at a 5% threshold, **LightGBM** results in the best performance in terms of precision-recall scores and overall time complexity, followed closely by the **RFE** and **XGBoost** methods. **Random Forest**, **Lasso**, and **Chi-square** are the worst performing methods, which indicate that these methods are not able to learn complex patterns and relationships in the data.

Although union selector with a combination of all methods results in the best performance in terms of model precision and recall scores, it has high computation complexity. In order to overcome this challenge and take advantage of its performance, we evaluate the union selector method based on Catboost, XGBoost and LightGBM algorithms.

As we can see this method is showing best result at the 15% and 5% thresholds with a significant reduction in time complexity compared to union with all methods.

The following two charts represent the performance comparison of feature selection methods in terms of model f1-score and time complexity. All methods except Random forest, Lasso and Chi-square have a high F1 score. However, in terms of computational complexity, LightGBM is showing the best performance among top methods.

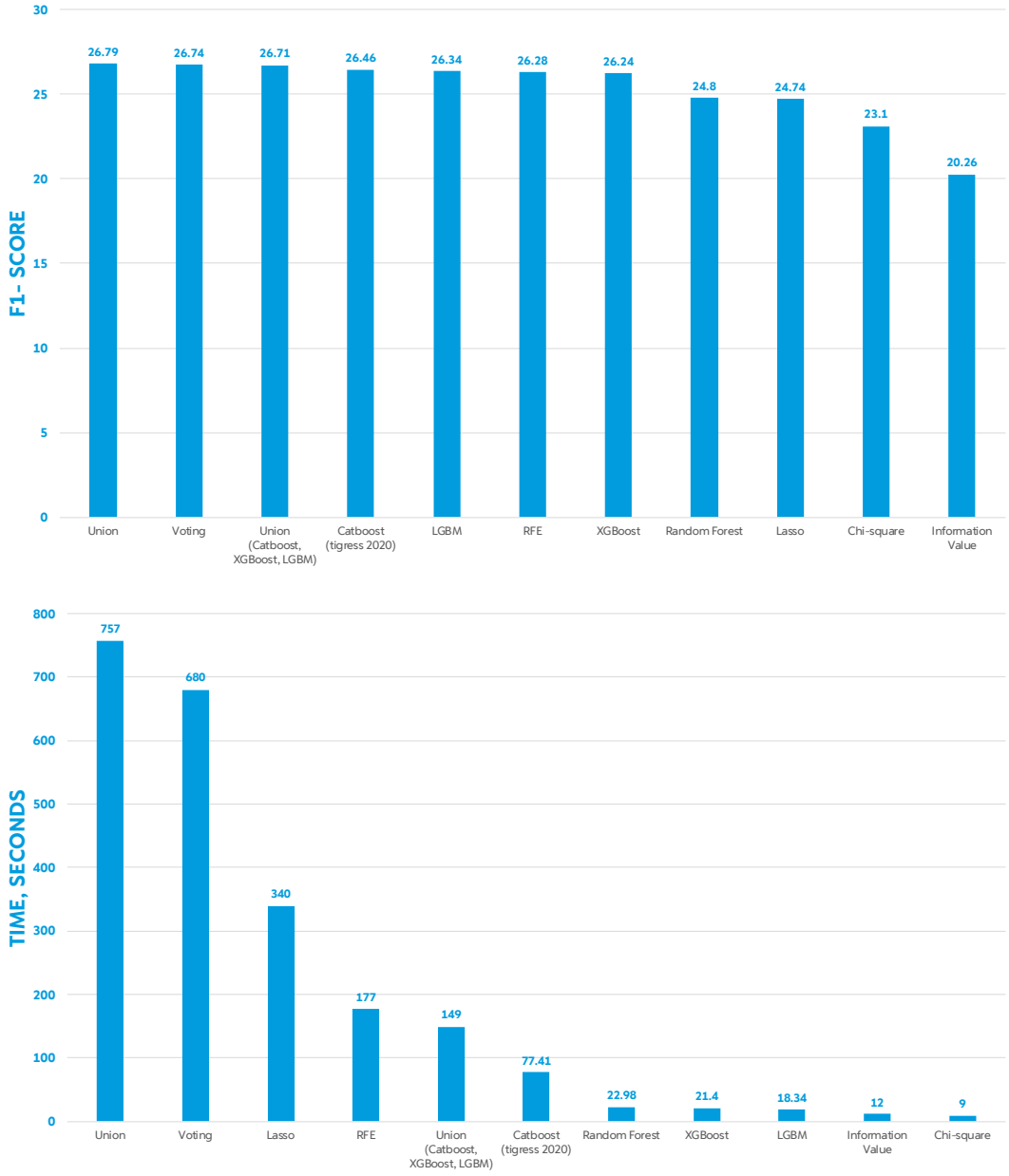


Figure 1

F1-score (A) and computational complexity for different features selection (B)

Based on internal PayPal data, 2020

Conclusion

In this paper we presented an empirical evaluation of different feature selection methods and compared the results on a sample data set. We also proposed two new approaches to select the features: union and voting selector, which combine multiple feature selection algorithms to select the best features.

These two methods were shown to provide better model performance compared to individual methods. Tree-based embedded methods such as Catboost and LightGBM perform effectively, both in terms of model performance and computational complexity.

The results achieved in this experiment can help data scientists decide which feature selection method and classifier to use to implement a prediction task. We consider all these techniques to select the features to help build effective fraud detection models.

PayPal

About PayPal

PayPal has remained at the forefront of the digital payment revolution for more than 20 years. By leveraging technology to make financial services and commerce more convenient, affordable, and secure, the PayPal platform is empowering more than 330 million consumers and merchants in more than 200 markets to join and thrive in the global economy.