Default of credit card clients

Abstract

Defaults of credit cards are one of the biggest problems for many companies and banks On time and efficient identification of probability of default plays a key role in saving banks from any crisis , particularly in the field of payments. In this project, I built an efficient and accurate model to detect the default credit card of the customers and the model is based on machine learning techniques. Particularly supervised learning algorithms. The model is developed based on classification algorithms including Support vector machine, Naive Bayes and Decision tree. I applied the models after exploring the best features. After I trained the models I evaluated each one using an accuracy metric.

Design

This project is part of the course requirements. Data provided by UCI Machine Learning Repository has been used in this project. Default of credit card clients Data Set to estimate the real probability of default. the binary result of classification - credible or not credible clients using machine learning algorithms would enable understanding of clients.

Data

This data is set to the case of customers' default payments in Taiwan and compares the predictive accuracy of the probability of default. The data includes 24 columns. 23 columns contain independent variables and one dependent variable which will predict default payment next month. And 30000 instances .the dataset columns mentioned as Xn . In following what each one refers to . as the UCI site mentioned

Algorithms

Feature Engineering:

Applied a Standard Scaler method to scaling data before applying the model.

Splitting data into train data and test data, 70% of data is used to train models during the learning process.

Checking missing values

Models

Decision Tree Classifier (DTC) , Naive Bayes and Support Vector Machine Classifier (SVM) . SVM gives higher accuracy than DTC.

Model Evaluation and Selection

The entire training dataset of records was split into 70/30 train vs. testing and all scores reported below were calculated.

The official metric was classification rate (accuracy); however, class weights were included to improve performance against F1 score.

Decision Tree Classifier:

Accuracy 0.727

	precision	recall	f1-score	support
0	0.83	0.81	0.82	7040
1	0.38	0.41	0.40	1960
accuracy			0.73	9000
macro avg	0.61	0.61	0.61	9000
weighted avg	0.74	0.73	0.73	9000

For SVM:

Accuracy: 0.818

	precision	recall	f1-score	support
0	0.84	0.96	0.89	7040
1	0.67	0.33	0.44	1960
accuracy			0.82	9000
macro avg	0.75	0.64	0.67	9000
weighted avg	0.80	0.82	0.79	9000

Naive Bayes:

Accuracy: 0.692

	precision	recall	f1-score	support
0	0.88	0.70	0.78	7040
1	0.38	0.66	0.48	1960
accuracy			0.69	9000
macro avg	0.63	0.68	0.63	9000
weighted avg	0.77	0.69	0.72	9000

```
Accuracy score is:  0.6923333333333334
```

Tools

Numpy and Pandas for data manipulation.
Scikit-learn for modeling.
Matplotlib and Seaborn for plotting and visualizations.
Jupyter Notebook to write code.

Communication

Here is the presentation made for this project ,and for more details can see README.