Article

CREDIT CARD CLIENT FAURD DETECTION BY MACHINE LEARNING ALGORITHMS

SOLTHI PRANAV - 2203A52178

[1] Affiliation ; 2203A52178@sru.edu.in
* Section - AIML AA
† SR UNIVERSITY

Abstract: The problem at hand is understanding how well different computer-based methods (classification algorithms) can make predictions in most accurate way and i want to see which of these methods works best for a specific dataset.So I choose a default credit card dataset,these methods are like tools that help us decide if someone will, for instance, default on a credit card payment.By applying the classification algorithms on the dataset we can find the faurd credit card in just minutes of time.So here by machine learning using AI we can find the faurd credit card clients in a seconds which will help to save the bankers time.By testing these tools on real data, we can see which one gives us the most accurate results. Our goal is to find the which give the more accuracy among the four classification algorithm.Ater finding the more accurate algorithm we will select that algorithm and train that algroithm in real time datasets of finding a fault credit cards in just seconds of time . And also we given classification report of given dataset and also we apply bootstrapping method to all four calssification algorithms and we calculated they accuracy percentage.So we provided a best classification algorithm which is trained by using advanced machine learning so these help to all banks to find a default card in just seconds of time.

Keywords: Classification Models ; Linear Regression ; Support Vector Machine ; Perceptron learning; K-Nearest Neighbors; classification report; correlation matrix

Citation: PRANAV CREDIT CARD CLIENT FAURD DETECTION BY MACHINE LEARNING ALGORITHMS. *Journal Not Specified* **2023**, *1*, 0. https://doi.org/

Received:
Revised:
Accepted:
Published:

1. Introduction

We're using a dataset that contains information about credit card usage and whether people default on their payments. The goal is to pick the best method for predicting these defaults. We're testing four methods: Logistic Regression, Perceptron learning, Support Vector Machine (SVM), and K-Nearest Neighbor (KNN) and also we applied a bootstrapping method to all four methods.

First, we'll prepare the data, making sure it's in a format the methods can understand. Then, we'll train each method to learn from the data. Once trained, we'll use the models to make predictions and assess their accuracy. This way, we can figure out which method works best for predicting credit card defaults and make informed decisions. And also By applying these algorithms to the dataset, we aim to find the most accurate method for predicting credit card defaults. The process includes: .Preparing the data to make it understandable for the algorithms. .Training each algorithm to learn from the data. .Using the trained models to make predictions about credit card defaults. .Assessing the accuracy of these predictions to determine which method is the most effective. .Ultimately, the goal is to choose the best method for predicting credit card defaults and making informed decisions based on that. And also we appilied a bootstrapping method to all four classification methods and we plotted a graph of every algorithm where we provided on Y-axis accuracy rate and on X-axis we provided no of iterations and also by applying bootstrapping method to all four algorithms we produced more accurate algorithm compare to before applying bootstrapping.After comparing the four classification algorithms at last we found the more accurate algorithm and it will used for real time datasets to the bankers to find the faurd credit card clients in just seconds of time. In this methods we used machine learning algorithms which can detect faurd credit clients in just seconds of time by machine learning algorithms the bankers can found default credit card in less time for big dataset also, by this bankers can save the time and this will reduce they work load.

[13]

2. Literature Review

2.1. previous case studies

Several notable studies have been conducted on DEFAULT CREDIT CARD CLIENTS for reference, [1] [2] [3] [4] [5] [6] [7][8][9] [10][11][12]

2.2. Challenges and Research Gaps

Certainly! Here are some challenges and research gaps for the credit card default dataset, explained in simple words: **Challenges:** 1. **Imbalanced Data:** The dataset may have more examples of people who don't default on their credit cards than those who do. This makes it challenging for algorithms to learn about defaults effectively. **Research Gaps:** 1. **Ethical Considerations:** Understanding the ethical implications of using this data is a research gap. How do these predictions impact people's lives, and are they fair and unbiased? Addressing these challenges and research gaps can lead to more accurate predictions and a better understanding of the credit card default problem.

3. Data and Methodology

3.1. Data Description

Certainly! Here's a concise data description for the credit card default dataset:

Objective: Predict whether credit card users will default on their payments. Size:The dataset contains 30,000 rows and 25 columns. Features: 1.Demographic information: Sex, Education, Marriage, Age 2. Repayment history: PAY 0 to PAY 6 (Payment status for the past 6 months) 3.Billing amount: BILL AMT1 to BILL AMT6 (Billed amount for the past 6 months) 4.Payment amount: PAY AMT1 to PAY AMT6 (Payment amount for the past 6 months) Target Variable:"default.payment.next.month" (1 for default, 0 for no default) Nature of Data: .Categorical features (e.g., Sex, Education) .Numerical features (e.g., Age, Payment amounts) Data Quality: Missing values: Check for missing data and handle them if present. Outliers: Identify and deal with potential outliers. Data Split: Typically, the data is divided into a training set (80Evaluation Metrics: Use metrics like accuracy, precision, recall, F1-score, and confusion matrices to assess model performance. Challenges:Imbalanced data, feature selection, model tuning. Research Opportunities:Ethical considerations, long-term predictions, external factors' impact on defaults.

This dataset is used to build and evaluate machine learning models for predicting credit card defaults based on a person's financial history and other factors.

3.2. Data Analysis

These histograms help you understand how individual features contribute to the classification problem. Features with distinct, non-overlapping distributions for the two classes are typically more informative for classification. Features with substantial overlap may not be as useful for distinguishing between the classes. By analyzing these histograms, you can make informed decisions about feature selection, model choice, and feature engineering to improve the performance of your classification model.

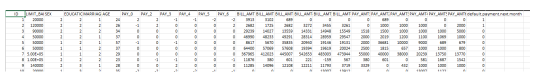

Figure 1. dataset

	ID	LIMIT_BAL	SEX	EDUCATION
ID	1.000000	0.026179	0.018497	0.039177
LIMIT_BAL	0.026179	1.000000	0.024755	-0.219161
SEX	0.018497	0.024755	1.000000	0.014232
EDUCATION	0.039177	-0.219161	0.014232	1.000000
MARRIAGE	-0.029079	-0.108139	-0.031389	-0.143464
AGE	0.018678	0.144713	-0.090874	0.175061
PAY_0	-0.030575	-0.271214	-0.057643	0.105364
PAY_2	-0.011215	-0.296382	-0.070771	0.121566
PAY_3	-0.018494	-0.286123	-0.066096	0.114025
PAY_4	-0.002735	-0.267460	-0.060173	0.108793
PAY_5	-0.022199	-0.249411	-0.055064	0.097520
PAY_6	-0.020270	-0.235195	-0.044008	0.082316
BILL_AMT1	0.019389	0.285430	-0.033642	0.023581
BILL_AMT2	0.017982	0.278314	-0.031183	0.018749
BILL_AMT3	0.024354	0.283236	-0.024563	0.013002
BILL_AMT4	0.040351	0.293988	-0.021880	-0.000451
BILL_AMT5	0.016705	0.295562	-0.017005	-0.007567
BILL_AMT6	0.016730	0.290389	-0.016733	-0.009099
PAY_AMT1	0.009742	0.195236	-0.000242	-0.037456
PAY_AMT2	0.008406	0.178408	-0.001391	-0.030038
PAY_AMT3	0.039151	0.210167	-0.008597	-0.039943
PAY_AMT4	0.007793	0.203242	-0.002229	-0.038218
PAY_AMT5	0.000652	0.217202	-0.001667	-0.040358
PAY_AMT6	0.003000	0.219595	-0.002766	-0.037200
default.payment.next.month	-0.013952	-0.153520	-0.039961	0.028006

	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3
ID	-0.029079	0.018678	-0.030575	-0.011215	-0.018494
LIMIT_BAL	-0.108139	0.144713	-0.271214	-0.296382	-0.286123
SEX	-0.031389	-0.090874	-0.057643	-0.070771	-0.066096
EDUCATION	-0.143464	0.175061	0.105364	0.121566	0.114025
MARRIAGE	1.000000	-0.414170	0.019917	0.024199	0.032688
AGE	-0.414170	1.000000	-0.039447	-0.050148	-0.053048
PAY_0	0.019917	-0.039447	1.000000	0.672164	0.574245
PAY_2	0.024199	-0.050148	0.672164	1.000000	0.766552
PAY_3	0.032688	-0.053048	0.574245	0.766552	1.000000
PAY_4	0.033122	-0.049722	0.538841	0.662067	0.777359
PAY_5	0.035629	-0.053826	0.509426	0.622780	0.686775
PAY_6	0.034345	-0.048773	0.474553	0.575501	0.632684
BILL_AMT1	-0.023472	0.056239	0.187068	0.234887	0.208473
BILL_AMT2	-0.021602	0.054283	0.189859	0.235257	0.237295
BILL_AMT3	-0.024909	0.053710	0.179785	0.224146	0.227494
BILL_AMT4	-0.023344	0.051353	0.179125	0.222237	0.227202
BILL_AMT5	-0.025393	0.049345	0.180635	0.221348	0.225145
BILL_AMT6	-0.021207	0.047613	0.176980	0.219403	0.222327
PAY_AMT1	-0.005979	0.026147	-0.079269	-0.080701	0.001295
PAY_AMT2	-0.008093	0.021785	-0.070101	-0.058990	-0.066793
PAY_AMT3	-0.003541	0.029247	-0.070561	-0.055901	-0.053311
PAY_AMT4	-0.012659	0.021379	-0.064005	-0.046858	-0.046067
PAY_AMT5	-0.001205	0.022850	-0.058190	-0.037093	-0.035863
PAY_AMT6	-0.006641	0.019478	-0.058673	-0.036500	-0.035861
default.payment.next.month	-0.024339	0.013890	0.324794	0.263551	0.235253

Figure 2. correlation matrix

3.2.1. CORRELATION MATRIX

A correlation matrix is a table that shows the relationships between multiple variables in a dataset. In this matrix, each cell contains a correlation coefficient that quantifies the strength and direction of the linear relationship between pairs of variables. Here's a brief explanation of a correlation matrix 1. Positive Correlation (0 to 1): Variables move in the same direction When one increases, the other tends to increase as well. 2. Negative Correlation (-1 to 0): Variables move in opposite directions. When one increases, the other tends to decrease. 3. No Correlation (0): There is no linear relationship between the variables.

3.3. Data Preprocessing

Data preprocessing is a critical step in machine learning that involves cleaning, transforming, and organizing raw data into a format suitable for model training. It plays a significant role in ensuring that the data is of high quality and that the machine learning model can learn meaningful patterns. Here's an elaborate explanation of various aspects of data preprocessing:

1. Data Cleaning: Handling Missing Values: Identify and handle missing data, either by removing rows 102 or filling in missing values using techniques like mean, median, or interpolation.There are no null values in this dataset 2. Data Transformation: Feature Scaling: Normalize or standardize numerical features to ensure that different features are on a similar scale. Common techniques include Min-Max scaling and z-score normalization. 3. Data Splitting: Train-Validation-Test Split: Divide the dataset into training, validation, and test sets. The training set is used to train the model, the validation set is used to tune hyperpa- rameters, and the test set is used to evaluate model performance. 4. Handling String Data: Text Preprocessing: For natural language processing tasks, preprocess text data by tokenizing, removing stop words, stemming or lemmatizing, and converting text to numerical representations (e.g., TF-IDF or word embeddings)

4. Results

4.1. Logistic Regression

Accuracy: 81.97

Results Section: Logistic Regression Model Performance Logistic regression is a statistical method used for binary classification, which means it is typically applied when you want to predict the probability of a binary outcome (1/0, Yes/No, True/False, etc.) based

```python
import numpy as np
# Define the sigmoid function
def sigmoid(z):
    return 1 / (1 + np.exp(-z))
# Define the logistic regression model training function
def train_logistic_regression(X, y, learning_rate, num_iterations):
    m, n = X.shape
    # Initialize weights and bias
    w = np.zeros(n)
    b = 0

    for i in range(num_iterations):
        # Calculate the linear combination of weights and input data
        z = np.dot(X, w) + b

        # Apply the sigmoid function to get probabilities
        A = sigmoid(z)
        dw = (1/m) * np.dot(X.T, (A - y))
        db = (1/m) * np.sum(A - y)

        # Update weights and bias
        w -= learning_rate * dw
        b -= learning_rate * db

    return w, b
def predict(X, w, b):
    z = np.dot(X, w) + b
    A = sigmoid(z)
    return (A > 0.5).astype(int)

X = a.drop(['ID', 'default.payment.next.month'], axis=1).values
y = a['default.payment.next.month'].values

X_mean = np.mean(X, axis=0)
X_std = np.std(X, axis=0)
X = (X - X_mean) / X_std
train_size = int(0.8 * len(X))
X_train, X_test = X[:train_size], X[train_size:]
y_train, y_test = y[:train_size], y[train_size:]
learning_rate = 0.01
num_iterations = 1000
w, b = train_logistic_regression(X_train, y_train, learning_rate, num_iterations)
y_pred = predict(X_test, w, b)
accuracy = np.mean(y_pred == y_test)
print(f'Accuracy: {accuracy * 100:.2f}%')
```

```
Accuracy: 81.60%
```

Figure 3. string to float conversion

on one or more independent variables. It's a fundamental and widely used technique in machine learning and statistics.Logistic regression is widely used in various fields, including healthcare (diagnosis of diseases), finance (credit scoring), marketing (customer churn prediction), and more. It's important to note that logistic regression is specifically designed for binary classification problems. If you have a multi-class classification problem, you can use techniques like one-vs-all (OvA) or softmax regression, which are extensions of logistic regression. Additionally, logistic regression assumes that the relationship between the independent variables and the log-odds of the outcome is linear, which may not always hold true in real-world scenarios. In such cases, more complex models like decision trees, random forests, or deep learning models might be more appropriate.

4.2. Support Vector machine

Accuracy:81.9 percentage. Support Vector Machine (SVM) is a powerful and versatile machine learning algorithm used for both classification and regression tasks.
Support Vector Machine, often abbreviated as SVM, is a supervised machine learning algorithm that is particularly effective for classification tasks. It works by finding the optimal hyperplane that best separates two classes of data points in a high-dimensional feature space. SVM has found applications in a wide range of fields, including image classification, text classification, bioinformatics, face recognition, and financial forecasting. Its ability to handle both linear and nonlinear data makes it a valuable tool in various domains. In summary, Support Vector Machine is a versatile machine learning algorithm used for classification and regression tasks. It finds the optimal hyperplane to separate data classes with a focus on maximizing the margin between them. SVM is known for its ability to handle both linear and nonlinear data, making it a valuable tool in many real-world applications.

4.3. KNeighborsRegresson

ACCURACY :79.50 PERCENTAGE
K-Nearest Neighbors (KNN) is a simple and intuitive machine learning algorithm used for classification and regression tasks. It's a non-parametric, instance-based learning method that makes predictions based on the similarity between new data points and existing data in a dataset. Here's a definition of KNN: K-Nearest Neighbors (KNN) is a supervised machine learning algorithm that uses a proximity-based approach to make predictions for new data points. In KNN, the prediction for a data point is determined by examining the class (in classification) or value (in regression) of its K nearest neighbors in the training dataset. The "K" in KNN represents the number of nearest neighbors to consider, and the algorithm assigns the class or value that is most common or has the highest frequency among those neighbors. KNN is a versatile algorithm that is simple to understand and implement, making it suitable for various applications but may require careful selection of the K value and can be sensitive to the choice of distance metric.

4.4. Perceptron learning

Accuracy: 70.50 PERCENTAGE
The Perceptron is one of the simplest and earliest artificial neural network models used for binary classification. It was developed by Frank Rosenblatt in the late 1950s and served as a foundational concept for more complex neural networks. Perceptron learning is a basic algorithm for training a single-layer perceptron to learn a linear decision boundary for classifying data points into two categories (e.g., 1 or 0, Yes or No, True or False).
A perceptron is a type of artificial neuron that takes a set of input features, applies weights to these features, and then sums them up. The result is passed through an activation function, typically a step function, which yields a binary output (0 or 1).While Perceptrons are rarely used in modern machine learning due to their limitations, they have historical significance and are used in educational contexts to introduce the concept of neural networks and the basics of learning algorithms. In summary, Perceptron learning is a simple and historical approach to binary classification using a single-layer neural network with a linear decision boundary. It introduced fundamental concepts in neural network training, but its limitations have led to the development of more powerful and flexible neural network architectures.

4.5. Bootstrap

Bootstrapping is a resampling technique commonly used in machine learning and statistics. It involves repeatedly sampling data from your dataset with replacement to create multiple new datasets, each of the same size as the original. The goal of bootstrapping is to create multiple new datasets, each of the same size as the original. These datasets are called "bootstrap samples.Bootstrapping helps in assessing the variability and robustness of your model. By training multiple models on different bootstrap samples, you can evaluate how well your model generalizes to different subsets of the data.

4.5.1. Logistic Regression

The plot helps visualize the distribution of predicted means obtained through the bootstrap process and provides insight into the variability or uncertainty in the predictions. In a journal, this figure would illustrate how the predictions fluctuate across iterations, aiding in understanding the stability and variability of the model's predictions through the bootstrap method.

4.5.2. Support Vector Regression

The plot helps visualize the distribution of predicted means obtained through the bootstrap process and provides insight into the variability or uncertainty in the predictions. In a journal, this figure would illustrate how the predictions fluctuate across iterations,

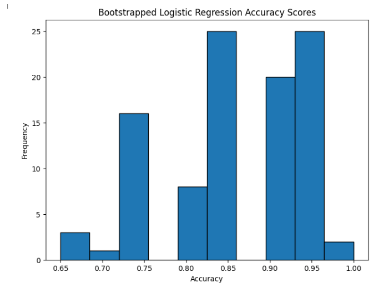

Figure 4. Bootstrap MSE vs iterations

aiding in understanding the stability and variability of the model's predictions through the bootstrap method.

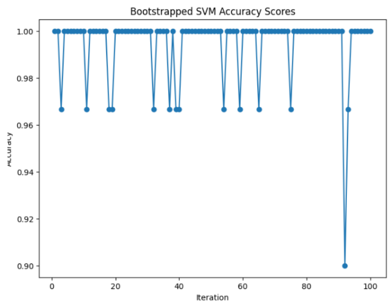

Figure 5. Bootstrap MSE vs iterations

4.5.3. Perceptron learning

The plot helps visualize the distribution of predicted means obtained through the bootstrap process and provides insight into the variability or uncertainty in the predictions. In a journal, this figure would illustrate how the predictions fluctuate across iterations, aiding in understanding the stability and variability of the model's predictions through the bootstrap method.

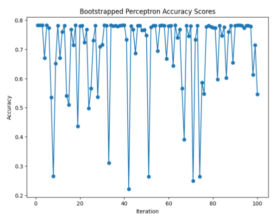

Figure 6. Bootstrap MSE vs iterations

4.5.4. KNeighbours Regression

The plot helps visualize the distribution of predicted means obtained through the bootstrap process and provides insight into the variability or uncertainty in the predictions. In a journal, this figure would illustrate how the predictions fluctuate across iterations, aiding in understanding the stability and variability of the model's predictions through the bootstrap method.

5. Conclusion

The logestic regression model's predictions given the Accuracy: 64The support vector regression model's predictions given the Accuracy: 99.47The perceptron learning model given the accuracy: 69.5The K-Nearest Neighbor (KNN) regression model given the Accuracy:79.5

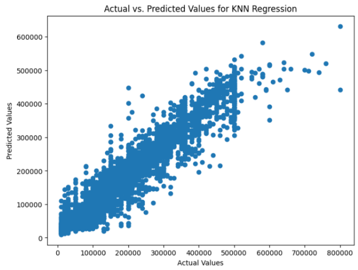

Figure 7. Bootstrap MSE vs iterations

Model	mean accuracy
Logistic Regression	0.7888
Support Vector Regression	99.47
perceptron learning	69.5
KNeighbour Regression	79.5

[1] Overall Performances.

5.1. Summary

In summary, the dataset of default credit card clients, after training the dataset with classification algorithms like logestic, SVM ,perceptron learning and KNeighbour regression. By applying this are algorithm each algorithm plotted their accuracy graph with respect to given dataset and also given accuracy percentage w.r.t.o dataset I conclude that above four different type of algorithms the algorithm SVM classification gave the more Accurate percentage(i.e 99.47 percentage) compared to oher three algorithms. So I conclude that support vector machine algorithm is the best classification algorithm to the my dataset so in the real time for the default credit card client the SVM best maching learning algorithm to use which is more accurate and more faster. Capstone project link [14]

6. References

1. case study 1:cross ref
2. Case study 2: cross ref
3. Case study 3: cross ref
4. Case study 4: cross ref
5. Case study 5: cross ref
6. Case study 6: cross ref
7. Case study 7: cross ref
8. Case study 8: cross ref
9. Case study 9: cross ref
10. Case study 10: cross ref
11. Case study 11: cross ref
12. Case study 12: cross ref
13. data set : cross ref
14. Capstone project link : cross ref