Credit Card Approval Prediction

Batch - 8
Guide: Dr.J.Hyma

Rakesh Naidu	121810305023
Mani Chandra	121810305032
Shashidhar	121810305043
Vishnu	121810305008

Abstract

The increased credit card defaulters have forced the companies to think carefully before the approval of credit applications. Credit card companies usually use their judgment to determine whether a credit card should be issued to the customer satisfying certain criteria. Some machine learning algorithms have also been used to support the decision.

Abstract

The performance of the built model is compared with the multiple traditional machine learning algorithms: Naive Baye's , K Nearest Neighbours, Decision Tree, Random Forest, XG Boost . Many other results show that the overall performance of our deep learning model is slightly better than that of the other models.

Abstract

This Project is a machine learning solution to automate the process of credit card approval to an account. The model is trained with multiple attributes and the accuracy is measured with both train and test data.We will use the Credit Approval Dataset which is a collection of credit card applications and the credit approval decisions.

Introduction

The accurate assessment of consumer credit risk is of uttermost importance for lending organizations. Credit scoring is a widely used technique that helps financial institutions evaluates the likelihood for a credit applicant to default on the financial obligation and decide whether to grant credit or not.The precise judgment of the creditworthiness of applicants allows financial institutions to increase the volume of granted credit while minimizing possible losses.

Introduction

The credit industry has experienced a tremendous growth in the past few decades.The increased number of potential applicants impelled the development of sophisticated techniques that automate the credit approval procedure and supervise the financial health of the borrower.

Introduction

The large volume of loan portfolios also imply that modest improvements in scoring accuracy may result in significant savings for financial institutions (West, 2000). The goal of a credit scoring model is to classify credit applicants into two classes: the “good credit” class that is liable to reimburse the financial obligation and the “bad credit” class that should be denied credit due to the high probability of defaulting on the financial obligation.

Introduction

The goal here is to build an end to end automated Machine Learning solution where a user will be able to predict whether a bank customer should be approved for attaining the credit card or not. The user is only need to give the value of feature variables and the model will able to predict the binary outcome (Approve/ Not Approve).

Introduction

The model will be able take care of all intermediate functionalities like cross validation, hyper parameter tuning, algorithm selection etc.

This project shall be delivered in two phases:

Phase 1: All the functionalities with PyPi packages.
Phase 2: Integration of UI to all the functionalities.

Note: All the code will be written in python version 3.6

About Dataset

This file concerns credit card applications. All attribute names and values have been changed to meaningless symbols to protect confidentiality of the data.

This dataset is interesting because there is a good mix of attributes -- continuous, nominal with small numbers of values, and nominal with larger numbers of values. There are also a few missing values.

Data Source: UCI, Kaggle

Attribute Information

A1	b, a	Gender
A2	continuous	Age
A3	continuous	Debt
A4	u, y, l, t	Marital status
A5	g, p, gg	Bank
A6	c, d, cc, i, j, k, m, r, q, w, x, e, aa, ff	Education Level
A7	v, h, bb, j, n, z, dd, ff, o	Ethnicity
A8	continuous	Years Employed

Attribute Information

A9	t, f	Prior default
A10	t, f	Employed
A11	continuous	Credit score
A12	t, f	Drivers license
A13	g, p, s	Citizen
A14	continuous	Zip Code
A15	continuous	Income
A16	+,- (class attribute)	Approved

Literature Review

Credit card has evolved to a great level in banking industry. Each banking system consists of an enormous number of datasets to carry customer's transactions of their credit cards. So, banks would be in need of customer profiling. Customer Profiling in banks cognizes the issuer's decisions about whom to give banking facilities and what credit limit to be provided.

Literature Review

In previous researches, profiling mainly depended on transaction data or demographic data, but in this research, both transaction and demographic data are merged in order to get more accurate results and minimize the possibility of risk occurrence.By using the best techniques, it leads to improvement in accuracy and helps banks to have high profitability through customer satisfaction by focusing on the valuable customer (companies) which are considered as the main engine in the bank's profitability.

Literature Review

This study used k-mean, improved k-mean, fuzzy c-means and neural networks. The used dataset is labeled and for neural network classification creating a new label as a target becomes the main aspect of this study, which helps to reduce the execution time of clustering process and provide the best results with accuracy.

Problem Identification & Objectives

The proposed project is built end to end. Starting from Data Preprocessing to Deployment. This project includes the features like:

a) Statistical analysis
b) Hyper parameter tuning
c) Best algorithm selection
d) Deployment in Heroku using flask.

Problem Identification & Objectives

The main objectives of the proposed project are to:

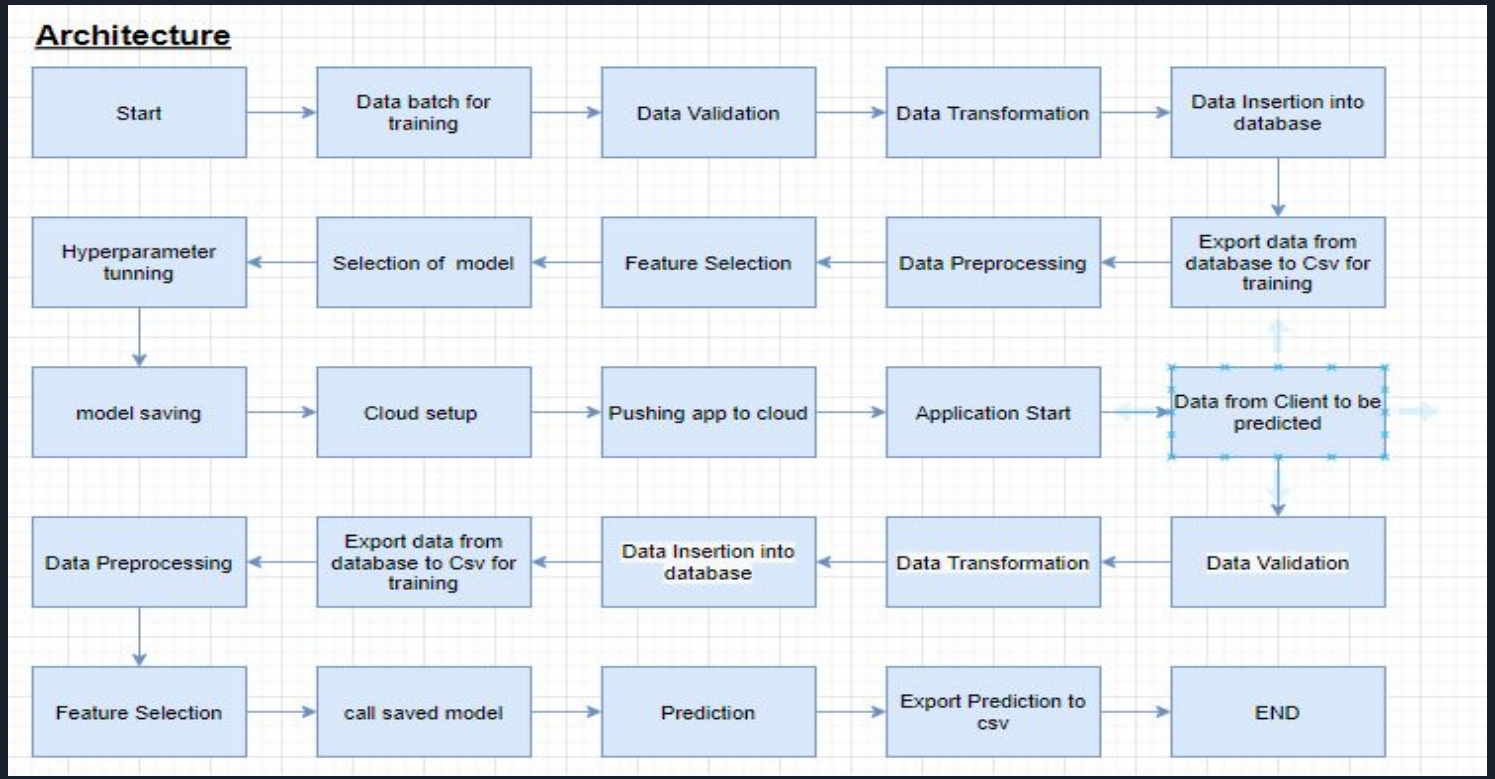
- Increase the accuracy
- Do Exploratory data analysis
- Test the model with different algorithms
- Try different model selection criteria
- Do Hyperparameter tuning
- Deploy the project for easy use

Problem Identification & Objectives

Different technologies or libraries used in the project are:

- numpy, pandas
- matplotlib, seaborn
- scipy, scikit learn
- xgboost
- html, css
- flask, gunicorn

System Methodology



Overview of Technologies

- Numpy

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

Overview of Technologies

- Pandas

Pandas is an open source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named Numpy, which provides support for multi-dimensional arrays. As one of the most popular data wrangling packages, Pandas works well with many other data science modules inside the Python ecosystem, and is typically included in every Python distribution, from those that come with your operating system to commercial vendor distributions like ActiveState's ActivePython

Overview of Technologies

- Matplotlib

Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy. As such, it offers a viable open source alternative to MATLAB. Developers can also use matplotlib's APIs (Application Programming Interfaces) to embed plots in GUI applications.

Overview of Technologies

- Seaborn

Seaborn is an open-source Python library built on top of matplotlib. It is used for data visualization and exploratory data analysis. Seaborn works easily with dataframes and the Pandas library. The graphs created can also be customized easily. Below are a few benefits of Data Visualization.

Overview of Technologies

- Scipy

SciPy is a scientific computation library that uses NumPy underneath. SciPy stands for Scientific Python. It provides more utility functions for optimization, stats and signal processing. Like NumPy, SciPy is open source so we can use it freely. SciPy was created by NumPy's creator Travis Olliphant.

Overview of Technologies

- Scikit learn

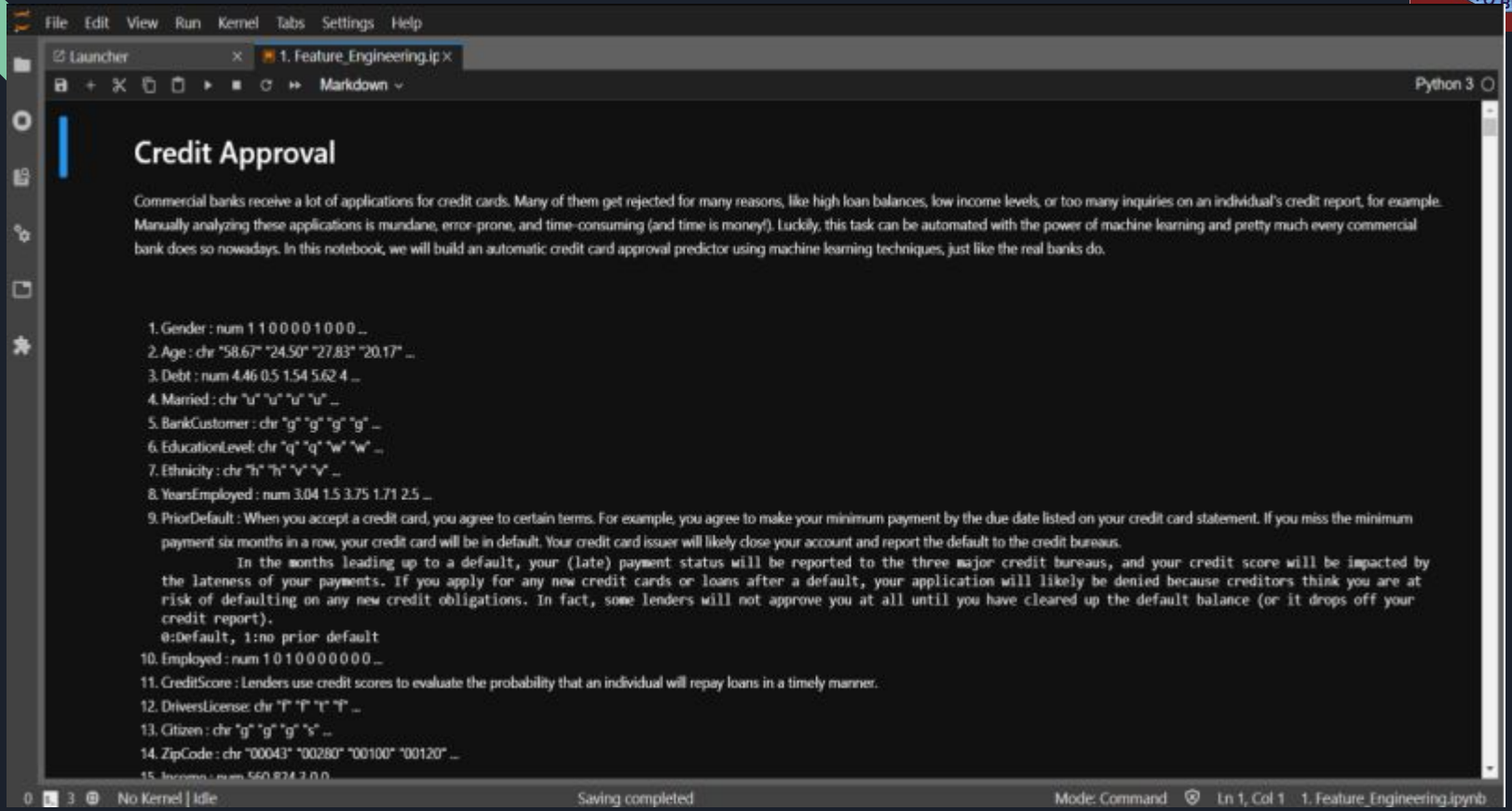
Scikit-learn is a library in Python that provides many unsupervised and supervised learning algorithms. It's built upon some of the technology you might already be familiar with, like NumPy, pandas, and Matplotlib! The functionality that scikit-learn provides include: Regression, including Linear and Logistic Regression Classification, including K-Nearest Neighbors Clustering, including K-Means and K-Means++ Model selection Preprocessing, including Min-Max Normalization

Overview of Technologies

- Xgboost

XGBoost is a tree based ensemble machine learning algorithm which is a scalable machine learning system for tree boosting. XGBoost stands for Extreme Gradient Boosting. It uses more accurate approximations to find the best tree model. In prediction problems involving unstructured data (images, text, etc.) artificial neural networks tend to outperform all other algorithms or frameworks. However, when it comes to small-to-medium structured/tabular data, decision tree based algorithms are considered best-in-class right now.

Implementation



Implementation

15. Income : num 560 824 3 0 0 ...
16. Approved : chr "+" "+" "+" "+" ...

Credit card being held in hand

We'll use the Credit Card Approval dataset from the UCI Machine Learning Repository. The structure of this notebook is as follows:

First, we will start off by loading and viewing the dataset. We will see that the dataset has a mixture of both numerical and non-numerical features, that it contains values from different ranges, plus that it contains a number of missing entries. We will have to preprocess the dataset to ensure the machine learning model we choose can make good predictions. After our data is in good shape, we will do some exploratory data analysis to build our intuitions. Finally, we will build a machine learning model that can predict if an individual's application for a credit card will be accepted. First, loading and viewing the dataset. We find that since this data is confidential, the contributor of the dataset has anonymized the feature names.

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

## Display all the columns of the dataframe
pd.pandas.set_option('display.max_columns',None)
```

[2]:
```
import warnings
warnings.filterwarnings('ignore')
```

[3]:
```
dataset = pd.read_csv('crx.data',header=None)
dataset.head()
```

[3]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	b	30.83	0.000	u	g	w	v	1.25	t	t	1	f	g	00202	0	+

3 No Kernel | Idle Saving completed Mode: Command Ln 1, Col 1 1. Feature_Engineering.ipynb

Implementation

```
[4]: dataset.columns
```

```
[4]: Int64Index([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15], dtype='int64')
```

The output may appear a bit confusing at its first sight, but let's try to figure out the most important features of a credit card application. The features of this dataset have been anonymized to protect the privacy, but Analysis of Credit Approval Data blog gives us a pretty good overview of the probable features. The probable features in a typical credit card application are Gender, Age, Debt, Married, BankCustomer, EducationLevel, Ethnicity, YearsEmployed, PriorDefault, Employed, CreditScore, DriversLicense, Citizen, ZipCode, Income and finally the ApprovalStatus. This gives us a pretty good starting point, and we can map these features with respect to the columns in the output.

As we can see from our first glance at the data, the dataset has a mixture of numerical and non-numerical features. This can be fixed with some preprocessing, but before we do that, let's learn about the dataset a bit more to see if there are other dataset issues that need to be fixed.

```
[5]: columns = ["Gender","Age","Debt","Married","BankCustomer","EducationLevel","Ethnicity","YearsEmployed","PriorDefault","Employed","CreditScore","Dr
```

```
[6]: dataset.columns = columns
```

```
[7]: dataset.head()
```

[7]:

	Gender	Age	Debt	Married	BankCustomer	EducationLevel	Ethnicity	YearsEmployed	PriorDefault	Employed	CreditScore	DriversLicense	Citizen	ZipCode	Income	Approved
0	b	30.83	0.000	u	g	w	v	1.25	t	t	1	f	g	00202	0	+
1	a	58.67	4.460	u	g	q	h	3.04	t	t	6	f	g	00043	560	+
2	a	24.50	0.500	u	g	q	h	1.50	t	f	0	f	g	00280	824	+
3	b	27.83	1.540	u	g	w	v	3.75	t	t	5	t	g	00100	3	+
4	b	20.17	5.625	u	g	w	v	1.71	t	f	0	f	s	00120	0	+

Implementation

```
[4]: dataset.columns
```

```
[4]: Int64Index([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15], dtype='int64')
```

The output may appear a bit confusing at its first sight, but let's try to figure out the most important features of a credit card application. The features of this dataset have been anonymized to protect the privacy, but Analysis of Credit Approval Data blog gives us a pretty good overview of the probable features. The probable features in a typical credit card application are Gender, Age, Debt, Married, BankCustomer, EducationLevel, Ethnicity, YearsEmployed, PriorDefault, Employed, CreditScore, DriversLicense, Citizen, ZipCode, Income and finally the ApprovalStatus. This gives us a pretty good starting point, and we can map these features with respect to the columns in the output.

As we can see from our first glance at the data, the dataset has a mixture of numerical and non-numerical features. This can be fixed with some preprocessing, but before we do that, let's learn about the dataset a bit more to see if there are other dataset issues that need to be fixed.

```
[5]: columns = ["Gender","Age","Debt","Married","BankCustomer","EducationLevel","Ethnicity","YearsEmployed","PriorDefault","Employed","CreditScore","Dr
```

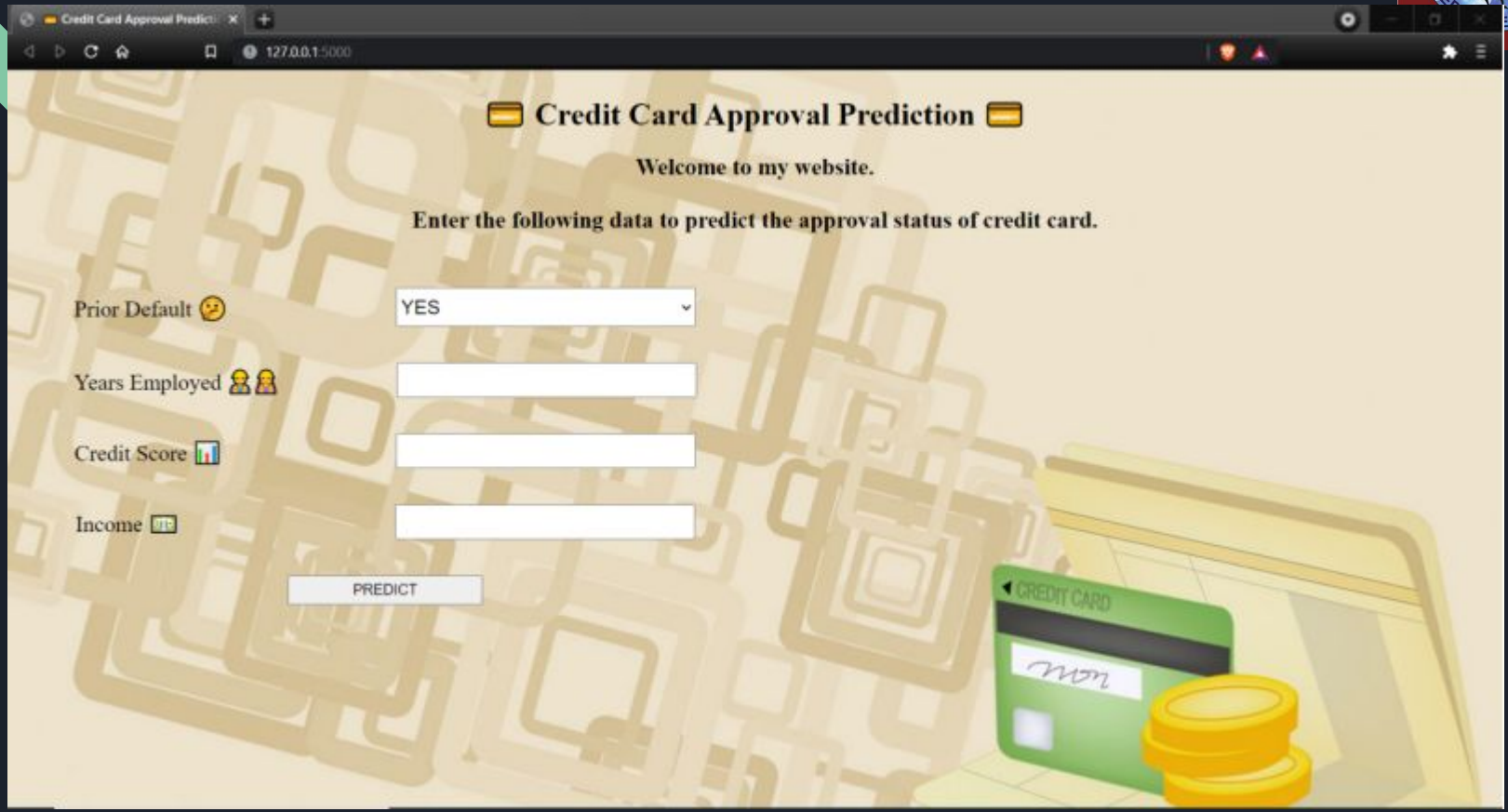
```
[6]: dataset.columns = columns
```

```
[7]: dataset.head()
```

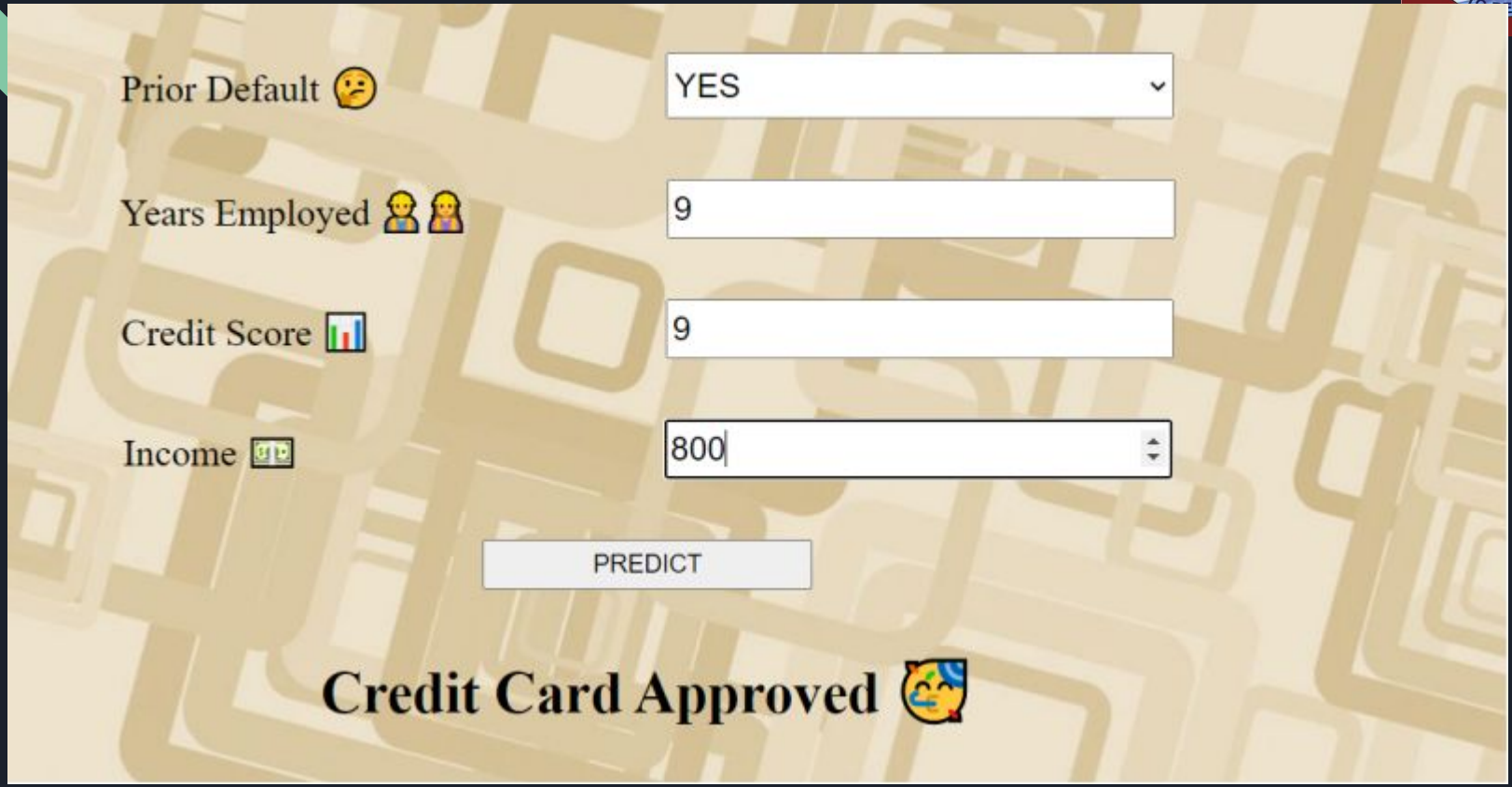
[7]:

	Gender	Age	Debt	Married	BankCustomer	EducationLevel	Ethnicity	YearsEmployed	PriorDefault	Employed	CreditScore	DriversLicense	Citizen	ZipCode	Income	Approved
0	b	30.83	0.000	u	g	w	v	1.25	t	t	1	f	g	00202	0	+
1	a	58.67	4.460	u	g	q	h	3.04	t	t	6	f	g	00043	560	+
2	a	24.50	0.500	u	g	q	h	1.50	t	f	0	f	g	00280	824	+
3	b	27.83	1.540	u	g	w	v	3.75	t	t	5	t	g	00100	3	+
4	b	20.17	5.625	u	g	w	v	1.71	t	f	0	f	s	00120	0	+

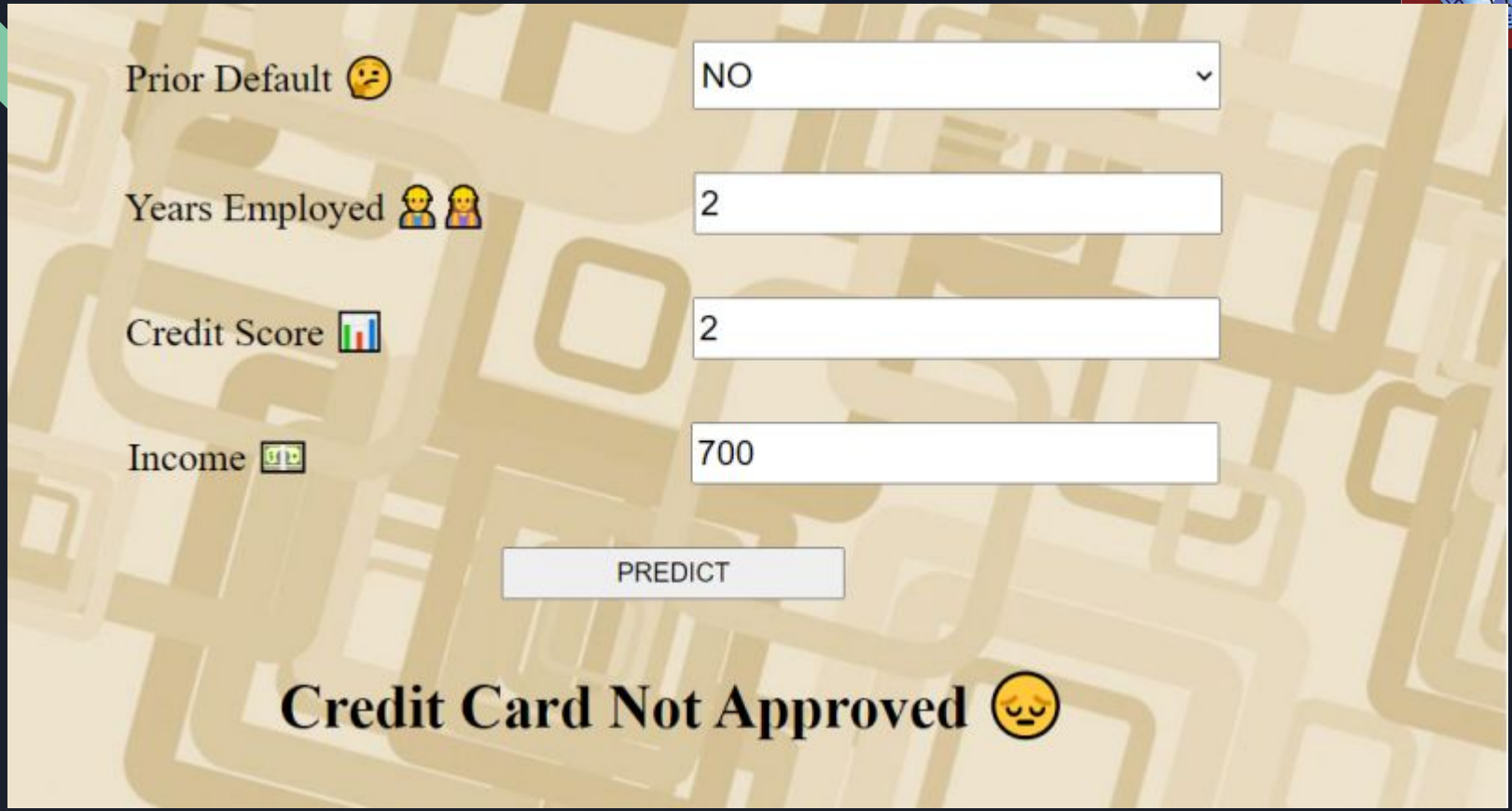
Implementation



Implementation



Implementation



Implementation

System requirements

For Model Training	For Model Testing
• 8 GB RAM	• 4 GB RAM
• 2 GB of Hard Disk Space	• 2 GB of Hard Disk Space
• Intel Core i5 Processor	• Intel Core i5 Processor

Note: These are just Recommended.

Results and Discussions

The accuracy of predicting the credit card is approved and credit card is not approved are almost the same 0.88 and 0.89. The classifier conservatively predicts that the status of credit card approval which will decrease a lost of work for the banks.

Results and Discussions

Accuracy Measure	Value
Specificity	0.91
Recall	0.84
Precision	0.88
F-Measure	0.86
Accuracy	0.88

Conclusion and Future Scope

As of now the project has the models like KNN, Naive Bayes, Decision Tree, Random Forest, Xgboost. Neural networks can be used to get a better accuracy model. Neural network models can lead to give accuracies close to 98%.

References

1. https://www.ijrar.org/papers/IJRAR190B030.pdf
2. https://www.ijeat.org/wp-content/uploads/papers/v9i4/D7293049420.pdf
3. https://www.researchgate.net/publication/321002603_Credit_Approval_Analysis_using_R
4. https://rstudio-pubs-static.s3.amazonaws.com/73039_9946de135c0a49daa7a0a9eda4a67a72.html

Thank You