

Connor Della-Savia
Spencer Keene
COSC4426

Machine Learning Report

Proposal	1
Introduction	1
Background/Literature Review	1
Datasets	2
About	2
Data Summary	2
Manipulating Data	3
Converting Classifiers	3
Results	4
Model	4
Evaluation	4
Discussion	5
Correlation	5
Pairplot	6
Website	7
References	9

Proposal

For the final project in this course our group would like to make a supervised classification machine. The machine will be used to predict whether or not a candidate is eligible for a loan. Two csv files will be used from the website <https://www.kaggle.com/>. One of which will be a file full of information of successful candidates and the other csv will contain unsuccessful candidates. Our machine will use this data to perform binary classification and classify a potential candidate as either successful to receive the loan, or unsuccessful.

Introduction

Machine learning as defined by Oxford languages is “the use and development of computer systems that are able to learn and adapt without following explicit instructions, by using algorithms and statistical models to analyze and draw inferences from patterns in data.” In this report we outline our implementation of machine learning on a provided dataset. The dataset in this case is information that is used to predict whether or not a client is granted for a loan. Since the outcome of our model will be a yes or no it is considered a binary classifier. For our model we implemented a neural network to determine the outcome of our data. The entire program is written in python 3 and uses tools from; pandas, matplotlib, seaborn, numpy, sklearn and tensorflow.

Background/Literature Review

Banks give out money to successful applicants which is called a loan. The bank determines the eligibility of a loan through information that is provided by a client. The goal of our machine is to predict if a client will be eligible for a loan before going to a financial advisor. For our machine the client will need to provide 11 different types of personal data so that our machine can guess if a client will be eligible. Furthermore, the trained model will be easily accessible online such that anyone in the public may use the model.

Datasets

About

The dataset we used to train our model was provided to us by www.kaggle.com. The website allowed us to download a dataset that contains 2 .csv files. One file is called loan-test.csv and the other is loan-train.csv. Unfortunately, loan-test.csv did not contain the label column to determine if the client was eligible for the loan so we opted to use loan-train.csv which had the label. Our dataset originally contains 13 columns by default. The column names are

- Loan_ID
- Gender
- Married
- Dependents
- Education
- Self_Employed
- ApplicantIncome
- CoapplicantIncome
- LoanAmount
- Loan_Amount_Term
- Credit_History
- Property_Area
- Loan_Status

Data Summary

Each row in the data is a different client which is labeled with Loan_ID. Loan_Status shows if the client was eligible for a loan with “Y” meaning yes and “N” meaning no. The Gender column is a string value that says either “Male” or “Female” which is the gender of a client. Married and Self_Employed columns are both a “Yes” or “No” string value to determine if the client is married and/or is Self_Employed. Dependents column shows either a 1, 2 or 3+ to determine the amount of dependents the client has. Our dataset also contains ApplicantIncome and CoapplicantIncome which is how much the client and Coapplicant make if totalled. Credit_History is a binary value to represent if the client has previous credit history. Property_Area shows where the client’s property is located with a given string that is either “Urban”, “Rural” or

“Semiurban”. Finally, the LoanAmount and Loan_Amount_Term give an integer value that represents how much money the client will be receiving (LoanAmount) over a given period of time (Loan_Amount_Term).

Manipulating Data

The dataset we used was not formatted to fit a machine and was missing values in some columns so we had to manipulate the data after loading it into a dataframe. Firstly, we modified the Applicant and Coapplicant income to represent an entire year's salary in a float value. A new column was also made called “JointIncome” which is a sum of the Applicant and Coapplicant income. Next, Loan_ID and CoapplicantIncome columns were dropped because they are no longer necessary.

Many columns were missing values which needed to be filled. Credit_History had its missing values replaced with 0 since we assume if the information was not provided there was no history. Any missing values in the Gender column were replaced with “Unknown/Other” since we do not know the client's gender or that the client does not identify as male or female. After, values in the dependents column that contained “3+” were replaced with “3” such that we are now able to turn the column into a float value instead of a string. Missing values within the column were replaced with the average. Married and Self_Employed missing values were replaced with “No”. LoanAmount missing values were replaced with the median. Lastly, Loan_Amount_Term missing values were replaced with the mean.

Converting Classifiers

To make the data work with our model it had to be converted into numbers. Binary classifiers had to be converted into binary values. This contains; Education, Married, Self_Employed and Loan_Status to all be converted to either a 1 or a 0. Categorical data was to be encoded with OneHotEncoder. Gender and Property_Area were encoded with One Hot Encoder and fit into the dataframe.

Results

Model

The final model that we used was a sequential keras model. We found that 3 layers gave us the best outcome. The layers consisted of:

- 1 input layer of 16 inputs.
- 1 hidden dense layer of 4 neurons with relu activation function
- 1 output layer of 1 output with sigmoid activation function

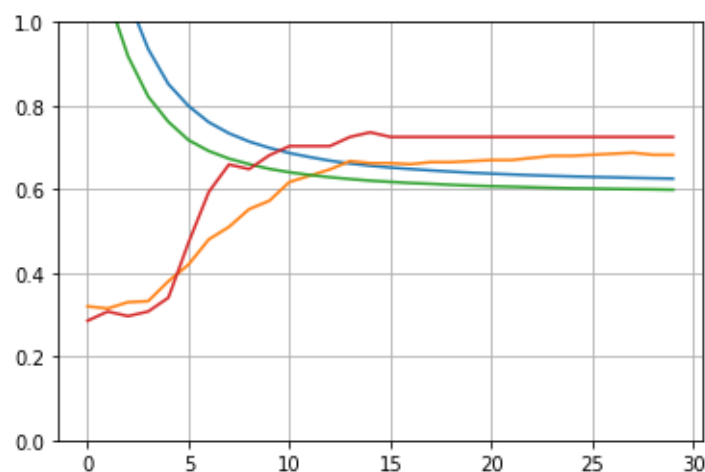
This gave us a total of 73 parameters in the model summary with 68 in the hidden layer and 5 in the output layer.

Our final model was compiled with the loss function called `binary_crossentropy`, optimizer as `adam` and used accuracy metrics. The model fit with the x and y train dataframes used:

- 30 epochs
- 8 batch size
- 91 validation sets

Evaluation

The final evaluation of our model after it was run on our datasets was 0.60 loss and 0.72 accuracy. Shown below is that graph over 30 epochs where; blue is showing the loss, green is showing `val_loss`, orange is showing accuracy and red is showing `val_accuracy`:



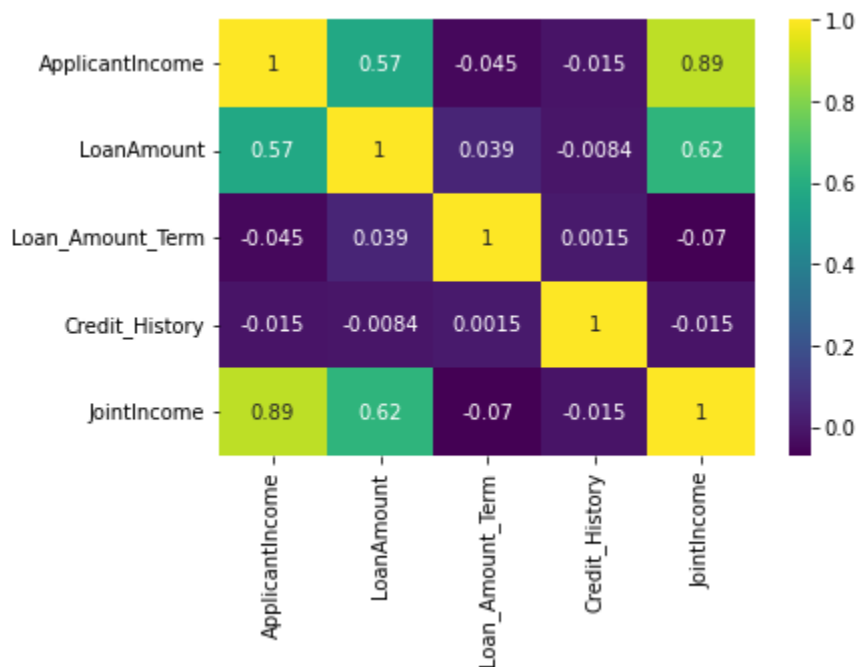
Reflection

Possible things we could have done differently through this project is that we could have used a different dataset that contains more information. If our dataset had more data in it then our model would have been more accurate and which would give a better representation of that applicant's outcome. Furthermore, another thing we could have done is to experiment with the many different features included with keras. Examples of some of the features we could have tested are the different keras layers and model functions.

Discussion

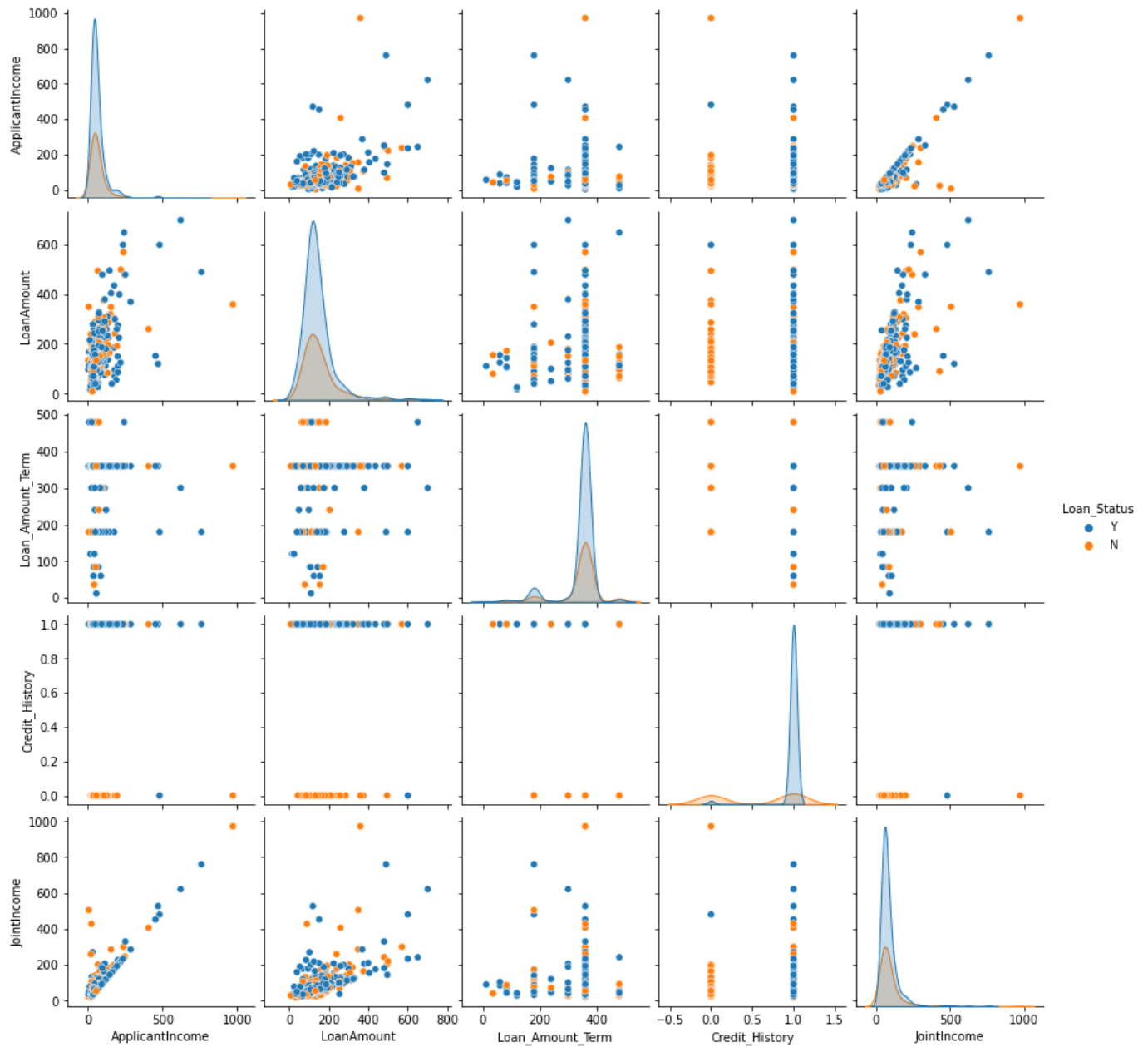
Correlation

A correlation graph can be generated by pandas that shows how columns interact with each other and their interaction on the outcome. Below is the correlation chart for our data:



As shown in the chart, clients with no credit history are much less likely to be granted a loan by the bank. Alternatively, applicants that have a higher amount of joint income are more likely to be accepted. This relationship is clearly displayed in the pairplot (shown below) between LoanAmount and Credit_History.

Pairplot



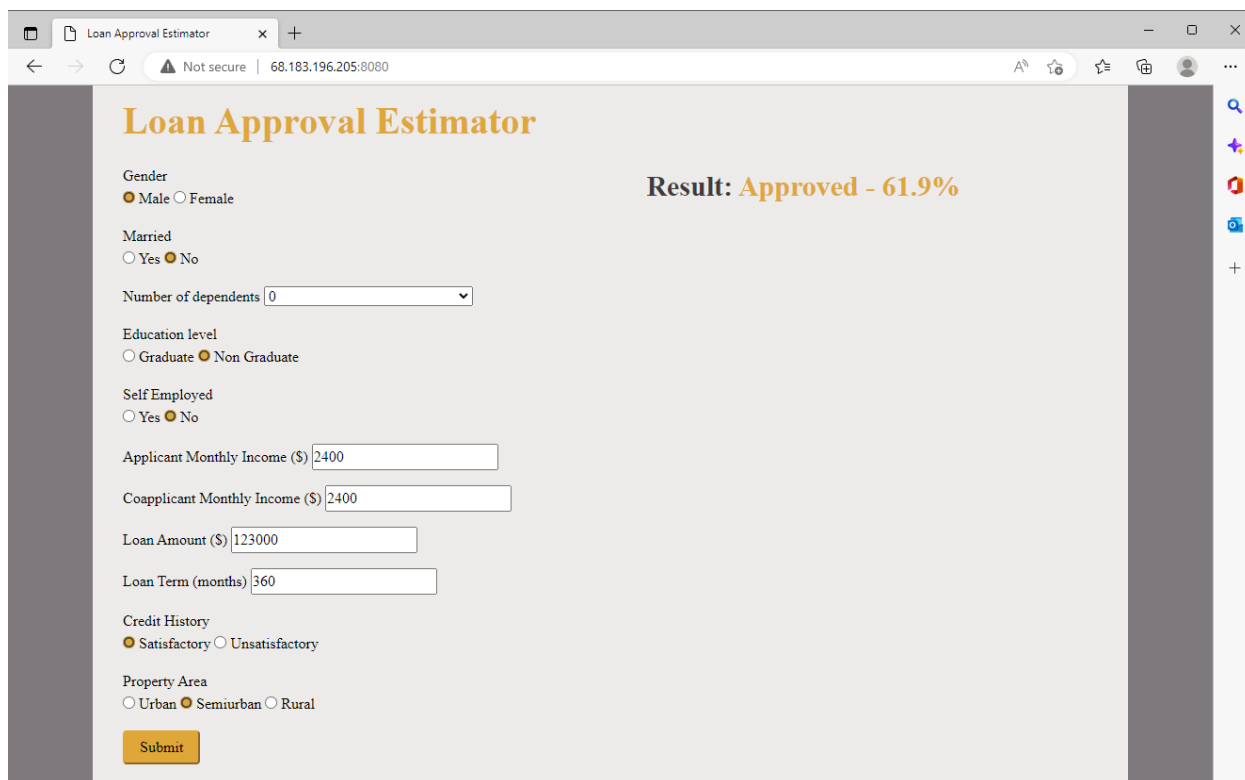
Website

After becoming satisfied with the model we trained, we saved the trained model so that it could be placed in the cloud. We then created a server using the Flask library which would accept POST requests with a JSON body containing the loan application data. This data that comes into the server isn't prepared though so we also saved the pipeline we used when training the model. We loaded the pipeline with joblib and the model using Tensorflow.

When the server receives a request, it extracts the loan application data from the JSON body, prepares it using the pipeline, and passes it into the model before returning the predicted value back. The predicted value, between 0 and 1, represents the likelihood of the application being approved with 1 being approved and 0 being rejected. The website then displays the predicted value as a percentage and states whether the application was accepted or rejected (accepted if prediction > 0.5)

The website and server are both hosted with DigitalOcean cloud computing services. The Droplet (cloud server) running our application is in Toronto.

URL: <http://68.183.196.205:8080/>



The screenshot shows a web browser window with the title "Loan Approval Estimator". The address bar shows the URL "http://68.183.196.205:8080/" and a "Not secure" warning. The website has a light gray background with a dark gray sidebar on the right. The main content area features the title "Loan Approval Estimator" in a large, bold, orange font. Below the title, there is a form with several input fields and radio buttons. The form includes fields for Gender (Male/Female), Married (Yes/No), Number of dependents (a dropdown menu), Education level (Graduate/Non Graduate), Self Employed (Yes/No), Applicant Monthly Income (\$), Coapplicant Monthly Income (\$), Loan Amount (\$), Loan Term (months), Credit History (Satisfactory/Unsatisfactory), and Property Area (Urban/Semiurban/Rural). A "Submit" button is located at the bottom of the form. To the right of the form, the result is displayed in a bold, orange font: "Result: Approved - 61.9%".

Loan Approval Estimator

Gender
☒ Male ☐ Female

Married
☐ Yes ☒ No

Number of dependents

Education level
☐ Graduate ☒ Non Graduate

Self Employed
☐ Yes ☒ No

Applicant Monthly Income (\$)

Coapplicant Monthly Income (\$)

Loan Amount (\$)

Loan Term (months)

Credit History
☒ Satisfactory ☐ Unsatisfactory

Property Area
☐ Urban ☒ Semiurban ☐ Rural

Result: Approved - 61.9%

Loan Approval Estimator

Result: Not Approved - 33.2%

Gender
☐ Male ☒ Female

Married
☐ Yes ☒ No

Number of dependents

Education level
☐ Graduate ☒ Non Graduate

Self Employed
☐ Yes ☒ No

Applicant Monthly Income (\$)

Coapplicant Monthly Income (\$)

Loan Amount (\$)

Loan Term (months)

Credit History
☐ Satisfactory ☒ Unsatisfactory

Property Area
☒ Urban ☐ Semiurban ☐ Rural

Loan Approval Estimator

Gender
☐ Male ☒ Female

Married
☐ Yes ☒ No

Number of dependents

Education level
☐ Graduate ☒ Non Graduate

Self Employed
☐ Yes ☒ No

Applicant Monthly Income (\$)

Coapplicant Monthly Income (\$)

Loan Amount (\$)

Loan Term (months)

Credit History
☐ Satisfactory ☒ Unsatisfactory

Property Area
☒ Urban ☐ Semiurban ☐ Rural

Result: Not Approved - 33.2%

References

Machine learning definition: <https://languages.oup.com/google-dictionary-en/>

Dataset: <https://www.kaggle.com/>

Online Hosting: <https://www.digitalocean.com/>