

Loan predictive analysis and credit card fraud detection using artificial neural networks

¹Prof. Bharat S. Dhak, ²Ms. Trisha K. Sinha, ³Ms. Sakshi N. Atram, ⁴Ms. Pooja A. Halbe, ⁵Ms. Mansi B. Patiyee

¹Professor, ^{2,3,4,5}Student
Computer Science and Engineering
Nagpur, India

Abstract: In this project, we have built a model on how bank management team would like To build and train a simple, deep neural network model to predict the likelihood off customers buying personal loans based on their features, such as their age, experience, income, family education and credit card information as well. And you can simply apply this project to predict customer's credit worthiness and also increase the effectiveness off the bank marketing strategy. In the second module, we have built Credit Card Fraud Detection using artificial neural network Algorithm This project requires basic python programming and basic knowledge of machine learning as well. We divided the project into a series off manageable cells. The bank management team would like to build, train and deploy a simple, deep neuron network model that will be able to predict the likelihood off liability customers. And when we say liability customers, these are depositor customers. These are people or customers who deposit money in the bank. And simply the bank would like to identify those customers and kind of, like make them by personal loans. So the idea here is to try to essentially make money. We want them to issue them loans so the bank would be able to charge them interest and the point is, we wanted to target those customers. So to do that, we want to know their age, their experience, their income level, their location, family education level, existing mortgage if they have an existing mortgage with the bank or not. And we also want to know if they have a credit card with our bank account or not, because all these factors play a big role in the acceptance off customers. To these marketing strategies, which is again asking them to buy our personal loans. So, there is a company or a startup named Lindo included a link here the website and it's actually leading startup that uses advanced machine learning strategies to analyze over 12,000 features from various customers. And the idea here is that we wanted to predict the customer's credit worthiness so it actually go beyond the banking information so you can look at social media account use geo location, data, tons of information that can kind of tell you if this customer has a high credit worthiness or not.

Credit card transaction fraud costs billions of dollars to card issuers every year. A well-developed fraud detection system with a state-of-the-art fraud detection model is regarded as essential to reducing fraud losses. New advances in electronic commerce systems and communication technologies have made the credit card the potentially most popular method of payment for both regular and online purchases thus, there is significantly increased fraud associated with such transactions. The detection of fraudulent transactions has become a significant factor affecting the greater utilization of electronic payment.

Keywords: Bank marketing prediction, neural networks, machine learning, visualization, neuron mathematical model, single neuron model math, Ann training process, Credit fraud Detection

DOMAIN INTRODUCTION

- Data mining is the computing process of discovering patterns in large datasets involving methods at the intersection of machine learning, statistics and database systems.
- The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use.
- Data mining is the analysis step of the "knowledge discovery in databases" process, or KDD.
- Data mining is about finding new information in a lots of data.
- The information obtained from data mining is hopefully both new and useful.
- The migration of business to the Internet and the electronic monetary transactions that occur in the continuously growing cash-less economy have made the accurate detection of fraud a significant factor in securing such transactions.
- Credit card fraud occurs when a thief uses credit card information to complete purchase processes without permission from the credit card owner.
- In recent years, the volume of credit card transactions has been dramatically increasing due to the popularization of credit cards and the rapid development of e-services.
- The large-scale utilization of credit cards and the various transaction scenarios without rigid verification and supervision inevitably lead to billion-dollar losses caused by credit card fraud.

Bank loan prediction analysis:

In this project, we would like to build and train a simple model to predict the likelihood of a liability customer (depositor) buying personal loans based on customer features like age, experience, income, locations, family, education, existing mortgage, and credit card etc.

The idea is-

Most bank customers are depositors and the bank would like to start targeting those customers and issue personal loans to them.

The bank would like to retain those depositors and convert them to personal loan customers and therefore growing annual revenue.

The bank management team would like to analyze this campaign results and develop a model to target customers who have high probability of obtaining personal loans.

Features:

ID: customer ID

Age: customer's age in years

Experience: Number of years of professional experience

Income: Annual income of customer

ZIP code: Home address ZIP code

Family: Family size of the customer

CC Avg: Average spending on credit cards per month

Education: Education level (1: Undergrad; 2: Graduation; 3: Advance/Professional)

Mortgage: Value of house mortgage

Personal loan: Did this customer accept the personal loan offered in the last campaign or not?

Securities Account: Does the customer have a certificate of deposit (CD) account with the bank?

Online: Does the customer use internet banking facilities?

Credit Card: Does the customer uses a credit card issued by universal Bank?



1. Understand the problem statement and business case
2. Import libraries/datasets and perform preliminary data processing
3. Perform Exploratory data analysis
4. Perform data visualization
5. Prepare the data for deep learning model training (Normalization)
6. Understand the theory and intuition behind Deep Neural Networks
7. Build Deep Neural Networks

8. Compile and train deep learning model
9. Assess/evaluate trained model performance

Using artificial neural networks, artificial intelligence and machine learning banking industry is being transforming in many aspects, such as automatic fraud detection, bank customers, segmentation and robotic process automation.

In first step, we will import our data sets and our libraries. in module number one we have the problem statement and business case, Which is simply building a deep neural network that try to Absorb or consume all the customer features and try to predict whether the customer will accept the personal loan, offer or not. So the idea here is we're going to install Jupiter themes, like J T plot and the idea here is when we plot any figure We just when you run these two cells, when you set, they seem to be monarch. The idea here is that we just wanted to show the X label and why label in white so it can actually appear on this black background. Because it will be very difficult to see any data when we plot the figures, because it will be simply be black on black. So regarding the libraries and helper functions that we plan to leverage, we're going to import pandas. SPD and pandas is primarily used for data frame manipulation. We're going to import Matt plot lib, and that will be used for data visualization which is going to import NumPy for numerical analysis and raise essentially manipulations.

Later, import Seaborn as well, which is also used for advanced data visualization. It's actually very powerful library. The idea here is that we are going to import TensorFlow and TensorFlow is Google's framework to build, train and deploy AI and ML models. And we are going to use TensorFlow 2.0 in this project. And then we want to go here to the next cell. Now we will load our data set and use pandas to read the data.

Out[4]:

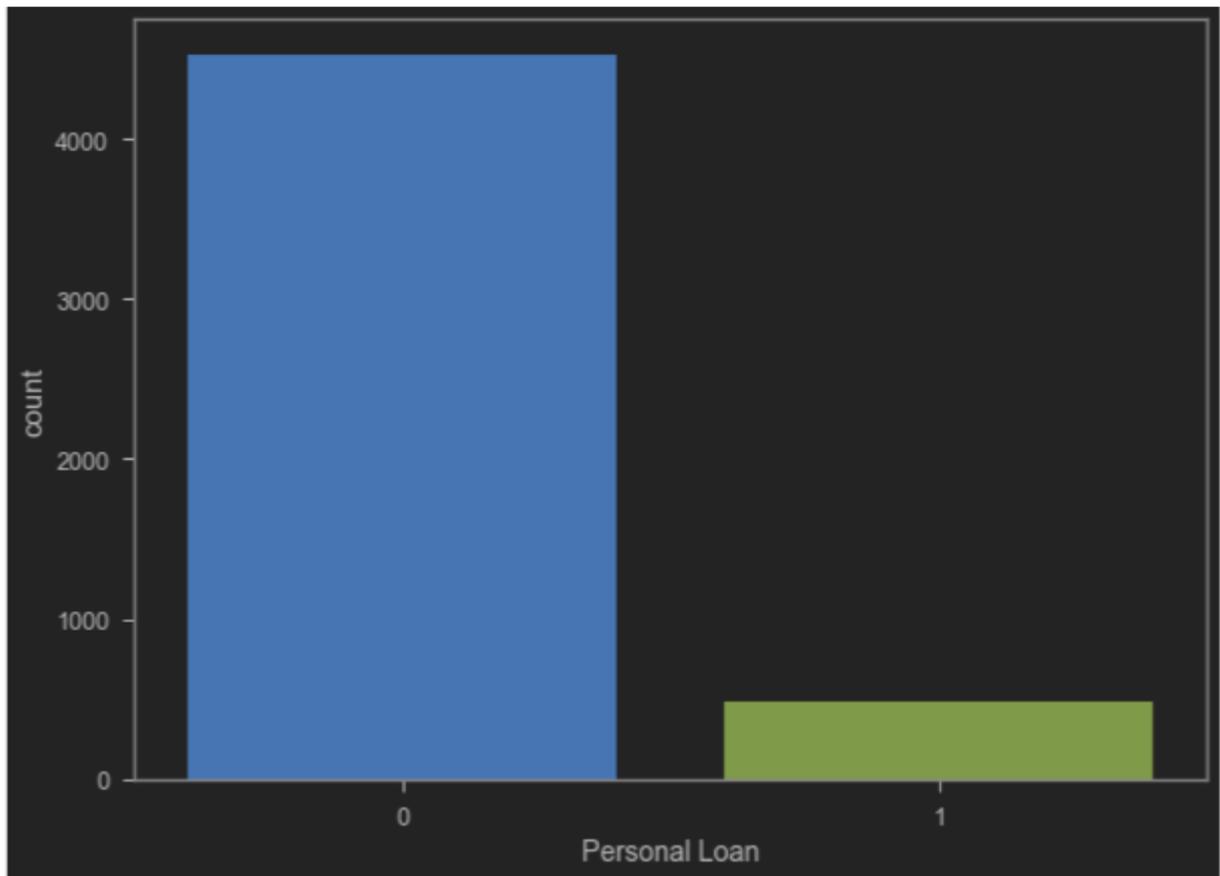
	ID	Age	Experience	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan	Securities Account	CD Account	Online	CreditCard
0	1	25	1	49	91107	4	1.6	1	0	0	1	0	0	0
1	2	45	19	34	90089	3	1.5	1	0	0	1	0	0	0
2	3	39	15	11	94720	1	1.0	1	0	0	0	0	0	0
3	4	35	9	100	94112	1	2.7	2	0	0	0	0	0	0
4	5	35	8	45	91330	4	1.0	2	0	0	0	0	0	1

Out[5]:

	ID	Age	Experience	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan	Securities Account	CD Account	Online	CreditCard
0	1	25	1	49	91107	4	1.6	1	0	0	1	0	0	0
1	2	45	19	34	90089	3	1.5	1	0	0	1	0	0	0
2	3	39	15	11	94720	1	1.0	1	0	0	0	0	0	0
3	4	35	9	100	94112	1	2.7	2	0	0	0	0	0	0
4	5	35	8	45	91330	4	1.0	2	0	0	0	0	0	1
5	6	37	13	29	92121	4	0.4	2	155	0	0	0	1	0
6	7	53	27	72	91711	2	1.5	2	0	0	0	0	1	0
7	8	50	24	22	93943	1	0.3	3	0	0	0	0	0	1
8	9	35	10	81	90089	3	0.6	2	104	0	0	0	1	0
9	10	34	9	180	93023	1	8.9	3	0	1	0	0	0	0

In second step, we need to perform exploratory data analysis. In module number one, we have been able to import our libraries and data set here. So we imported all our key libraries. We also imported our data. And we also went through a quick, many challenge to visualize the first couple of rows and the last couple of rows in our data frame. We used info method and the describe method to give information about data frame and also give summary of the statistical content in our data. Here we get summary of all columns. So here we have all the Id. All the age experience and so on and Also, we able to see all the different non null elements. So here it shows that I have 5000 non-null elements, and when we train any AI or ML model, we have to make sure that we do not have any null or missing elements in our data frame. In this data set we don't have any missing element. So here it says in 64 or integer 64 here for the CC average for the credit card average balance. It's actually float 64 the rest as well are in 64 which indicates that they are 64 bits of data. We can also see the memory usage for our data frame, which is 547 kilobytes, we can also tell me the average off each individual column or each individual features that we have in our data. We are also able to see the minimum age which is 23 the maximum age which is 67 years old. And also, the 25 percentile, 50 percentile and 75 percentile as well. And we are also

able to do that entire data frame or for all for all features. we used the method called the transpose. It's kind of another way of visualizing the exact same information, but maybe in a more friendly way. Next check how many elements do we have in data frame. We actually made that check. And if we apply is null simply the result will be null, It's kind of you're asking a question if there another element in my data frame or not. Also we wanted to know what the percentage of customers who have credit cards is. We also wanted to know what is the percentage of customers who accepted the personal loan as well. Would say, pass that meaning challenge.

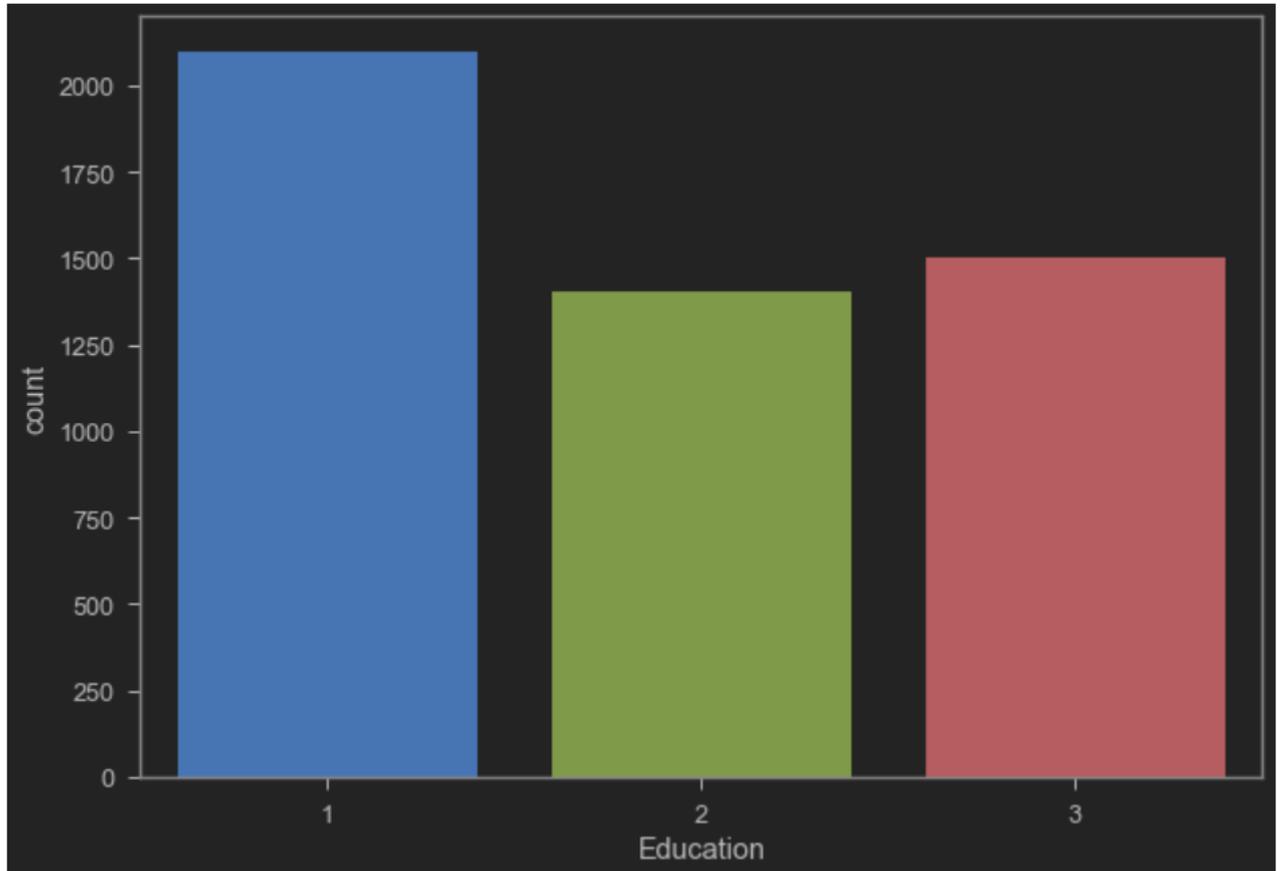


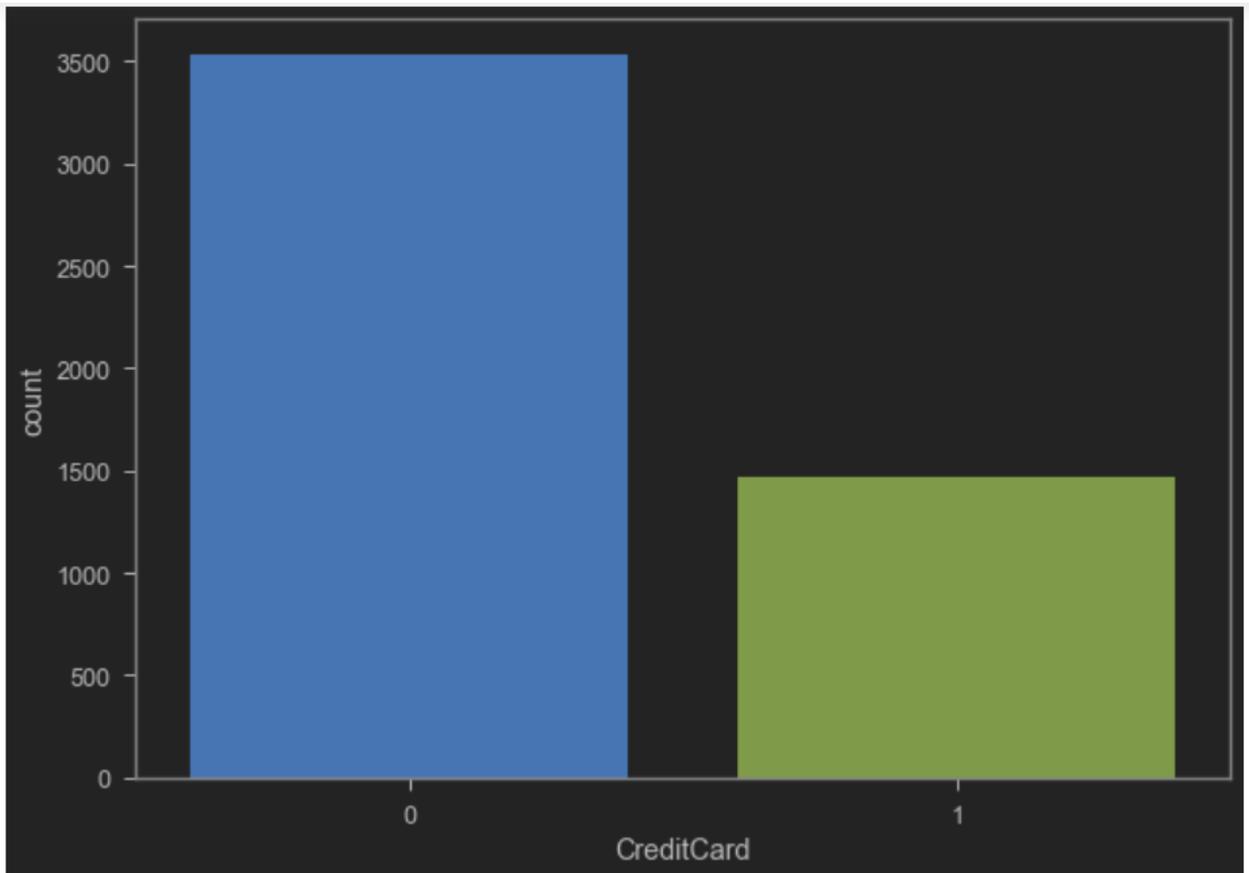
So there after we get the credit card column. It's just essentially bunch of ones and zeros. It just kind of indicate that if the customer has a credit card with the Universal Bank, it will has a flag here off one. And if not, that means it will be zero. So what we could do is essentially, is I can simply count how many elements, how many rows or how many customers belong to the one class and how many customers belong to the zero class. And that will give us idea of the percentage of customers who have credit cards. So essentially, we are just going to divide my data frame into two data frames, one of them that has the credit cards and the other one who do not have credit cards.

In third step, we will perform data visualization. In module number two, we have been able to perform exploratory data analysis. We visualize the data frame, and we also have been able to see the percentage of customers who have credit cards with the bank and also the percentage of customers who accepted the personal loan, which was around 9%. In this module, we are going to specify my fig size to be equals two. And here I'm going to say maybe 10/7. And what I could do next is I can use seaborn count plot to plot out the number of people that accepted the personal loan and who denied the person alone. So here our column is my personal space loan column. Here I have approximately the 9% that we had before, which is essentially here. These are the customers who accepted the loan, which is again here. I would say maybe around maybe 700 customers Also. And here are the number of customers who deny the personal law. we would say the education level two and three are very close and a lot of people belong to the education group. we would say distribution approximately between the age of maybe 30 years old and 60 years old, Pretty consistent. And then it tapered off here as we go to 67 years old. And it tapers as well down here as we go to a younger age, which, let's say, 23 years old. Next, we need to visualize the credit card availability for customers so I can say as well. We can specify my fig size to be equals two and the fixed size. Next, we wanted to print out the distribution of the income. used what we call it this plot or distribution plot. So I can say this plot, I can specify. Kind of like slopes down until we go beyond the \$100,000 range, which makes sense, because again, the average salary or the average income in the us is around \$50,000 approximately. So next, we would simply divide customers into two groups, One of them that that has personal loans. So whenever we find personal loan equals to one, that will be the first group and the second group who do not have personal loans, and we call it no personal loans. And after visualizing the personal loans data frame here again, that's our first beta frame. Have 480 approximately customers who have personal loans. And what we could do next is the following. We can go ahead and apply as well the describe method on personal loans data frame. now would be able to kind of get an idea off the people who have personal loans with the bank. So you will find that the mean income off customers who have personal loans with the bank is actually generally quite high, said I'm with \$144,000 and the average credit card spending kind of is around \$4000 approximately. So if we do the same analysis on the other data frame,

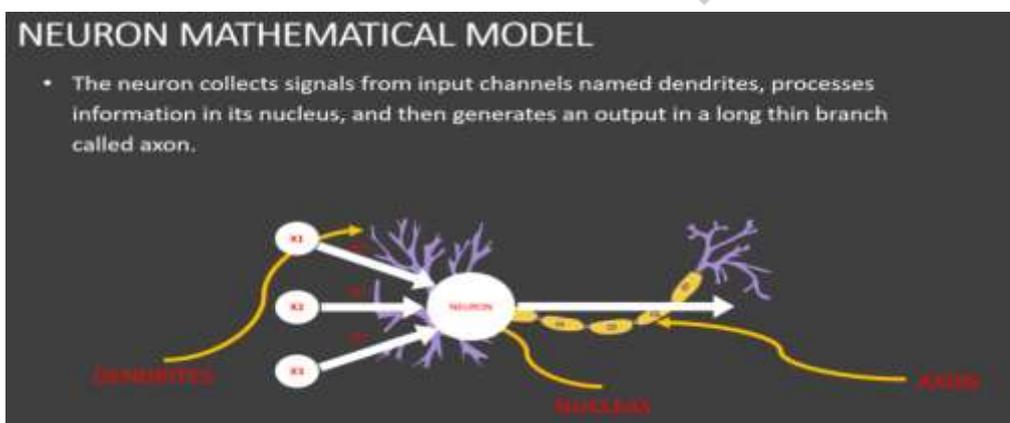
which has no personal loans. So again, here I have my own personal loans. And if I can go here and do the same thing here as well so I can go here and I can do my no personal loans. So we should be able to see the average income here is around \$66,000 and the average credit card debt is around \$1700. That actually tells you a lot of information already kind of give you a summary off. Who are the people who actually accepted the person alone? So it actually seems which is a very important piece of information. The people who have higher incomes tend to use or accept the personal loan in this example. So next what I wanted to do here is I wanted to plot out the distribution plot for both classes separately.

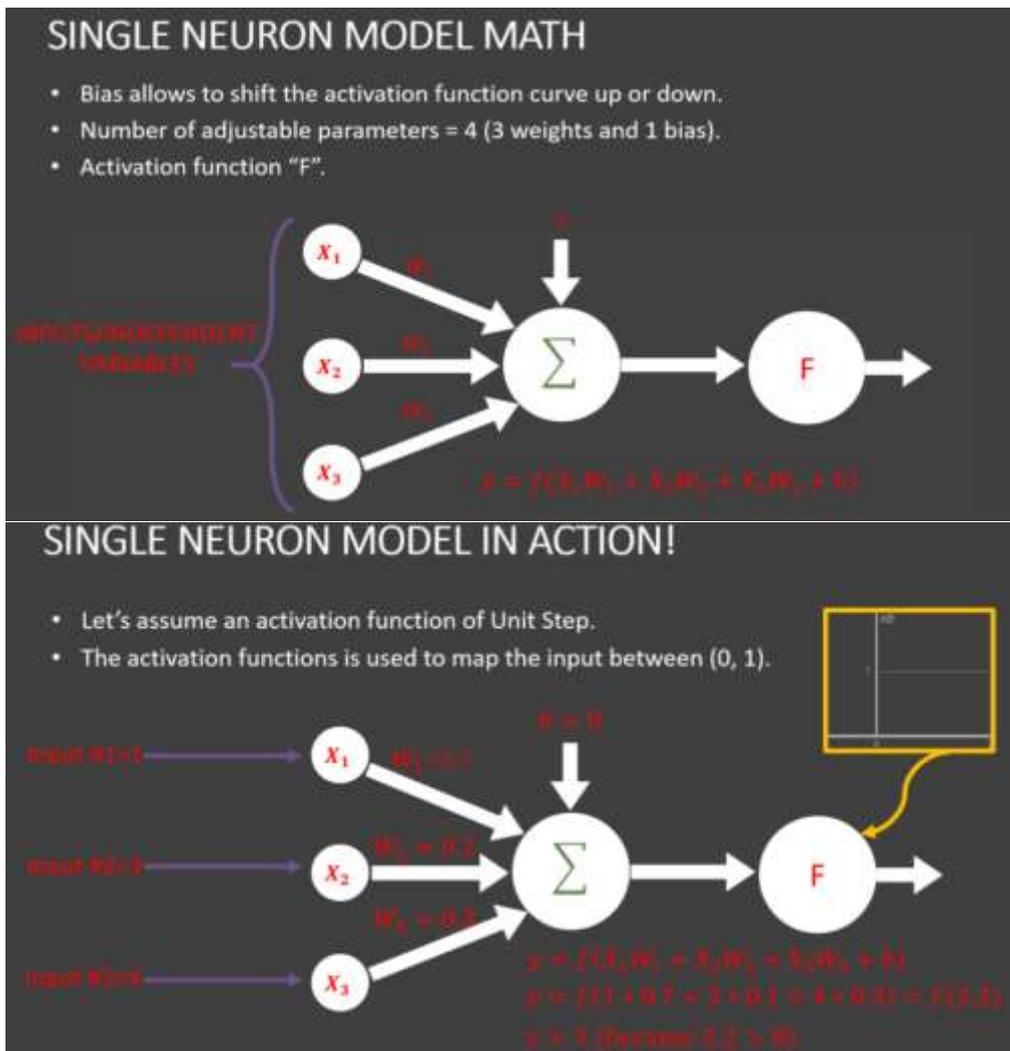
```
Out[17]: <AxesSubplot:xlabel='Education', ylabel='count'>
```





Step 4, In this model, we will prepare the data before training our deep learning model. In model number 3, it was quite intense and we went through a lot of data visualization. So we this plot or distribution plot. We did the count plot pair plot, and we also plotted out the correlation matrix. Now, we prepared data before trained the model. Simply, we just wanted to take the entire data frame and divided into inputs and outputs. Except for the personal loan column, which indicates whether the customer accepted the loan or not. And that essentially, will be my target variable. So all the inputs we are going to call them X upper case and all the outputs, which is only one column or only one output. So again, all of them are inputs except for the personal loan, and that will be our output. So, what we did is, say X equals two. Bank underscored the F and going to say dot drop and specified which column that we wanted to drop. And I put that between quotation marks, we obtained all the features except target variable. Next one is the output. So, which is going to be the output or the target variable will be bank underscored the F and going to only get my personal loan column essentially, So, here, simply we have just bunch of ones and zeros indicating whether the customer accepted the personal loan or not. Next, what we wanted to do is I wanted to convert the one single column here in the output into essentially two outputs. We called them basically 10 or zero and one indicating. The two classes that we have that will come really handy when we start to train our neural network. So now we saved from TensorFlow Kerry's utilities going to import too categorical. And then categorical off why? And that will give the new y. I have 10 or 01 indicating with either talking to class, the first class or the second class. Next, we installed psychic learn and psychic learn is a super powerful library that could be used for machine learning in general. And we're going to use psychic learn to print out various KP eyes or key performance indicators.



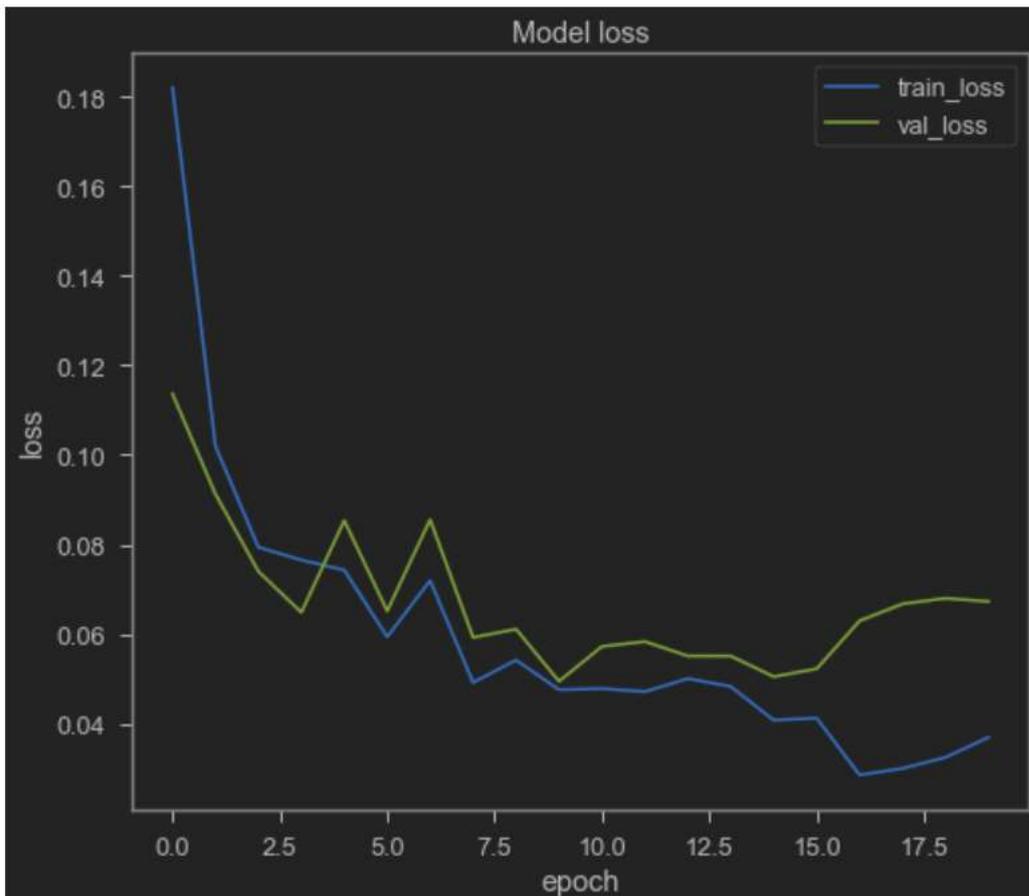


In 5th step, we will build a simple, multi layer artificial neural network model to essentially perform our classifications task. So we are going to build the model. We're going to train it, and then we are going to test our model as well. And that will be in the next couple of module eight and module nine. How do we built artificial neural networks and how do we train them as well, using the ingredient descent algorithm. So here we built our network using caress API. So simply caress is an API that sits on top of TensorFlow to Plano, so it will make our life extremely easy, and you will find that we will build a super powerful artificial neural network using just maybe, like, eight lines of code, like really simple and very, very easy to build. As we imported caress beforehand in the first module. And then here we can say we built our model in a sequential fashion. So essentially, we built our model layer by layer. So in first layer, which is the input layer, followed by the next hidden layer by the next hidden layer until we reach the output. So here we had our artificial neural network model object and then think of it as kind of your building a network kind of in the same fashion like a Lego Mindstorms. So we will use rectify linear unit and pretty much every hidden layer. However, in the output used or chose a different activation function. So simply pop out is a technique known as a regularization technique. It's simply a way to try to ensure that the artificial neural network is not over fitting the training data. So what we do is that randomly would drop couple off neurons when we train the network, and we found that this technique actually generates amazing results. It just improve the generalization ability of the network and it make it extremely robust. The output layer has two neurons. as, we did too categorical to the why which is the output, so essentially going to have two neurons, the first neuron If it's the first neuron fires, that means the client has accepted the personal loan. If the second neuron fired, that means the client has declined the personal loan. The idea here is that because we want the output to be saturated and that's why we're using a soft max activation function in the output. And finally, we can say artificial neural network model dot summary. And that was simply going to show the summary for the entire network.

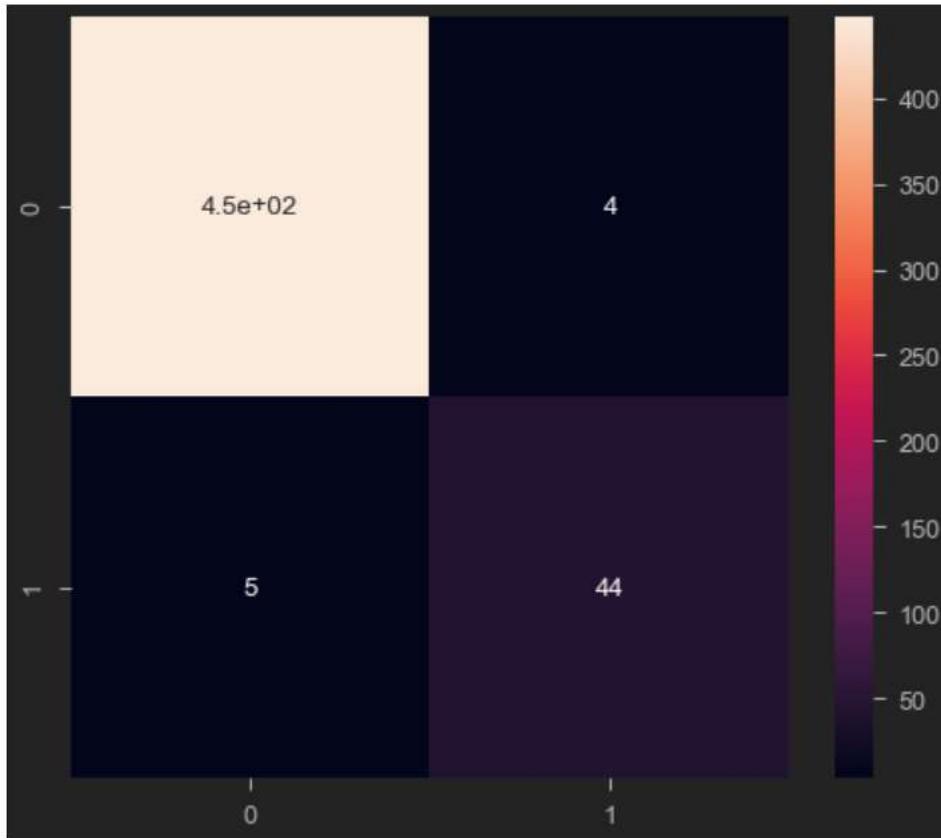
Layer (type)	Output Shape	Param #
dense (Dense)	(None, 250)	3500
dropout (Dropout)	(None, 250)	0
dense_1 (Dense)	(None, 500)	125500
dropout_1 (Dropout)	(None, 500)	0
dense_2 (Dense)	(None, 500)	250500
dropout_2 (Dropout)	(None, 500)	0
dense_3 (Dense)	(None, 500)	250500
dropout_3 (Dropout)	(None, 500)	0
dense_4 (Dense)	(None, 500)	250500
dropout_4 (Dropout)	(None, 500)	0
dense_5 (Dense)	(None, 2)	1002
Total params: 881,502		
Trainable params: 881,502		
Non-trainable params: 0		

In sixth step are going to compile and train our deep neural network model. In task number seven, we have been able to build our model.

Now we have 750,000 train every parameters, and what we did here is need to compile the model. So simply, we need to specify what type of optimizers that we used. What are the metrics that we used to optimize the network And pass it along. our training data, which is ex train and white train and also specify how many epochs do we need to train the model for? So that will compile the model, need to specify what is my loss. And my loss here will be categorical across entropy. So categorical cross entropy, next we specified what is the type of the optimizers. And if we specify the optimizer here, we are going to call it Adam Optimizer. And there are so many optimizers. But actually, Adam Optimizer works really well. So what we did next here is that the model is being trained as we're running the model. we just want to make sure that the model is not over fitting the training data, which means that the model learns all, the ins and outs all the details off the training data. However, it might fail to generalize. so if we take the exact same model and deployed, for example in the field on a new data set, you will find that the model will perform very, very poorly, even though during training the model was great. However, when we use it in testing, the model performed very poorly and we actually hate these models as you know, like engineers or machine learning. Engineers will try to ensure that our model is able to generalize and not memorize, and that's a very important point. So the validation split here will be set to point to, and I can actually go ahead and specify the verbose B equals to one. And that's essentially the amount the number of content here, like, what is what is going to be printed out here. The loss started at a 9.96 That's that the accuracy is around 93 on, then jumped to 96 which is great on the 97. Think of the loss of the error on the training data, and we also wanted show the validation loss, which is for the validation data set.



In 7th step we will assess the performance of the train model. In module eight, we have been able to train our model and we went through 28 books. So now we have the train model, which means now we applied the optimizer, and was able to obtain the optimized values for the weights on basis of network. And now can take the artificial neural network model and apply the predict method on my model. Pass it along. The testing data, which is a very important point here that we wanted to indicate, is that the testing data has never been seen by the model during training. And that's a crucial part because, as mentioned, we want to make sure that the model is able to generalize and not memorize. We want to make sure that the model will still perform really well. Even though the model has never seen this data ever before, which is the testing data. So here just going to apply the predict method on model, pass it along the testing data and regenerate predictions. And next here, we are going to obtain what we call it the maximum index off the maximum value. So , the output might be one or zero right, or zero or one indicating whether the customer accepted the personal loan or not. So when we actually run this cell here, shift enter. Here we go and can simply apply the evaluate message here, pass it along the testing data, pass it along the Y testing data to generate the results. So keep your shift enter. that will tell you simply that we have 97% accuracy on the testing data, which is pretty great. And what we wanted to do next is I wanted to generate the predictions that is coming out from my model. Right. And these are the original. These are the actual values that we got in here, we wanted to compare it with the original values, which is the actual y values, which is why testing data that I had in my testing data originally. So here, just going to obtain the original values. And we can simply compare the original values versus my model predictions. So what we could do here is the following.



Matrix equals two metrics, And pass it along original and pass it along the predicts And next, say ss dot heat map and pass it along here. My confusion matrix, confusion matrix again tap should or a complete and specify the annotations to be equal to true as well and actually make sure that the tea here is upper case. So that would be the true. And we could go ahead if you press shift Enter. Here we got, this is simply kind of, a visual representation of classifier performance. So basically, here we have the model predictions and ground truth, and we are simply comparing what the model is predicting versus what's actually happening in reality. So, these are the total number of samples that have been correctly classified. These are the total number of samples that the model has correctly classified, which is what we call it the true positives and the true negatives. And these samples which is seven plus 27. These are the samples that the model has messed up. Basically, the model said that the customer accepted the personal loan. However, in reality, the customer denied the person alone. And this is essentially the number of samples that have been misclassified. So if we go ahead and print out the classifications report, we can say from S K learn .metrics going to import by classification on the score reports. And we can go ahead and say Prince and can print out classifications, reports education on the score report and pass along my Y original and my predict here. so if you press shift enters, it takes some time for it to print. But overall, will be able to come up with kind of a summary of all the information of classifier so should be able to see the accuracy. I should be able to see the performance of the model, and here we go. So as we can see next, the overall accuracy of the model is around 96% which is pretty great, and will also find that the precision is around 98 95 for both classes. The recall is that at 1.99 point 83 which is pretty great and F one score, which is the harmonic mean between the precision and recall, that's the F one score. And again, the performance of the model is actually pretty great and our 99 88% and that's it. That's all we have for this project.

```
In [60]: #print out the classification report
from sklearn.metrics import classification_report
print(classification_report(y_original, predict))
```

	precision	recall	f1-score	support
0	0.99	0.99	0.99	451
1	0.92	0.90	0.91	49
accuracy			0.98	500
macro avg	0.95	0.94	0.95	500
weighted avg	0.98	0.98	0.98	500

Credit Card Fraud Detection using Artificial neural network Algorithm

DESCRIPTION

DATA SELECTION AND LOADING

- Data selection is the process of determining the appropriate data type and source, as well as suitable instruments to collect data.
- Data selection precedes the actual practice of data collection and it is the process where data relevant to the analysis is decided and retrieved from the data collection.
- Data loading refers to the "load" component.
- After data is retrieved and combined from multiple sources, cleaned and formatted, it is then loaded into a storage system, such as a cloud data warehouse.
- In this project, the credit card dataset is used for detecting the fraud detection.
- The dataset which contains the information about the time, amount, class, v1 and v2, etc.

Out[12]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.1285
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.1671
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.3276
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.6475
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267	-0.2066

5 rows × 31 columns

Data Visualization



DATA PREPROCESSING

- The data can have many irrelevant and missing parts. To handle this part, data cleaning is done. It involves handling of missing data, noisy data etc.
- Missing Data:
 - This situation arises when some data is missing in the data. It can be handled in various ways.
 - ✓ Ignore the tuples:
 - This approach is suitable only when the dataset we have is quite large and multiple values are missing within a tuple.
 - ✓ Fill the Missing values:
 - There are various ways to do this task. You can choose to fill the missing values manually, by attribute mean or the most probable value.

SPLITTING DATASET INTO TRAIN AND TEST DATA

- Data splitting is the act of partitioning available data into two portions, usually for cross-validator purposes.

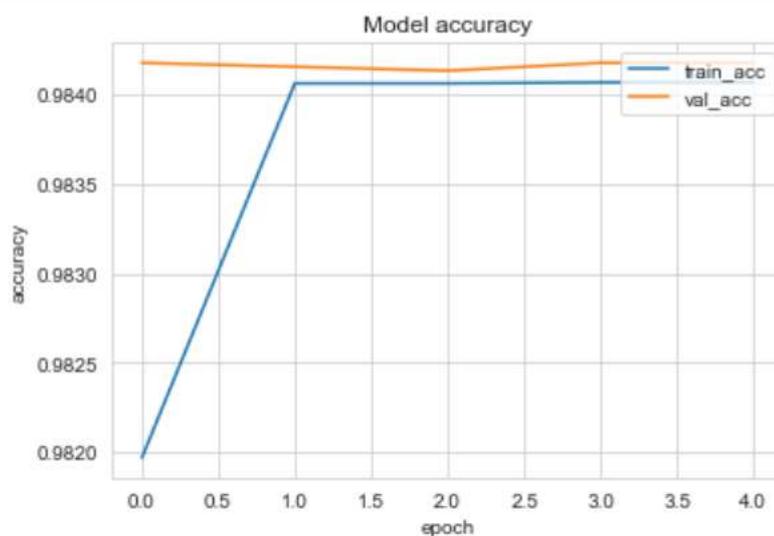
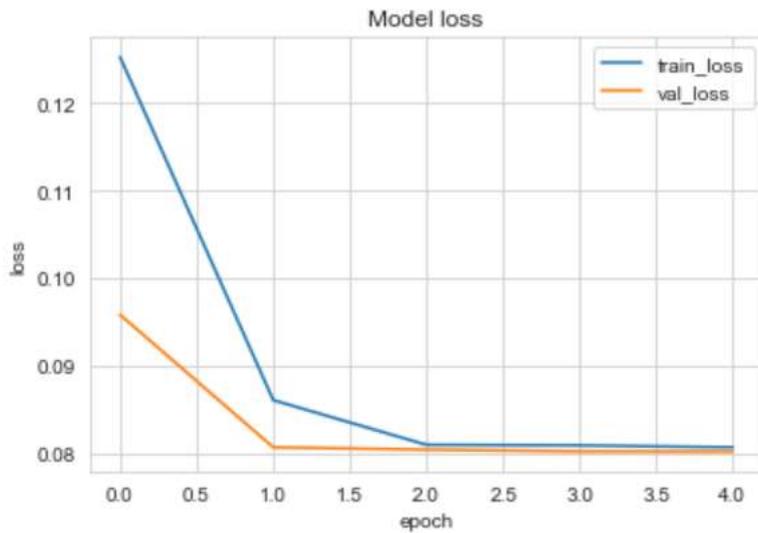
- One Portion of the data is used to develop a predictive model and the other to evaluate the model's performance.
- Separating data into training and testing sets is an important part of evaluating data mining models.
- Typically, when you separate a data set into a training set and testing set, most of the data is used for training, and a smaller portion of the data is used for testing.
- To train any machine learning model irrespective what type of dataset is being used you have to split the dataset into training data and testing data.

CLASSIFICATION

- The Supervised classification algorithm ANN (Artificial Neural Network)-a deep learning algorithm is used in Data Mining.
- An Artificial Neural Network is an information processing technique. It works like the way human brain processes information. ANN includes a large number of connected processing units that work together to process information. They also generate meaningful results from it.

PREDICTION

- Predictive analytics algorithms try to achieve the lowest error possible by either using “boosting” or “bagging”.
- **Accuracy** – Accuracy of classifier refers to the ability of classifier. It predict the class label correctly and the accuracy of the predictor refers to how well a given predictor can guess the value of predicted attribute for a new data.
- **Robustness** – It refers to the ability of classifier or predictor to make correct predictions from given noisy data.
- **Scalability** – Scalability refers to the ability to construct the classifier or predictor efficiently; given large amount of data.



RESULT GENERATION

- The Final Result will get generated based on the overall classification and prediction. The performance of this proposed approach is evaluated using some measures like,

- ✓ $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$
- ✓ $Precision = \frac{TP}{TP+FP}$

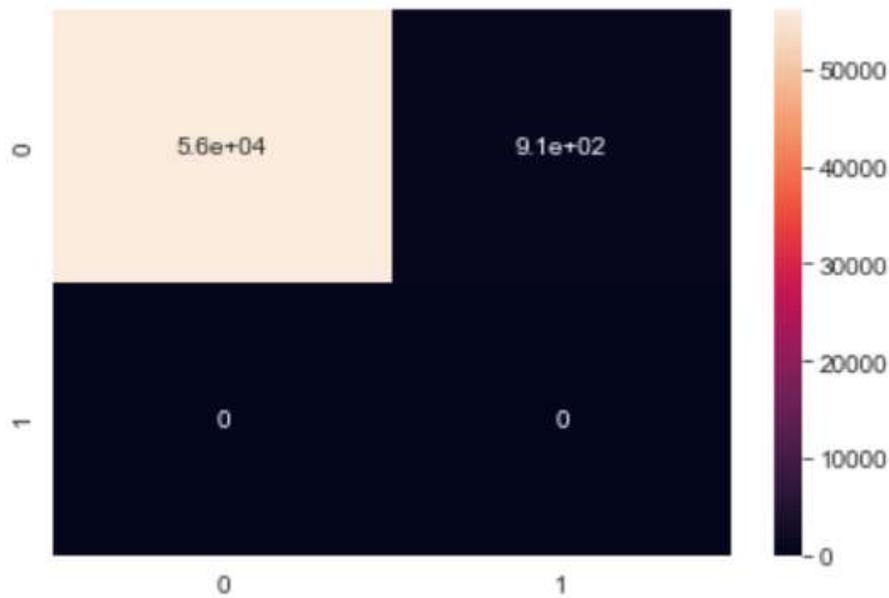
$$\checkmark \text{ Recall} = \frac{TP}{TP+FN}$$

$$\checkmark \text{ F1-Score} = \frac{2TP}{2TP+FP+FN}$$

```
In [36]: #print out the classification report
from sklearn.metrics import classification_report
print(classification_report(y_pred, y_test))
```

	precision	recall	f1-score	support
0	1.00	0.98	0.99	56962
1	0.00	0.00	0.00	0
accuracy			0.98	56962
macro avg	0.50	0.49	0.50	56962
weighted avg	1.00	0.98	0.99	56962

Out[35]: <AxesSubplot:>



Conclusion

- So in this project, we started with an understanding of the problem statement and business case. We have been able to build a classifier model based on artificial neural network that will be able to take in customer features such as their age, their experience, their income level and essentially whether the customer will accept the personal loan or not. And then, we have been able to import our libraries and data sets. And then, we performed exploratory data analysis. Then by performing data visualization, we have been able to prepare the data before model training. We also understood the theory and intuition behind artificial neural networks. And how do we train them using the ingredient descent algorithm and then, we have been able to build our model to compile and train our model. And then, we have been able to assess the performance of the train model.
- The detection of credit card fraud is significant to the improved utilization of credits cards.
- With large and continuing financial losses being experienced by financial firms and given the increasing difficulty of detecting credit card fraud, it is important to develop more effective approaches for detecting fraudulent credit card transactions.
- The results show that the deep learning approaches dramatically outperform all other baselines.
- The experimental results indicate that the proposed approach outperformed the deep learning algorithms and achieved the highest performance in terms of Accuracy, Precision and F1-score.

References:

[1] <http://sersc.org/journals/index.php/IJAST/article/view/15497>
 [2] <http://sersc.org/journals/index.php/IJAST/article/view/15497>

- [3] https://www.researchgate.net/publication/263054095_Bank_Direct_Marketing_Analysis_of_Data_Mining_Techniques
- [4] <https://ieeexplore.ieee.org/document/7219777>
- [5] https://repository.asu.edu/attachments/170603/content/Ejaz_asu_0010N_15774.pdf
- [6] <https://arxiv.org/pdf/1503.04344>
- [7] N. Sethi and A. Gera, “A revived survey of various credit card fraud detection techniques,” *Int. J. Comput. Sci. Mobile Comput.*, vol. 3, no. 4, pp. 780–791, Apr. 2014.
- [8] O. Adewumi and A. A. Akinyelu, “A survey of machine-learning and nature-inspired based credit card fraud detection techniques,” *Int. J. Syst. Assurance Eng. Manage.*, vol. 8, no. S2, pp. 937–953, Nov. 2017.
- [9] J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare, “Credit card fraud detection using machine learning techniques: A comparative analysis,” in *Proc. ICCNI, Lagos, Nigeria*, Oct. 2017, pp. 1–9.

