



# Prediction for Loan Approval using Deep Learning Algorithm (INT 247)

---

Name	ROLL NO.	Registration no.
Vishavjeet Singh	36	12012866

**Submitted to:** Ms. Niharika Thakur

## **Abstract –**

The two most occurring issues in the banking sector are:

- 1) How risky is the borrower?
- 2) Should we lend to the borrower given the risk?

By predicting the loan defaulters, the bank can reduce its Non-performing Assets. This makes the study of this phenomenon very important. But as the right predictions are very important for the maximization of profits, it is essential to study the nature of the different methods and their comparison.

Approach in predictive analytics is used to study the problem of predicting loan defaulters

- (i) Collection of Data,
- (ii) Data Cleaning and
- (iii) Performance Evaluation.

This model extracts and introduces the essential features of a borrower that influence the customer's loan status. Finally, it produces the planned performance (loan status). These reports make a bank manager's job simpler and quicker.

## **INTRODUCTION –**

For companies like Lending Club, predicting loan default with high accuracy is very important. Using the historical Lending Club data from 2007 to 2015, build a deep learning model to predict the chance of default for future loans.

The aim of this project is to provide quick, immediate, and easy way to choose the deserving applicants.

Customer first apply for loan after that company or bank validates the customer eligibility for loan.

Company or bank wants to automate the loan eligibility process (real time) based on customer details provided while filling application form.

The machine learning model is trained on that record to get accurate results. Our main objective of this project is to predict the safety of loan. To predict loan safety, the Deep Learning Implementation is used.

First the data is cleaned to avoid the missing values in the data set.

## **Libraries for Data Analysis –**

**Pandas:** It is a powerful open-source tool used for data analysis and data manipulation operations such as data cleaning, merging, selecting wrangling as well.

**Sklearn:** This python library is helpful for building machine learning and statistical models such as clustering, classification, regression etc

**NumPy:** It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices, and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.

**Tensorflow:** A Python-friendly open source library for numerical computation that makes machine learning and developing neural networks faster and easier. TensorFlow allows developers to create dataflow graphs—structures that describe how data moves through a graph, or a series of processing nodes.

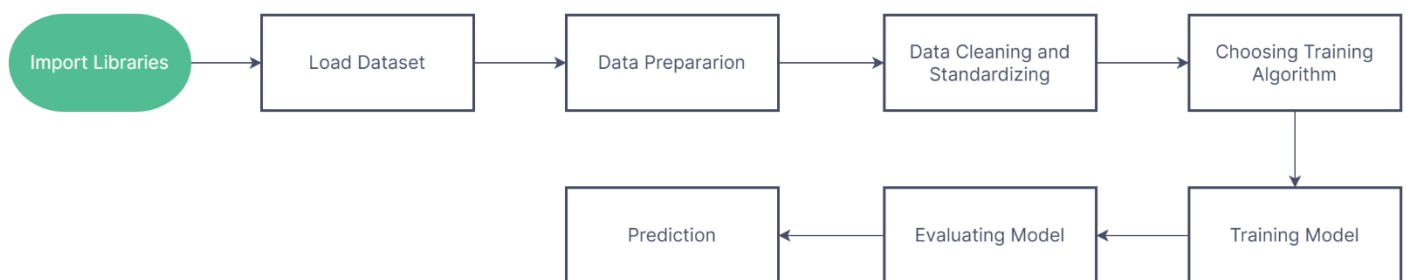
**Keras:** Keras is a high-level, deep learning API developed by Google for implementing neural networks.

## PROPOSED MODEL –

This system predict whether the loan is approve or reject. This System refers the following things or ways.

1. Data Collection
2. Data Pre-processing (Data Cleaning)
3. Model Selection
4. Model Evaluation
5. Classification
6. Result (output)

## FLOW CHART:



## IMPLEMENTATION DETAILS (MODULES):

1. **Loan Dataset:** Using the loan Dataset the system will automatically predict which costumer's loan it should approve and which to reject. System will accept loan application form as an input. Justified format of application form should be given as an input to get processed.
2. **Determine the training and testing data:** The system separate a dataset into a training set and testing set, most of the data use for training, and a smaller portion of data is use for testing. after a system has been processed by using the training set, it makes the prediction against the test set.
3. **Data cleaning and processing:** Data cleaning the system detect and correct corrupt or inaccurate records from database and refers to identifying incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing , modifying or detecting the dirty or coarse data. Data processing the system convert data from a given form to a much more usable and desired form i.e., make it more meaningful and informative.

## MODEL USED –

1. **Sequential:** The core idea of Sequential API is simply arranging the Keras layers in a sequential order and so, it is called Sequential API. Most of the ANN also has layers in sequential order and the data flows from one layer to another layer in the given order until the data finally reaches the output layer.

```
In [34]: model = Sequential()
          model.add(
              Dense(94, activation='relu')
          )
          model.add(
              Dense(30, activation='relu')
          )
          model.add(
              Dense(15, activation='relu')
          )
          model.add(
              Dense(1, activation='sigmoid')
          )
          model.compile(
              optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy']
          )
```

## PROPOSED ALGORITHM –

The following shows the pseudo code for the proposed loan prediction method

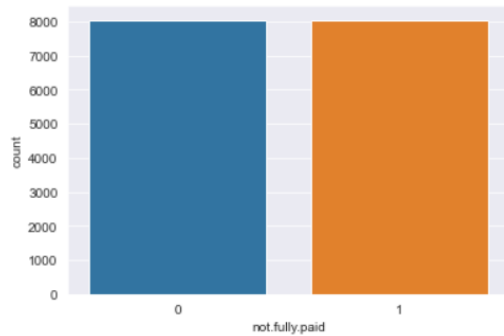
- Load the data
- Determine the training and testing data
- Data cleaning and pre-processing:
  - (i) Fill the missing values with mean values regarding numerical values.
  - (ii) Fill the missing values with mode values regarding categorical variables.
  - (iii) Outline treatment.
- Apply the modelling for prediction
  - (i) Removing the load identifier
  - (ii) Create the target variable, here target variable is the **not.fully.paid**.
  - (iii) Create a dummy variable for categorical variable and spilt the training and testing data for validation.
  - (iv) Apply the model.
- Determine the accuracy followed by Confusion Matrix.

# OUTPUT –

## Data Visualization

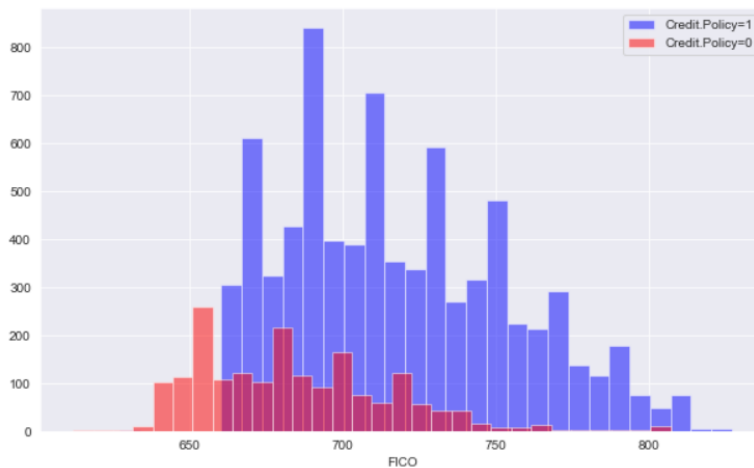
```
In [15]: #df_test_over['not.fully.paid'].value_counts().plot(kind='bar', title='Count (not.fully.paid)')  
sns.set_style('darkgrid')  
sns.countplot(x='not.fully.paid', data=df_test_over)
```

Out[15]: <AxesSubplot:xlabel='not.fully.paid', ylabel='count'>



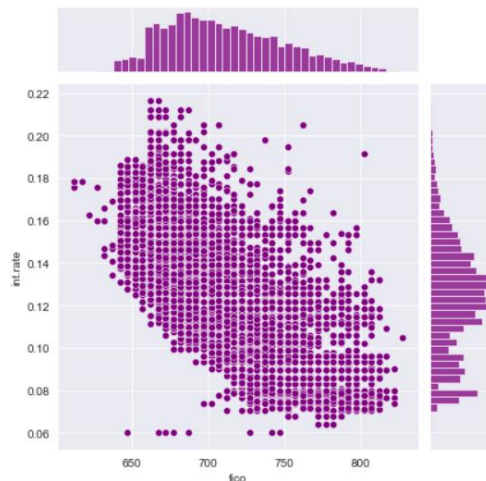
```
In [18]: plt.figure(figsize=(10,6))  
df[df['credit.policy']==1]['fico'].hist(alpha=0.5,color='blue',bins=30,label='Credit.Policy=1')  
df[df['credit.policy']==0]['fico'].hist(alpha=0.5,color='red',bins=30,label='Credit.Policy=0')  
plt.legend()  
plt.xlabel('FICO')
```

Out[18]: Text(0.5, 0, 'FICO')



```
In [21]: sns.jointplot(x='fico',y='int.rate',data=df,color='purple')
```

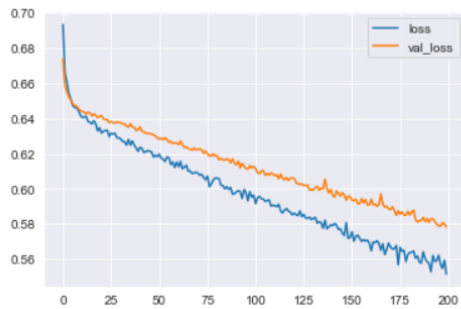
Out[21]: <seaborn.axisgrid.JointGrid at 0x1aa813aefd0>



In [39]: `#The graph shows that, by adding in Dropout layers, we have reduced the overfitting issue compared with the old model`

```
pd.DataFrame(model_new.history.history)[['loss', 'val_loss']].plot()
```

Out[39]: <AxesSubplot:>



## Accuracy Score – Training and Testing

In [40]: `predictions_new = (model_new.predict(X_test) >= 0.2).astype('int')  
predict_classes=np.argmax(predictions_new,axis=1)`

```
print(  
    confusion_matrix(y_test,predict_classes),  
    '\n',  
    classification_report(y_test,predict_classes)  
)
```

```
[[2437  0]  
 [2390  0]]  
      precision    recall  f1-score   support  
  
    0       0.50      1.00      0.67      2437  
    1       0.00      0.00      0.00      2390  
  
 accuracy          0.50          4827  
 macro avg         0.25      0.50      0.34          4827  
 weighted avg         0.25      0.50      0.34          4827
```

## CONCLUSION –

It works correctly and fulfils all requirements of bankers. This system properly and accurately calculate the result. It predicts the loan is approve or reject to loan applicant or customer very accurately.

From the research question, We understood that we need to perform various test and add visualizations to understand the pattern find the interest rate. Also, lot of financial terms while dealing with loans.

Reference to code: <https://github.com/VishavjeetS/Loan-Predicrtion-Deep-Learning>