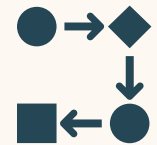
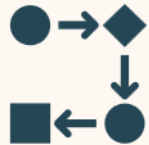


LOAN APPROVAL PREDICTION SYSTEM USING MACHINE LEARNING

Teacher Name: Kudratdeep Aulakh

Group Members:

Aparna Singh (102103414)
Akshay Khanna (102103415)



INTRODUCTION

1. Loans are the core business of banks. The main profit comes directly from the loan's interest.
2. The loan companies grant a loan after an intensive process of verification and validation. However, they still don't have an assurance if the applicant is able to repay the loan with no difficulties.
3. Thus, we'll build a predictive model to predict if an applicant is able to repay the lending company or not. we will prepare the data using jupyter notebook and use various models to predict the target variable.



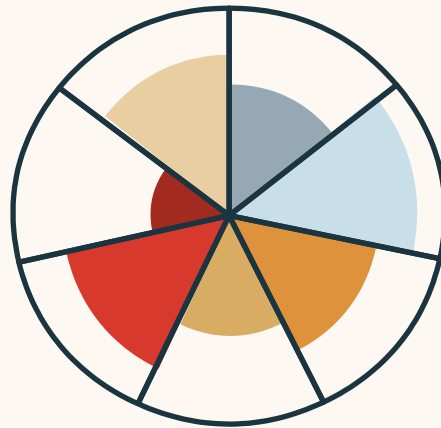
PROBLEM STATEMENT AND SOLUTION

PROBLEM STATEMENT

1. Intensive time consumption process of verification and validation.
2. Human errors can be introduced during the validation process.
3. No cross referencing previous loan records
4. Lot of human resource required.

PROBLEM SOLUTION

1. Our machine learning model calculates all the parameters given and predicts if the applicant is eligible for loan or not in very less time.
2. Time required for verification, and validation reduces significantly.



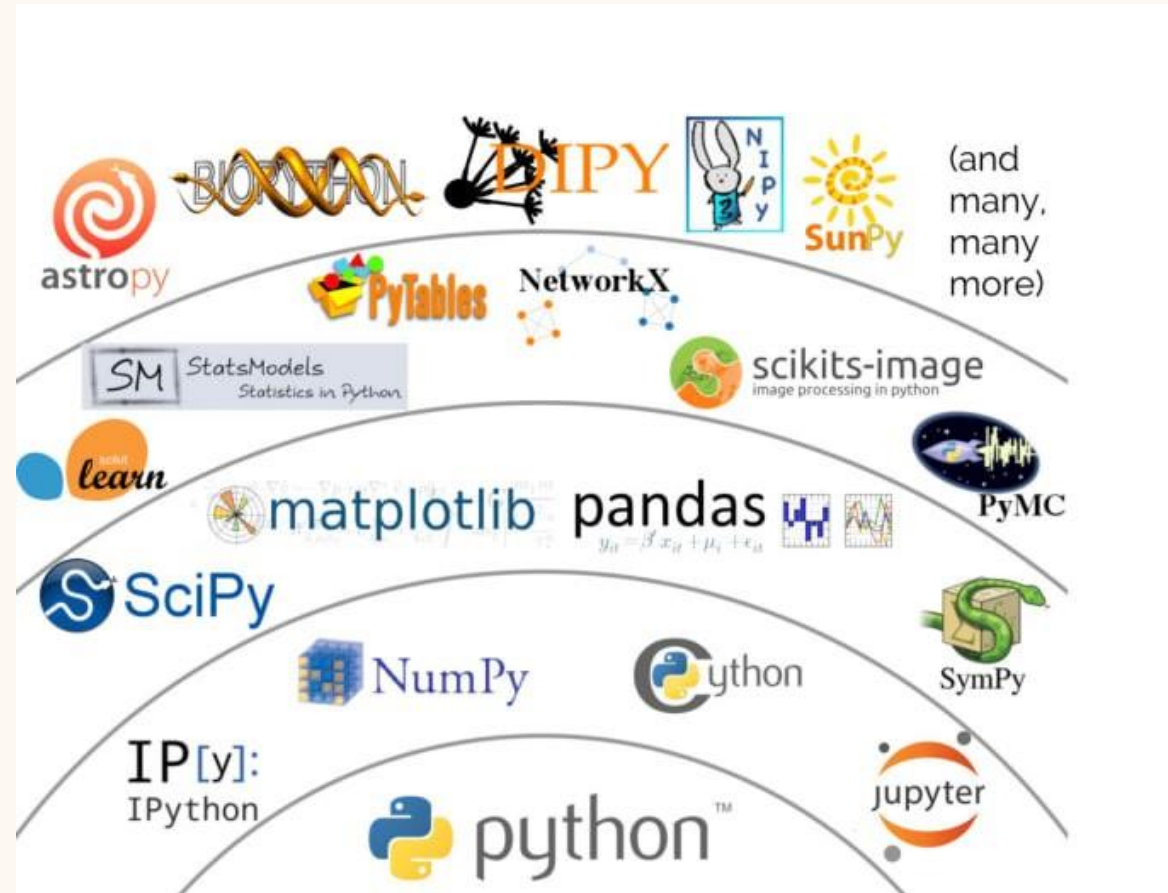
HARDWARE AND SOFTWARE USED

HARDWARE USED :

- Windows Computer

SOFTWARE USED :

- Language – Python
- Libraries –
 1. Pandas
 2. Matplotlib
 3. Scikit Learn
 4. NumPy



DATA

For this problem, we have three CSV files: train, test, and sample submission.

1. **Train file** will be used for training the model, i.e. our model will learn from this file. It contains all the independent variables and the target variable.
2. **Test file** contains all the independent variables, but not the target variable. We will apply the model to predict the target variable for the test data.
3. **Sample submission file** contains the format in which we have to submit our predictions.



MODELS

```
# decision tree classifier
from sklearn.tree import DecisionTreeClassifier
model2 = DecisionTreeClassifier()
model2.fit(x_train, y_train)
print("Accuracy is", model2.score(x_test, y_test)*100)
```

Accuracy is 68.181818181817

```
# logistic regression
from sklearn.linear_model import LogisticRegression
model3 = LogisticRegression()
model3.fit(x_train, y_train)
print("Accuracy is", model3.score(x_test, y_test)*100)
```

Accuracy is 77.272727272727

DECISION TREE CLASSIFIER
Accuracy: 68.1818%

LOGISTIC REGRESSION
Accuracy: 77.272%

RANDOM FOREST CLASSIFIER
Accuracy: 77.992%

NAÏVE BAYES CLASSIFIER
Accuracy: 82.927%

```
# randomforest classifier
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier()

model.fit(x_train, y_train)
```

RandomForestClassifier()

```
print("Accuracy is", model.score(x_test, y_test)*100)
```

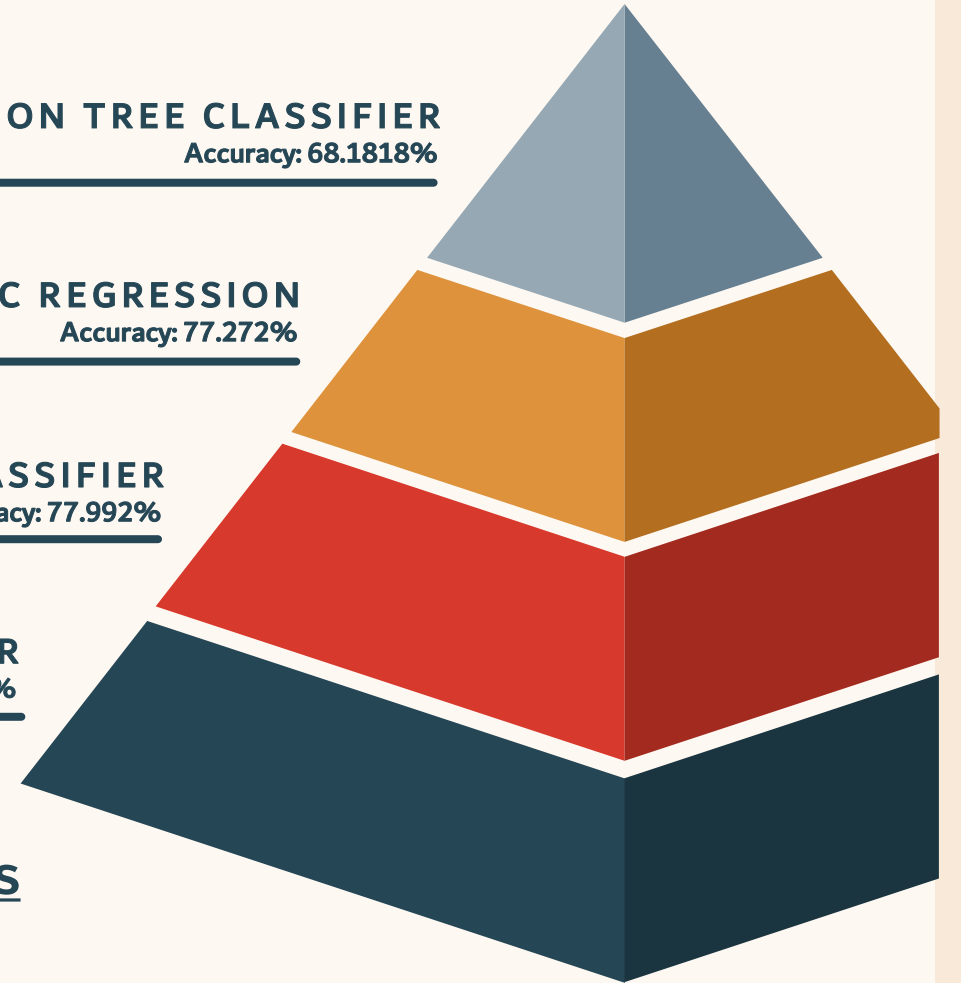
Accuracy is 77.92287792287793

```
Out[68]: array([[1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1,
1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1]])
```

```
In [69]: print('The accuracy of Naive Bayes is: ',metrics.accuracy_score(y_pred,y_test))
```

The accuracy of Naive Bayes is: 0.8292682926829268

We will use the Naïve Bayes Classifier, since it provides us the higher probability for correct prediction.



NAÏVE BAYES CLASSIFIER

1. Naive Bayes is a simple classification algorithm that uses probabilities to categorize items.
2. It assumes that features are independent and calculates the probability of an item belonging to a class based on its features.
3. The class with the highest probability is the predicted class.

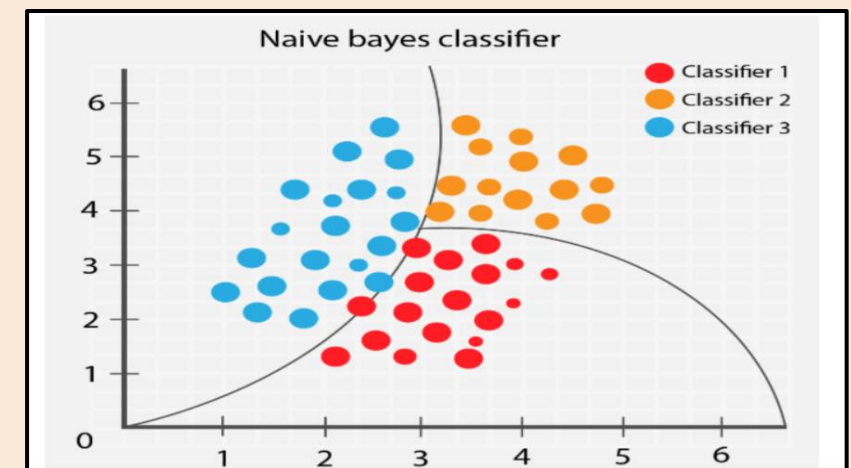
$$P(H|E) = \frac{P(E|H) * P(H)}{P(E)}$$

Likelihood of the Evidence given that the Hypothesis is True

Prior Probability of the Hypothesis

Posterior Probability of the Hypothesis given that the Evidence is True

Prior Probability that the evidence is True





DEMO IMAGES OF PROJECT

Running the Test Cases

```
In [86]: for i in range(0,5):  
         test[:,i]=labelencoder_X.fit_transform(test[:,i])
```

```
In [87]: test[:,7]= labelencoder_X.fit_transform(test[:,7])
```

```
In [88]: test
```

```
Out[88]: array([[1, 1, 0, ..., 1.0, 5720, 207],  
                [1, 1, 1, ..., 1.0, 4576, 124],  
                [1, 1, 2, ..., 1.0, 6800, 251],  
                ...,  
                [1, 0, 0, ..., 1.0, 5243, 174],  
                [1, 1, 0, ..., 1.0, 7393, 268],  
                [1, 0, 0, ..., 1.0, 9200, 311]], dtype=object)
```

```
In [89]: test= ss.fit_transform(test)
```

```
In [91]: pred= NBClassifier.predict(test)
```


Output of Test Cases in array format

```
In [92]: pred
```

[illegible]

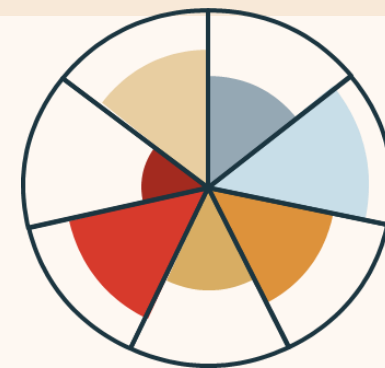
1 represents : Approval of loan

0 represents: Disapproval of loan

APPLICATIONS

- 1 In Banking Sector
- 2 Co-operate sectors which provides loans to their employees
- 3 An individual/applicant who wants to know about his capability of taking loan





THANK YOU

