An Internship Project Report on

VISA APPRROVAL PREDICTION

Submitted to
The Department of Information Technology
In partial fulfillment of the academic requirements of
Sreenidhi Institute of Science & Technology
For
The award of the degree of

Bachelor of Technology
in
Information Technology

By

B.R.ASHRITH
18311A12C7

Under the Guidance of

MD.JAFFER SADIQ

Sreenidhi Institute of Science and Technology
Yamnampet, Ghatkesar, R.R. District, Hyderabad - 501301

Affiliated to
Jawaharlal Nehru Technology University
Hyderabad - 500085

Department of Information Technology

Sreenidhi Institute of Science and Technology

The Department of Information Technology

CERTIFICATE

This is to certify that this Internship Project report on "**VISA APPROVAL PREDICTION**", submitted by B.R.ASHRITH 18311A12C7 in the year 2020 in partial fulfillment of the academic requirements of Jawaharlal Nehru Technological University for the award of the degree of Bachelor of Technology in Information Technology, is a bonafide work that has been carried out by them as part of their Internship Project during summer (2020), under our guidance. This report has not been submitted to any other institute or university for the award of any degree.

MD.JAFFER SADIQ,
Associate Professor,
Department of IT,
Internal Guide

Dr. V.V.S.S.S.
Balaram
Prof & Head, Depart

Dr. K. Sreerama Murthy,
Associate Professor,
Department of IT,
Project Coordinator

External Examiner

In Collaboration with

IBM

SMARTBRIDGE

TO WHOMSOEVER IT MAY CONCERN

This is to certify that Mr./Ms. **Ashrith .B.R**, has successfully completed the summer internship at **SmartBridge Educational Services Private Limited** from **05/18/2020 to 06/16/2020**

During the internship he/she has worked under the supervision of project mentor & developed the project entitled **"VISA Approval Prediction"**.

He/she was found hardworking, punctual and inquisitive, during the tenure of internship

We wish him/her every success in career.

Jayaprakash. Ch
Program Manager

June 18, 2020
Issued on

Powered by
Smart Internz
www.smartinternz.com

SB ID: SB20200065484

Authenticity of this certificate can be validated by going to:
https://smartinternz.com/internships/certificates/4fa7c62536118cc404dec4a0ca88d4f6

DECLARATION

I, B.R.ASHRITH 18311A12C7 student of **SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY, YAMNAMPET, GHATKESAR, of INFORMATION TECHNOLOGY** solemnly declare that the Internship project work, titled **"VISA APPROVAL PREDICTION"** is submitted to **SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY** for partial fulfillment for the award of degree of Bachelor of technology in **INFORMATION TECHNOLOGY**.

It is declared to the best of our knowledge that the work reported does not form part of any dissertation submitted to any other University or Institute for award of any degree.

ACKNOWLEDGEMENT

I would like to express my heart-felt gratitude to my parents without whom I would not have been privileged to achieve and fulfill my dreams. I am grateful to our principal, **Dr. T. Ch. Siva Reddy,** who most ably runs the institution and has had the major hand in enabling me to do my project.

I profoundly thank **Dr. V.V.S.S.S. Balaram,** Head of the Department of Information Technology who has been an excellent guide and also a great source of inspiration to my work.

I would like to thank my internal guide MD.JAFFER SADIQ for his/her technical guidance, constant encouragement and support in carrying out my project at college.

I would like to thank my coordinator **Dr K Sreerama Murthy, Associate professor,** for his technical guidance, constant encouragement and support in carrying out my project at college.

The satisfaction and euphoria that accompany the successful completion of the task would be great but incomplete without the mention of the people who made it possible with their constant guidance and encouragement crowns all the efforts with success. In this context, I would like to thank all the other staff members, both teaching and non-teaching, who have extended their timely help and eased my task.

B.R.ASHRITH
18311A12C7

Table of contents:

1.Introduction

1.1 Overview

H1-B Visa is one type of non-immigrant temporary visa granted by USCIS (United States Citizenship and Immigration Service) for the foreign nationals. These petitions are filed by the employers for their employees. This visa is also filed by international students after they get admissions into Universities. Since the number of applicants is very large than the number of selections and as the selection process is claimed to be as lottery there is no insight of how the attributes have influence over the outcome. So, we believe that a predictive model generated using all the past data can be a useful resource to predict the outcome for the applicants and the sponsors. In our project, we aim to predict the outcome of H-1B visa applications that are filed by many high-skilled foreign nationals every year. We framed the problem as a classification problem and applied in order to output a predicted case status of the application. The input to our algorithm is the attributes of the applicant. H-1B is a type of non-immigrant visa in the United States that allows foreign nationals to work in occupations that require specialized knowledge and a bachelor's degree or higher in the specific specialty. This visa requires the applicant to have a job offer from an employer in the US before they can file an application to the US immigration service (USCIS). USCIS grants 85,000 H-1B visa's every year, even though the number of applicants far exceed that number. The selection process is claimed to be based on a lottery, hence how the attributes of the applicants affect the final outcome is unclear. We believe that this prediction algorithm could be a useful resource both for the future H-1B visa applicants and the employers who are considering to sponsor them.

1.2 Purpose

Our goal for this project is to predict the case status of an application submitted by the employer to hire non-immigrant workers under the H-1B visa program. Employer can hire non-immigrant workers only after their LCA petition is approved. The approved LCA petition is then submitted as part of the Petition for a Non-immigrant Worker application for work authorizations for H-1B visa status.

2.SYSTEM ANALYSIS

System analysis is related to requirement analysis. The requirement defines the functional, non-functional, and technical requirements. Entire process is divided into steps to analyze the situation and analyze the project goals. Requirement documentation takes place in requirements analysis phase.

2.1 Functional Requirements specification

After careful analysis of the documentation the following are identified.

→LABARATOR MODULE:

Lab assistant has to test the required parameters and enter the tested results in the user interface then the system will analyze and compare with the past results and gives accurate results.

2.2 User Interface:

The system includes sample a input parameter boxes, GUI standard, screen layout constraints, standard PREDICT button and function will appear on every screen. It will run on localhost style guides are followed

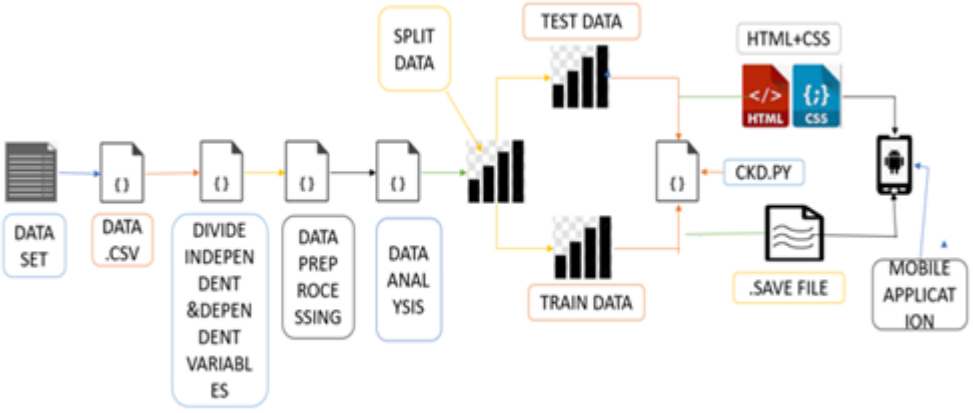
2.3 Hardware Interfaces:

- System : Dual core
- Hard Disk : 40GB
- Ram : 1 GB
- Processor : Intel P-IV based system

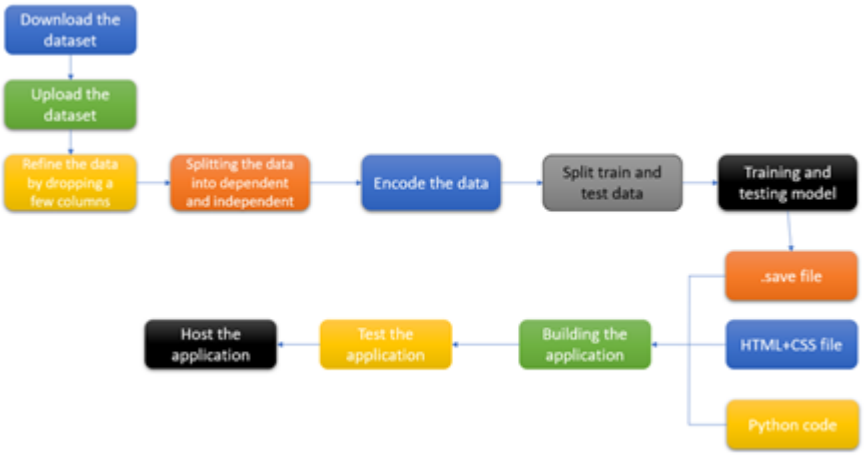
2.4 Software Requirements:

- Operating System : windows XP/7.
- Coding Language : Python,R
- IDE : Jupyter, spyder, anaconda prompt
- Front End : HTML, CSS

3.SYSTEM DESIGN



FLOW CHART:



4.LITERARY

4.1 Existing Problem

The dataset that we are studying is available on Kaggle under the name H1B Disclosure Dataset which is processed dataset from the original data. From data analysis performed on this data allow us to finding top Occupations, States, Employers and Industries that contribute to highest number of H1B visa application.

A project done by the students of UC Berkley aims to predict the waiting time to get a work visa for a given job title and for a given employer. They used KNN as the primary model to predict 'Quickest Certification Rate' across both occupations and companies.

4.2 Proposed Solution

The dataset we used for this problem is downloaded from Kaggle. It contained 10 features as shown in the Figure .

FULL_TIME_POSITION	PREVAILING_WAGE	PW_SOURCE_YEAR	PW_SOURCE_OTHER	WORKSITE_STATE	CASE_STATI
Y	59197.0	2015.0	OFLC ONLINE DATA CENTER	IL	CERTIFIEDW
Y	49800.0	2015.0	WILLIS TOWERS WATSON SURVEY	IL	CERTIFIEDW

FULL_TIME_POSITION: Positions are given in Full_time_position="Y", and Part_time_position="N". We converted them to Full Time Position = 1; Part Time Position = 0" format.

PREVAILING_WAGE: **P**revailing wage is the average wage paid to employees with similar qualifications in the intended area of employment. we are using this feature as it is.

CASE_SUBMITTED_YEAR: The year when the application was submitted.

CASE_SUBMITTED_MONTH: The month when the application was submitted.

CASE_SUBMITTED_DAY: The day when the application was submitted.

PW_SOURCE_YEAR: The is the year when the average wage paid to employees.

DECISION_DAY: The day when application got approved.

DECISION_YEAR: The year when application got approved.

DECISION_MONTH: The month when application got approved.

CASE_STATUS: The feature give us a complete prediction about either the application has been approved or denied.

5.THEORITICAL ANALYSIS

5.1 BLOCK DIAGRAM

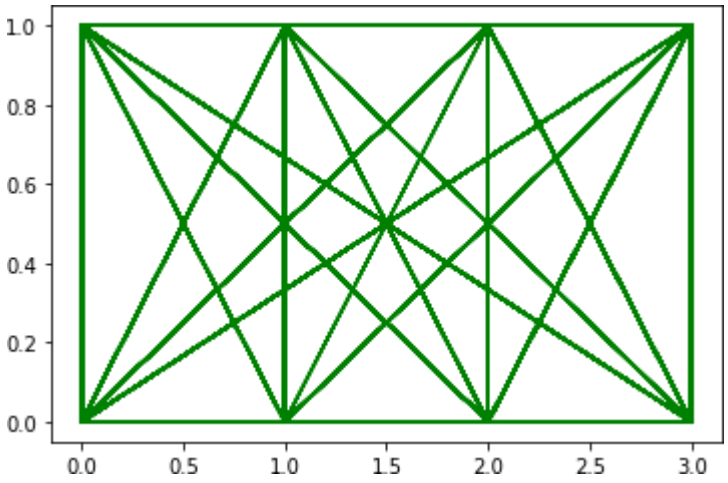


Figure: Matplot between “CASE_STATUS” AND “FULL_TIME_POSITION”

6. EXPERIMENTAL ANALYSIS

As our dataset is large and its also a classification problem, we thought of using Navie Bayes technique. Either Navie Bayes or Support Vector Machine technique can be used, but here we implemented Navie Bayes technique.

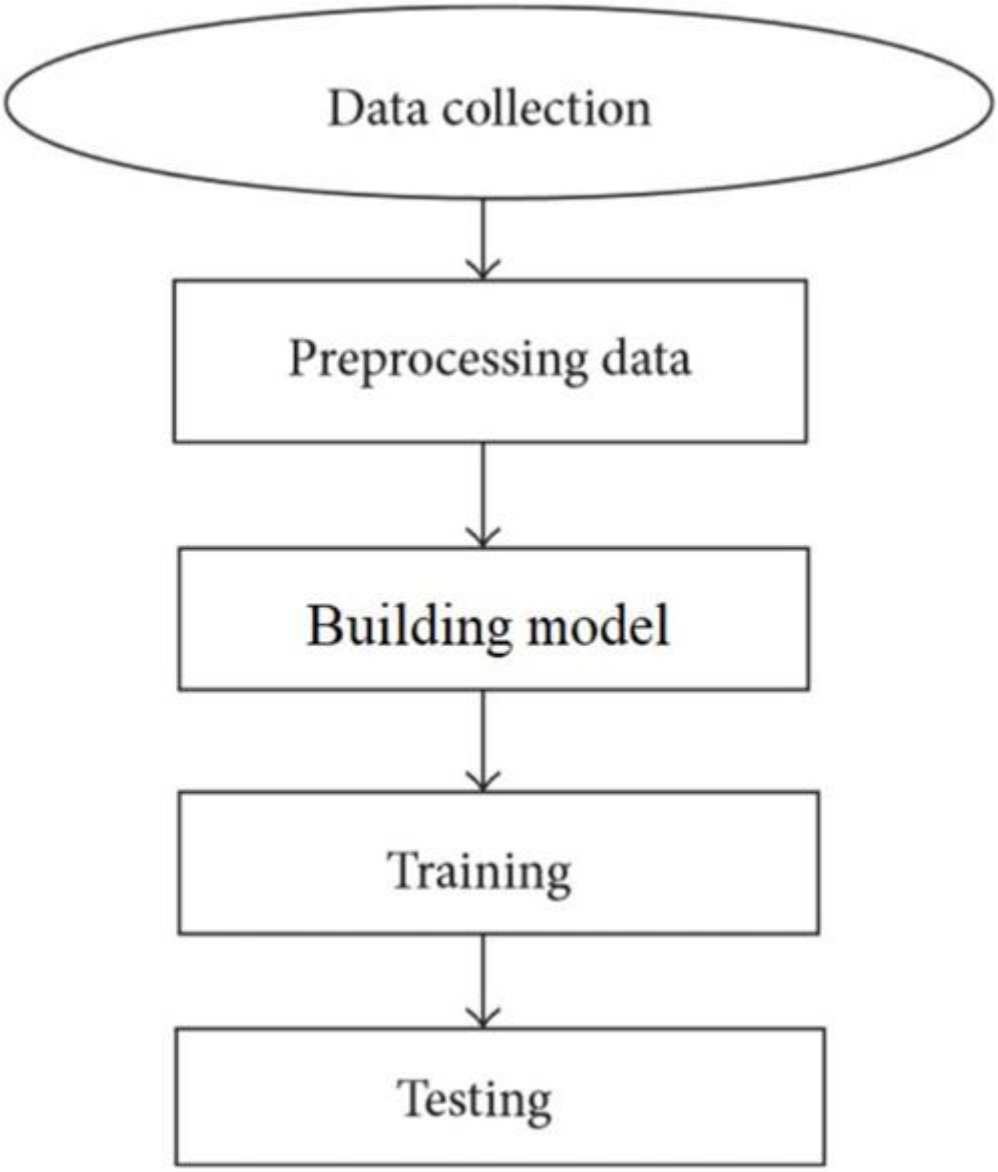
Navie Bayes:

Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem.It is a simple and interpretable model which assumes all features are conditionally independent given labels and are in gaussian distribution.

It calculates P(x/y=0), P(x/y=1) and P(y) by taking their maximum likelihood estimates in the joint likelihood of the data. While making a prediction, it considers both P(y=1) and P(y=0) on the Bayes rule and compares the two.

P(A|B) = P(B|A) * P(A) / P(B)

7.FLOWCHART\



8. SYSTEM IMPLEMENTATION

Steps for model Building:

- **Importing the Libraries**
- **Dataset Reading**
- **Data preprocessing**
- **Training and Testing the Data**
- **Machine Learning Algorithm**
- **Prediction**

8.1 Importing the libraries

8.2 Dataset Reading

Importing the Dataset

```
dataset=pd.read_csv(r'1. Master H1b Dataset.csv',encoding='latin1')
```

```
C:\ProgramData\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3063: DtypeWarning: Columns (25) have mixed types.S
pecify dtype option on import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

```
dataset
```

	CASE_SUBMITTED_DAY	CASE_SUBMITTED_MONTH	CASE_SUBMITTED_YEAR	DECISION_DAY	DECISION_MONTH	DECISION_YEAR	VISA_CLASS	EM
0	24	2	2016	1	10	2016	H1B	
1	4	3	2016	1	10	2016	H1B	
2	10	3	2016	1	10	2016	H1B	
3	28	9	2016	1	10	2016	H1B	
4	22	2	2015	2	10	2016	H1B	
...	...	...	...	...	...	...	...	
528129	30	6	2017	30	6	2017	H1B	U
528130	30	6	2017	30	6	2017	H1B	
528131	30	6	2017	30	6	2017	H1B	T
528132	30	6	2017	30	6	2017	H1B	
528133	30	6	2017	30	6	2017	E3 Australian	

528134 rows × 27 columns

8.3 Data Preprocessing

Data Vizualizing ¶

```
: dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 528134 entries, 0 to 528133
Data columns (total 27 columns):
 #   Column                 Non-Null Count   Dtype
---  ------                 --------------   -----
 0   CASE_SUBMITTED_DAY     528134 non-null  int64
 1   CASE_SUBMITTED_MONTH   528134 non-null  int64
 2   CASE_SUBMITTED_YEAR    528134 non-null  int64
 3   DECISION_DAY           528134 non-null  int64
 4   DECISION_MONTH         528134 non-null  int64
 5   DECISION_YEAR          528134 non-null  int64
 6   VISA_CLASS             528134 non-null  object
 7   EMPLOYER_NAME          528095 non-null  object
 8   EMPLOYER_STATE         528124 non-null  object
 9   EMPLOYER_COUNTRY       528134 non-null  object
 10  SOC_NAME               528134 non-null  object
 11  NAICS_CODE             528132 non-null  float64
 12  TOTAL_WORKERS          528134 non-null  int64
 13  FULL_TIME_POSITION     528131 non-null  object
 14  PREVAILING_WAGE        528134 non-null  float64
```

DATA ANALYSING

```
dataset.describe()
```

	CASE_SUBMITTED_DAY	CASE_SUBMITTED_MONTH	CASE_SUBMITTED_YEAR	DECISION_DAY	DECISION_MONTH	DECISION_YEAR	NAICS_CODE
count	528134.000000	528134.000000	528134.000000	528134.000000	528134.000000	528134.000000	528132.000000
mean	15.246426	4.444859	2016.752854	16.565991	4.552377	2016.826876	443548.680673
std	8.154979	3.201000	0.524568	8.457420	3.201752	0.379217	197875.907540
min	1.000000	1.000000	2011.000000	1.000000	1.000000	2016.000000	31.000000
25%	8.000000	3.000000	2017.000000	9.000000	3.000000	2017.000000	452910.000000
50%	15.000000	3.000000	2017.000000	17.000000	3.000000	2017.000000	541511.000000
75%	22.000000	5.000000	2017.000000	23.000000	5.000000	2017.000000	541511.000000
max	31.000000	12.000000	2017.000000	31.000000	12.000000	2017.000000	999990.000000

#taking care of missing data

dataset['FULL_TIME_POSITION'].fillna(dataset['FULL_TIME_POSITION'].mode()[0],inplace=True)

dataset['PW_SOURCE_YEAR'].fillna(dataset['PW_SOURCE_YEAR'].mode()[0,inplace=Tr)

Separating the Variables

```
x=dataset.iloc[:,0:9].values
y=dataset.iloc[:,9:10].values
```

```
x.shape
```

```
(528134, 9)
```

```
y.shape
```

```
(528134, 1)
```

```
x
```

```
array([[2.40000e+01, 2.00000e+00, 2.01600e+03, ..., 1.00000e+00,
        5.91970e+04, 2.01500e+03],
       [4.00000e+00, 3.00000e+00, 2.01600e+03, ..., 1.00000e+00,
        4.98000e+04, 2.01500e+03],
       [1.00000e+01, 3.00000e+00, 2.01600e+03, ..., 1.00000e+00,
        7.65020e+04, 2.01500e+03],
       ...,
       [3.00000e+01, 6.00000e+00, 2.01700e+03, ..., 1.00000e+00,
        7.94980e+04, 2.01600e+03],
       [3.00000e+01, 6.00000e+00, 2.01700e+03, ..., 1.00000e+00,
        1.18352e+05, 2.01600e+03],
       [3.00000e+01, 6.00000e+00, 2.01700e+03, ..., 1.00000e+00,
        4.91300e+04, 2.01600e+03]])
```

```
y
```

```
array([[0],
       [0],
       [0],
       ...,
       [0],
       [0],
       [0]], dtype=int64)
```

Label Encoding

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
dataset['FULL_TIME_POSITION']= le.fit_transform(dataset['FULL_TIME_POSITION'])
```

8.4 Training and Testing the data

Training:

The observations in the training set form the experience that the algorithm uses to learn. In supervised leaning problems, each observation consists of an observed output variables and one or more observed input variables.

Test:

The test set is a set of observations used to evaluate the performances of the model. It is important that no observations from training data set is included in test set, if it does contain it will be difficult to access whether the algorithm has learned to generalize from the training set or has simply memorized it.

Training and testing

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
```

```
x_train.shape
```

```
(422507, 9)
```

```
x_test.shape
```

```
(105627, 9)
```

```
y_train.shape
```

```
(422507, 1)
```

```
y_test.shape
```

```
(105627, 1)
```

```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
x_train=sc.fit_transform(x_train)
x_test=sc.fit_transform(x_test)
```

8.5 ML Algorithm

- As our dataset is large and its also a classification problem, we thought of using Naive bayes technique or SVM(Support Vector Machine) but here we implemented naive bayes techmique
- Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem.It is simple and interpretable model which assumes all features are conditionally independent given labels and are in guassian distribution.

Naive bayes:

Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. There is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable.

ML ALGORITHM Naive bayes

```
from sklearn.naive_bayes import GaussianNB
naive = GaussianNB()
```

```
naive.fit(x_train,y_train)
```

```
GaussianNB(priors=None, var_smoothing=1e-09)
```

```
from joblib import dump
dump(naive,'naive.save')
```

```
['naive.save']
```

```
y_pred=naive.predict(x_test)
```

8.6 Prediction

- In this work, we used different types of classifiers for determining the status of H-1B visa approval.
- We achieved the best in naive bayes which is 91.1%
- Other classifier prediction was less than 91.1%
- This leads to conclusion that how much important is selection feature and selection transformation

Evaluation of the model ¶

```
from sklearn.metrics import accuracy_score
acc=accuracy_score(y_test,y_pred)
```

```
acc
```

```
0.9111401440919462
```

9.Result

After Spliting the data in ratio 80:20, we applied Naive bayes on the data to predict the outcome. While using the naive bayes

Algorithm we used the default version of it without changing any hyper-parameter tuning.

Classifier	Accuracy
Naïve bayes	0.9111401440919462

10.ADVANTAGES & DISADVANTAGES

Advantages:

1. When assumption of independent predictors holds true, a Naive Bayes classifier performs better as compared to other models.

2. Naive Bayes requires a small amount of training data to estimate the test data. So, the training period is less.

3. Naive Bayes is also easy to implement.

Disadvantages:

1. Main imitation of Naive Bayes is the assumption of independent predictors. Naive Bayes implicitly assumes that all the attributes are mutually independent. In real life, it is almost impossible that we get a set of predictors which are completely Independent.

2. If categorical variable has a category in test data set, which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as Zero Frequency.

11. APPLICATIONS

1. Real-time Prediction: As Naive Bayes is super fast, it can be used for making predictions in real time.

2. Multi-class Prediction: This algorithm can predict the posterior probability of multiple classes of the target variable.

3. Text classification/ Spam Filtering/ Sentiment Analysis: Naive Bayes classifiers are mostly used in text classification (due to their better results in multi-class problems and independence rule) have a higher success rate as compared to other algorithms. As a result, it is widely used in Spam filtering (identify spam e-mail) and Sentiment Analysis (in social media analysis, to identify positive and negative customer sentiments).

4.Recommendation System: Naive Bayes Classifier along with algorithms like Collaborative Filtering makes a Recommendation System that uses machine learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not.

PYTHON CODE

```
import numpy as np
from flask import Flask, request, jsonify, render_template
from joblib import load
app = Flask(__name__)
model = load('naive.save')

@app.route('/')
def home():
    return render_template('index1.html')

@app.route('/y_predict',methods=['POST'])
def y_predict():
    '''
    For rendering results on HTML GUI
    '''
    x_test = [[int(x) for x in request.form.values()]]
    print(x_test)
    prediction = model.predict(x_test)
    print(prediction)
    output=prediction[0]
    return render_template('index1.html', prediction_text='CASE_STATUS
```

```
{}'.format(output))


@app.route('/predict_api',methods=['POST'])
def predict_api():
    '''
    For direct API calls trought request
    '''
    data = request.get_json(force=True)
    prediction = model.y_predict([np.array(list(data.values()))])

    output = prediction[0]
    return jsonify(output)

if __name__ == "__main__":
    app.run(debug=True)
```

HTML CODE

```
<!DOCTYPE html>

<html >

<!--From https://codepen.io/frytyler/pen/EGdtg-->

<head>

  <meta charset="UTF-8">

  <title>ML API</title>

  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>

<link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">


<style>

.login{

top: 20%;

}

</style>

</head>


<body>

 <div class="login">

      <h1>Visa Approval Prediction</h1>
```

```
 <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('y_predict')}}"method="post">
                </select>
        <input type="text" name="CASE_SUBMITTED_DAY"
placeholder="CASE_SUBMITTED_DAY" required="required" />
        <input type="text" name="CASE_SUBMITTED_MONTH"
placeholder="CASE_SUBMITTED_MONTH" required="required" />
                <input type="text" name="CASE_SUBMITTED_YEAR"
placeholder="CASE_SUBMITTED_YEAR" required="required" />
                <input type="text" name="DECISION_DAY" placeholder="DECISION_DAY"
required="required" />
                <input type="text" name="DECISION_MONTH"
placeholder="DECISION_MONTH" required="required" />
                <input type="text" name="DECISION_YEAR" placeholder="DECISION_YEAR"
required="required" />
                <input type="text" name="FULL_TIME_POSITION"
placeholder="FULL_TIME_POSITION" required="required" />
        <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
    </form>
   <br>
   <br>
   {{ prediction_text }}
 </div>
 </body>
</html>
```

OUTPUT SCREEN