**VISA APPROVAL PREDICTION**

Using Naive Bayes Classifier

### Developed by: MANDA ANIL,MAJJIGA KARTHIK,CHIPPA SHIVATEJA,KAITHA SAMBARAJU

**Smartinternz Externship ,TEAM N0:020**

# INTRODUCTION

This project deals with the task of predicting the outcome of immigration visa applications by using machine learning methods. The prediction task is treated as one of supervised inductive learning of classification functions. Exploratory data analysis was also performed to analyze the main characteristics of the data set. A prediction model was developed which trains on historical data of the same application domain and uses machine learning classifiers to predict the outcome of visa petitions.

The H-1B visa allows companies and organizations in the United States to employ foreign workers in specialty occupations that require major technical expertise in fields such as accounting, architecture, engineering, finance, information technology, mathematics, medicine, science, etc. A bachelor’s degree or equivalent work experience are some of the basic requirements for a H-1B visa. A job offer should also be forwarded to the worker from the employer filing the petition.

## Overview

In the Guided project,Our goal is to predict the outcome of H-1B visa applications that are filled by many professional foreign nationals every year.This visa requires applicant to have a job offer from an employer in the US.

## 1.2 Purpose

Our aim from the project is to make use of pandas, matplotlib,&seaborn libraries from python to extract the libraries for machine learning for the Visa prediction.

Secondly, to learn how to hyper tune the parameters using grid search cross validation for theNaiveBayes machine learning algorithm.

And in the end, to predict whether the Visa applicant can replay the Visa or not using voting ensemble techniques of combining the predictions from multiple machine learning algorithms and withdrawing the conclusions.

# 2. LITERATURE SURVEY

Data mining is the process of analyzing data from different perspectives and extracting useful knowledge from it. It is the core of knowledge discovery process. The various steps involved in extracting knowledge from raw data as depicted in figure-1. Different data mining techniques include classification, clustering, association rule mining, prediction and sequential patterns, neural networks, regression etc. Classification is the most commonly applied data mining technique, which employs a set of preclassified examples to develop a model that can classify the population of records at large. Fraud detection and credit risk applications are particularly well suited to classification technique. This approach frequently employs Decision tree based classification Algorithm. In classification, a training set is used to build the model as the classifier which can classify the data items into its appropriate classes.

A test set is used to validate the model.

## 2.1 Proposed Solution

### Machine Learning (Naive Bayes):

Naive Bayes algorithm in machine learning methods which efficiently performs both classification and regression tasks. Naive Bayes is a kind of classifier which uses the Bayes Theorem. It predicts membership probabilities for each class such as the probability that given record or data point belongs to a particular class. The class with the highest probability is considered as the most likely class. And the mot likely class will be the output predicted for the loan estimation.

And also we have created an UI using the Flask for the loan status prediction, this UI will allow the users to predict the loan status very easily and the User interface is user friendly not at least one complication in using the interface, and it can be used just by entering some necessary details into the UI in real time it'll give the predicted value like if the customer is beneficial to take a loan and how often does he pays the loan interest amount to the bank.

Basically this model will give the predicted value when a customer with details will pay the loan back to bank, by just taking some necessary details of the customer in real time, and those details will be collected by bank employee within minutes.

# 3. THEORETICAL ANALYSIS

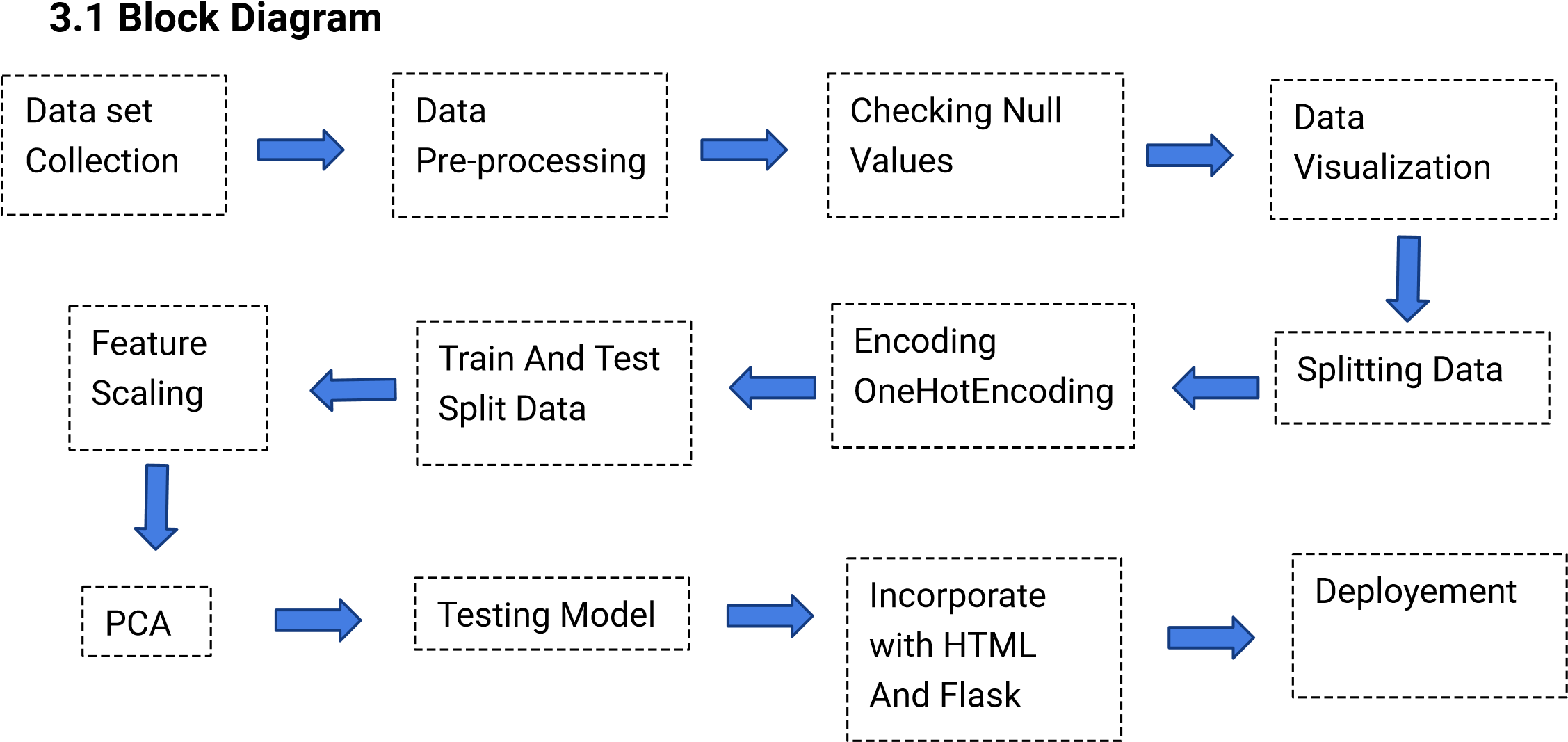
While selecting the algorithm that gives an accurate prediction we gone through lot of algorithms which gives the results abruptly accurate and from them we selected only one algorithm for the prediction problem that is Naive Bayes Classifier, it assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

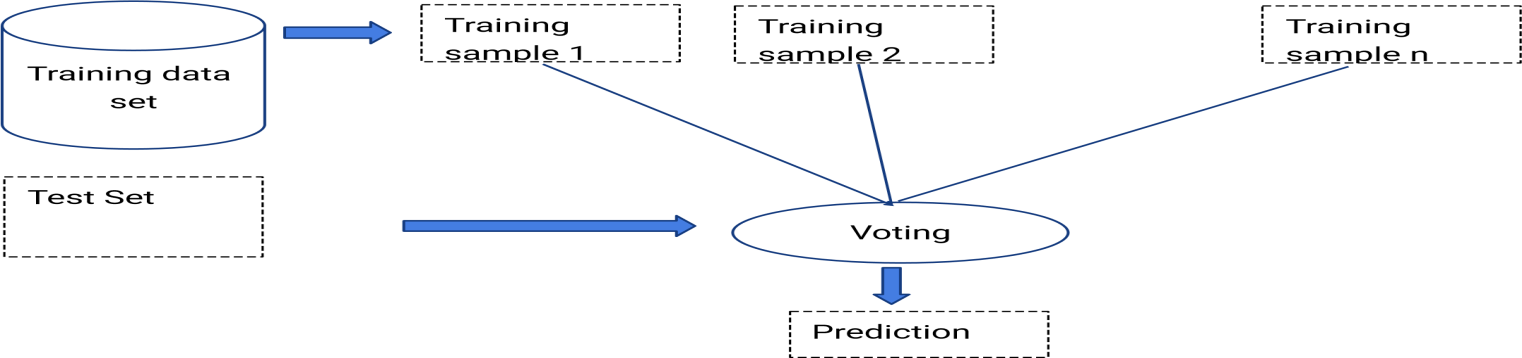
thats how the prediction work great with the Naive Bayes Algorithm.

The peculiarity of this problem is collecting the customers details real time and working with the prediction at the same time, so we developed an user interface for the people who'll be accesssing for the Visa status prediction. Accuracy is defined as the ratio of the number of samples correctly classified by the classifier to the total number of samples for a given test data set. The formula is as follows

Accuracy=TP+TN/TP+TN+FT+FN

At first we got like lot of worst accuracies because we tried lot of algorithms for the best accurate algorithm , finally after all of that we tried the best suitable algorithm which gives the prediction accurately is Naive Bayes Classifier. And developed it to use as a real time prediction probelm for the visa status prediction.





## 3.2 Software Designing

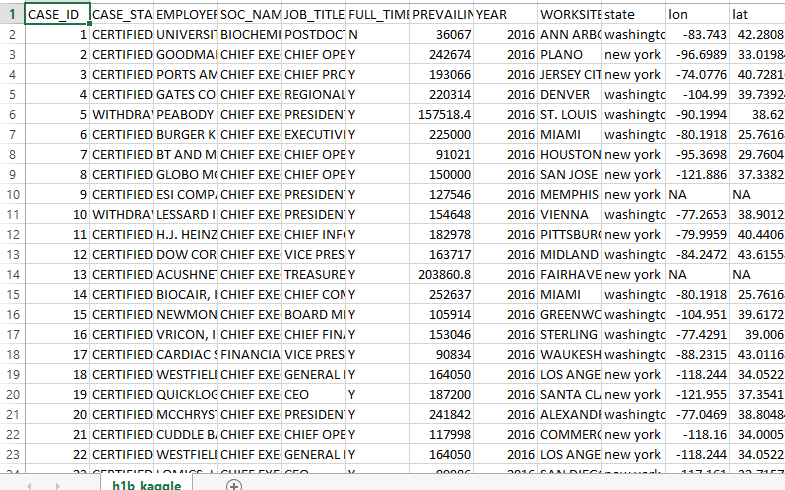
* Jupyter Notebook Environment
* Spyder Ide
* Machine Learning Algorithms
* Python (pandas, numpy, matplotlib, seaborn, sklearn)
* HTML
* Flask

We developed this Visa Approval status prediction by using the Python language which is a interpreted and high level programming language and using the Machine Learning algorithms. for coding we used the Jupyter Notebook environment of the Anaconda distributions and the Spyder, it is an integrated scientific programming in the python language.

For creating an user interface for the prediction we used the Flask. It is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions, and a scripting language to create a webpage is HTML by creating the templates to use in th functions of the Flask and HTML.

# 4. EXPERIMENTAL INVESTIGATION

In this paper, the dataset we used is derived from H-1B\_Kaggle .It contains more than 10L H-1B Visa data of users .It contained 7 features and 1 label which can be examined attributes. Those attributes were shown below in the screenshot of the data set we used.



CASE STATUS: We excluded the cases ’CERTIFIED-WITHDRAWN’ and ’WITHDRAWN’, since ’WITHDRAWN’ decisions are either made by the petitioning employer or the applicant, therefore not predictive of USCIS’s future behavior. We labeled ’CERTIFIED’ cases as 1 and ’DENIED’ cases as 0.

FULL TIME POSITION: Positions are given in ”Full Time Position = Y; Part Time Position = N” format. We converted them to ”Full Time Position = 1; Part Time Position = 0” format. YEAR: Year in which application was ﬁled. We converted the data into one-hot-k representation.

PREVAILING WAGE: Prevailing wage is the average wage paid to employees with similar qualiﬁcations in the intended area of employment. We discarded the outlier terms and used the rest of the data as it was. APPS PER

EMPLOYER\_NAME: We created a feature for the number of H-1B applications per employer, and discarded data points that are petitioned by an employer that has less than 4 applications. Although this processing step undesirably gets rid of applications ﬁled by small companies, it signiﬁcantly helps with cleaning up the misspelled company names.

We created a feature for the success rate per employer. APPS PER

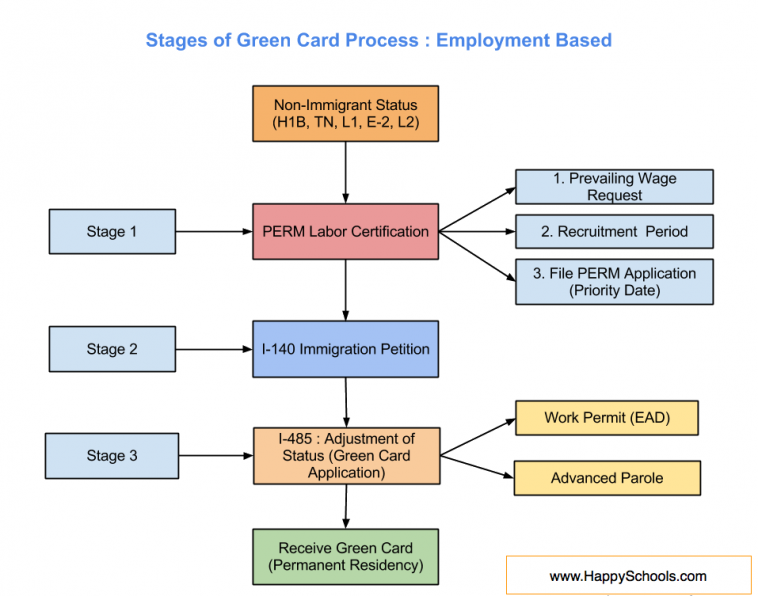
SOC\_NAME: SOC stands for Standard Occupational Classiﬁcation System, which is a federal occupational classiﬁcation system. We created a feature for the number of H-1B applications per SOC type, and discarded data points with SOC types that appear less than 4 times in the data. This processing step undesirably gets rid of applications with uncommon jobs, but helps with cleaning up .

WORKSITE: Data is given in the ”City, State” format. We only included ”State” and converted the data

into one-hot-k representation.

After the pre-processing steps described above, we split the training, Training set had a total of 1.2 million examples. Due to the inherent bias in our dataset towards the ”CERTIFIED” label, we created two versions and test sets in order to make sense of the error analysis later on. First version of dev and test sets were both unbalanced, each consisting of 400K examples. More speciﬁcally, around 90% of the examples had a ”CERTIFIED” label, mimicking the nature of the original dataset. Second version of dev and test sets were manually balanced by sampling ”CERTIFIED” labeled examples roughly equal to the number of ”DENIED” labeled examples.

1. **Process Flow of Project**



1. **FLOW CHART**

# C:\Users\hghyj\Desktop\moom\viv.jpg 7. RESULT

In this paper, the Naive Bayes algorithm is used to predict its performance, and compared with another machine learning methods namely the decision tree, the logistic regression and the SVM. The obtained results are displayed in Table below. The results show that, the performance of Naive Bayes have comparable performance than that of logistic regression, random forest, SVM and decision tree, but the Naive Bayes still performs the best, with an accuracy of 98%,

# 8. ADVANTAGES AND DISADVANTAGES

### Advantages:

* Naive Bayes give the accurate result of the prediction upto 98% which is the algorithm we used for prediction.
* H-1B visa benefit,and perhaps the main reason for its popularity, is the board requirements associated with qualifying for the visa
* Duration of Stay
* Portability
* Anyone Can Apply
* Dual Intent (pursue legal permanent residency) while under H-1B non-immigrant status.

### Disadvantages:

* Lottery.
* Extensions.
* Due to lottery process,there are strict dates the must be adhgered to during process.
* Fees.

# 9. CONCLUSION

Inorder to predict the outcome of H-1B visa applications based on the attributes of the applicant ,several machine learning models like SVM, Naive Bayes can be used.

Finally,this can be integrated to a web appliction.

# 10. FUTURE SCOPE

In further Naive Bayes algorithm can be applied on other data sets available for visa approvals to furtherinvestigate its accuracy. A rigorous analysis of other machine learning algorithms other than these six can also be done in future to investigate the power of machine learning algorithms for visa status prediction. In further study, we will try to conduct experiments on larger data sets or try to tune the model so as to achieve the state -of-art performance of the model and a great UI support system making it complete web application model.

# 11. BIBLIOGRAPHY

1. “H-1B Fiscal Year (FY) 2018 Cap Season,” USCIS. [Online]. Available: https://www. uscis.gov/working-united-states/temporary-workers/h-1b-specialty-occupationsand-fashion-models/h-1b-fiscal-year-fy-2018-cap-season. [Accessed: 20-Oct-2017].
2. “High-skilled visa applications hit record high,” CNNMoney. [Online]. Available: http: //money.cnn.com/2016/04/12/technology/h1b-cap-visa-fy-2017/index.html. [Accessed: 20-Oct-2017].
3. “Using Text Analysis To Predict H-1B Wages,” The Official Blog of BigML.com, 01-Oct2013. [Online]. Available: https://blog.bigml.com/2013/10/01/using-text-analysisto-predict-h1-b-wages/ . [Accessed: 20-Oct-2017].
4. “Predicting Case Status of H-1B Visa Petitions.” [Online]. Available: https://cseweb. ucsd.edu/classes/wi17/cse258-a/reports/a054.pdf.
5. “H-1B Visa Data Analysis and Prediction by using K-means Clustering and Decision Tree Algorithms.” [Online]. Available: https://github.com/Jinglin-LI/H1B-VisaPrediction-by-Machine-Learning-Algorithm/blob/master/H1B\%20Prediction\%20Research\ %20Report.pdf.
6. H-1B Visa Petitions 2011-2016 — Kaggle. [Online]. Available: https://www.kaggle. com/nsharan/h-1b-visa/data. [Accessed: 20-Oct-2017].
7. A. Ng, “CS229 Lecture Notes” [Online]. Available: http://cs229.stanford.edu/notes/. [Accessed: 14-Dec-2017].
8. Wes McKinney. Data Structures for Statistical Computing in Python, Proceedings of the 9th Python in Science Conference, 51-56 (2010)
9. FabianPedregosa, Ga¨elVaroquaux, AlexandreGramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, AlexandrePassos, David Cournapeau, MatthieuBrucher, Matthieu Perrot, EdouardDuchesnay. Scikit-learn: Machine Learning in Python, Journal of Machine Learn- ´ ing Research, 12, 2825-2830 (2011)
10. St´efan van der Walt, S. Chris Colbert and Ga¨elVaroquaux. The NumPy Array: A Structure for Efficient Numerical Computation, Computing in Science & Engineering, 13, 22-30 (2011)

11. Fernando P´erez and Brian E. Granger.IPython: A System for Interactive Scientific Computing, Computing in Science & Engineering, 9, 21-29 (2007)

# APPENDIX

### HTML: home page

<!DOCTYPE html>

<html>

<head>

<style>

body {

background-image: url("https://images.app.goo.gl/HYRuUkg9yX5kHNzT9");

background-position: 100%;

background-repeat: no-repeat;

background-size: cover;

background-color: #cccccc;

}

h1 {

text-align: center;

font-size:70px;

text-transform: uppercase;

color: #000000;

}

h4{

text-align: center;

font-size: 40px;

color: #eeffa3;

}

a:link, a:visited {

background-color: #f44336;

color: white;

padding: 19px 29px;

text-align: center;

text-decoration: none;

display: inline-block;

vertical-align: middle;

margin-left:700px;

}

a:hover, a:active {

padding: 0px 5px

vertical-align: middle;

background-color: red;

}

mark{background-color:white;

}

mark1{ background-color:#5c6631;

}

</style>

</head>

<body>

<h1><mark>Welcome to<br> Visa Approval Status<br>Analysis</mark></h1><br><br>

<h4><mark1><br>Here you can analyse the visa approval status</mark1></h4><br><br>

<a href="{{url\_for('indexp')}}">ANALYSE</a>

</body>

</html>

### HTML: Indexp

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>

\* {

box-sizing: border-box;

}

input[type=text],[type=number], select, textarea {

width: 70%;

padding: 12px;

border: 1px solid #ccc;

border-radius: 15px;

resize: vertical;

}

label {

padding: 12px 12px12px 0;

display: inline-block;

color:white;

}

vi{

color:white;

}

button[type=submit] {

background-color: #4CAF50;

color: white;

padding: 12px 20px;

border: none;

border-radius: 14px;

float: center;

}

button[type=submit]:hover {

background-color: #45a049;

}

.container {

border-radius: 200px;

background-color: #667292;

padding: 45px;

}

.col-25 {

float: left;

width: 20%;

margin-top: 6px;

}

.col-75 {

float: left;

width: 70%;

margin-top: 6px;

}

/\* Clear floats after the columns \*/

.row:after {

content: "";

display: table;

clear: both;

}

</style>

</head>

<body>

<center>

<h2>VISA Approval Prediction</h2>

</center>

<div class="container">

<form action="{{ url\_for('y\_predict')}}"method="post">

<center>

<div class="row">

<div class="col-25">

<label for="FULL\_TIME\_POSITION">Duration of work :</label>

</div>

<div class="col-75">

<select name="FULL\_TIME\_POSITION" required="required" >

<option value="">Duration of work </option>

<option value="1">Full Time</option>

<option value="0">Part Time </option>

</select>

</div>

</div>

<br>

<br>

<div class="row">

<div class="col-25">

<label for="PREVAILING\_WAGE">Prevailing wage :</label>

</div>

<div class="col-75">

<input type="number" name="PREVAILING\_WAGE" placeholder="PREVAILING WAGE" required="required"/>

</div>

</div>

<br>

<br>

<div class="row">

<div class="col-25">

<label for="YEAR">Year :</label>

</div>

<div class="col-75">

<select name="YEAR" required="required" >

<option value="">year </option>

<option value="2015">2015</option>

<option value="2016">2016</option>

</select>

</div>

</div>

<br>

<br>

<div class="row">

<div class="col-25">

<label for="state">State :</label>

</div>

<div class="col-75">

<select name="state" required="required" >

<option value="">Select State </option>

<option value="new york">new york</option>

<option value="washington">washington</option>

<option value="indiana">indiana</option>

</select>

</div>

</div>

<br>

<br>

<div class="row">

<div class="col-25">

<label for="occupation name">Occupation :</label>

</div>

<div class="col-75">

<select name="SOC\_NAME1" required="required" >

<option value="">occupation name </option>

<option value="it">it</option>

<option value="others">others</option>

<option value="manager">manager</option>

</select>

</div>

</div>

<br>

<br>

<button type="submit" class="row">Predict</button>

</form>

<br>

<br>

<h2>

<vi>

{{ prediction\_text }}

</vi>

</h2>

</div>

</head>

<body>

</center>

</body>

</body>

</html>

</center>

</body>

</html>

</div>

</body>

</html>

### APP.PY:

importnumpy as np

from flask import Flask, request, jsonify, render\_template

fromjoblib import load

app = Flask(\_\_name\_\_)

model= load('vi.save')

trans=load('vivtransform')

trans2=load('vivtransform2')

@app.route('/')

def home():

returnrender\_template('homepage.html')

@app.route('/indexp')

defindexp():

returnrender\_template('indexp.html')

@app.route('/y\_predict',methods=['POST'])

defy\_predict():

'''

For rendering results on HTML GUI

'''

x\_test = [[x for x in request.form.values()]]

print(x\_test)

test=trans.transform(x\_test)

test=test[:,1:]

test=trans2.transform(test)

test=test[:,1:]

print(test)

prediction = model.predict(test)

print(prediction)

if prediction==0:

output="Certified"

else:

output="denied"

returnrender\_template('indexp.html', prediction\_text=' {}'.format(output))

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)