## horizontal line



Visa Approval Prediction

Team Members

18BCE0856 - Atharva Chaudhary

18BIS0032 - Ashwath A

18BCI0246 - Lakshit Mangla

18BEC0460 - T Yashwanth

# 

# Introduction

Over 2 Million visa petitions are filed by the employers each year and only 65000 petitions are approved. So, the goal is to explore the petitions filed and their outcomes for the past six years i.e., from 2011 to 2016, and to find a pattern to predict the outcome by using a predictive model developed using Machine Learning techniques.

Our goal is to predict the outcome of H-1B visa applications that are filed by many professional foreign nationals every year. Here, we framed the problem as a classification problem and applied it 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). 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 sponsoring them.

In order to predict the case status of the applicants, we will be feeding the model with the dataset which contains the required fields by which the machine can classify the case status as certified or denied.

# Problem Statement

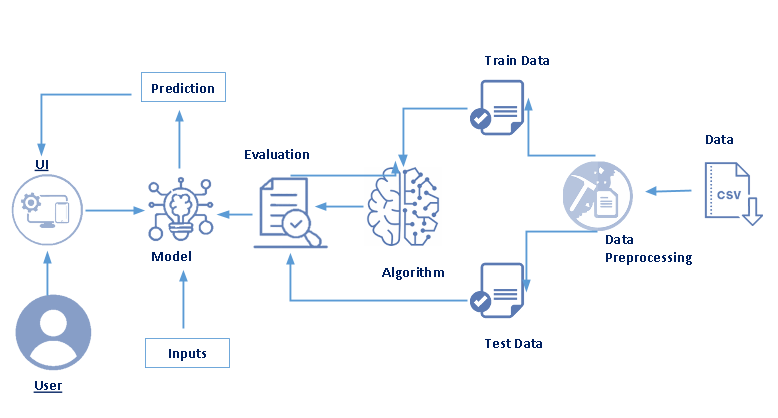
The objective of the project is to predict the outcome of H-1B visa applications that are filed by many professional foreign nationals every year.

# Literature Survey

## [1] Prediction of H1B Visa Using Machine Learning Algorithms:

The authors have estimated the likelihood of visa approval on the basis of metadata provided. A classifier has been designed which serves as a dual purpose for the H1B applicants as well as the employers to measure the probability of getting certified. They have used logistic regression and random forest classifier and found the first works better than the later.

# Flow Chart



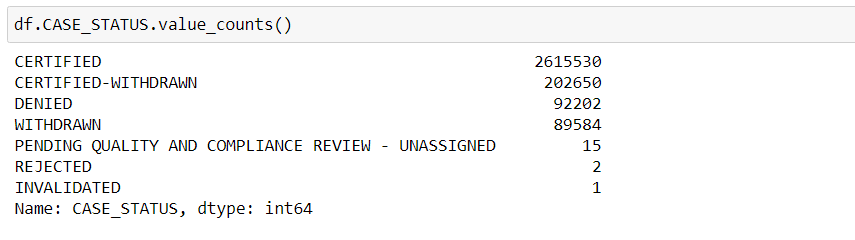
The process is as follows:

1. First, the neural network model is trained on the dataset for different output values for numerous data.
2. The user submits their details through the website.
3. The flask application collects this data from the website and provides it as input to the model.
4. The model classifies this input data into one of the classes and the prediction is displayed back in the website.

# Dataset

# 

The dataset consists of ten columns and four columns namely Soc Name, Full time position, prevailing wage and year impact Case Status the most and hence selected as input.



The Case Status consists of seven classes for the model to predict. Equal number of rows from each class are trained.

# Solution

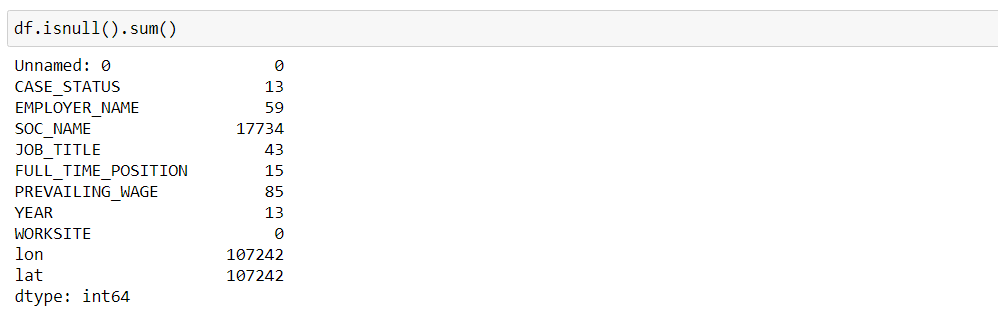
The problem is framed as a classification problem where the model is trained on the dataset and the output is within the categories such as Certified, Denied, etc.,.

The steps involved are:

1. Data preprocessing
2. Train and test data
3. Build Model
4. Flask Application

## Data Pre-Processing

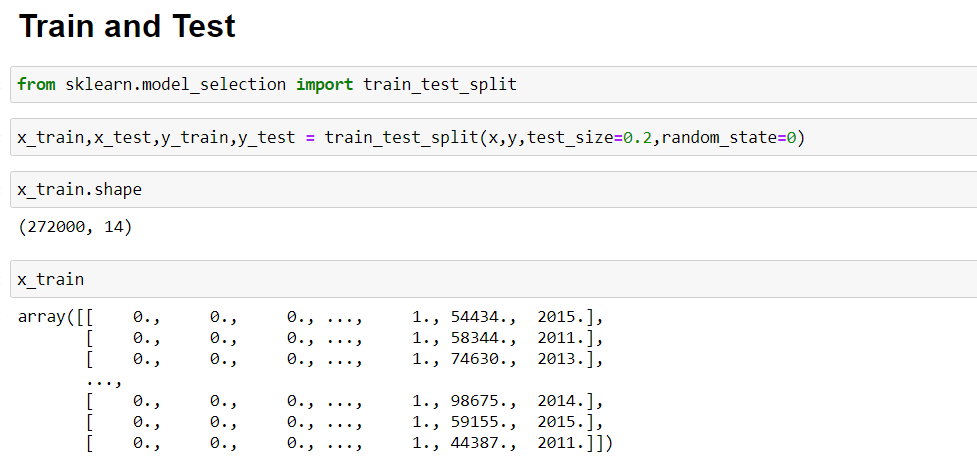
First the dataset is read into the jupyter notebook. Missing data are found if any. The rows consisting of the null data under the output column are dropped from the dataset. Other columns where the data are missing, are replaced by the mean or mode of their respective columns.



As it is a classification problem, the data for each output class should be equal. The classes are observed to have different numbers of rows and hence equal amounts of data are sampled for each class. The columns having categorical data are converted using One Hot Encoder.

## Train and Test Data

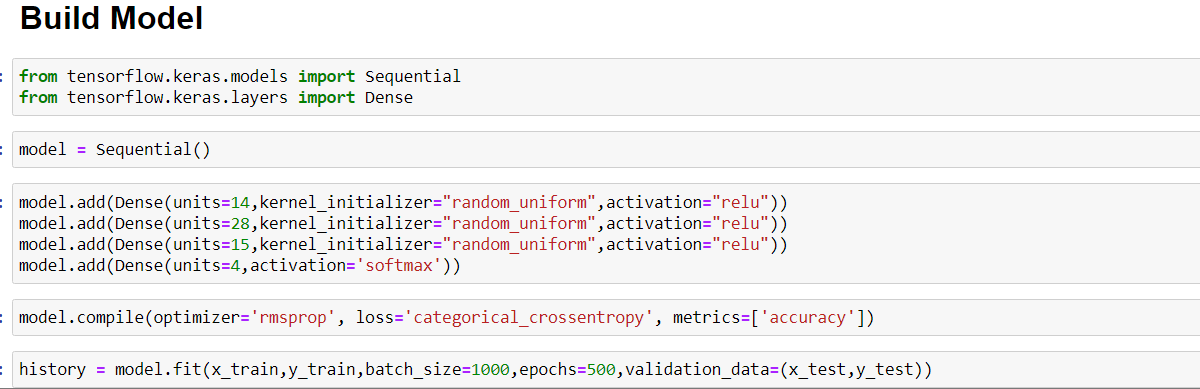
Next step is to split the data into train and test data. For that, we need to identify the input columns which matters the most or which impacts the output.



Four columns namely Soc Name, Full time position, prevailing wage and year are selected as input to the model. The output column is Case Status which will be the prediction of the model. The data for training is taken as 80% and for the test is taken as 20% randomly from the preprocessed dataset.

## Build Model and Evaluate

The model is built using the keras library. At first, the neural network is added the input nodes which are equal to the number of input columns in the dataset with relu as the activation function. Further, hidden layers are added for increasing the accuracy of the model with relu activation function.



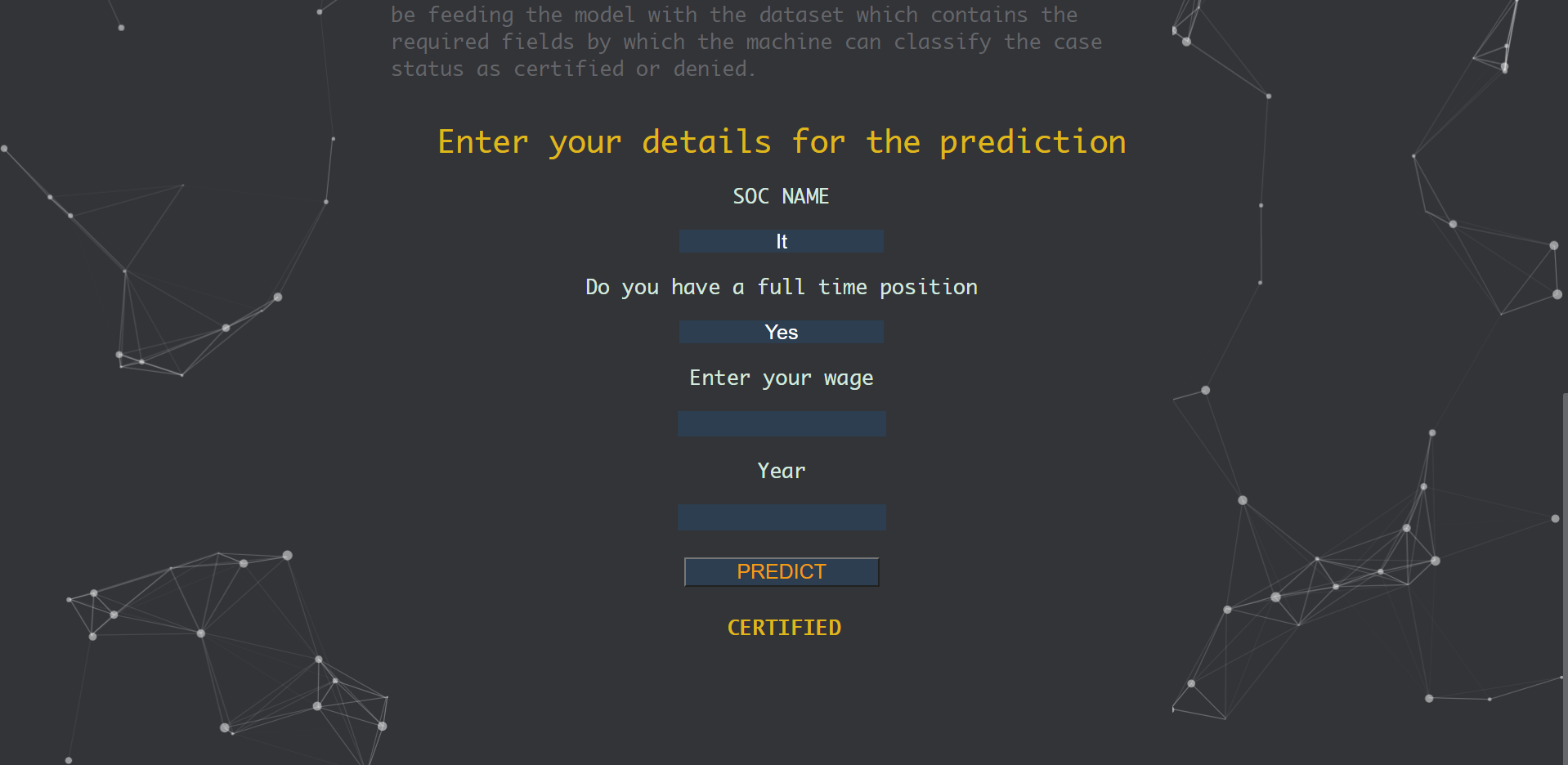
At the final output layer, the number of nodes added are equal to the number of classes present in the Case Status columns with softmax as the activation function. After fitting the model with certain epochs, input values are given for prediction. Finally the model is saved.

## Flask Application

A website is created for the user to input values and observe the output. Flask application is made in order to integrate the model with the website. The front-end is written in HTML and CSS while the back-end is programmed using python.

A form is presented in the webpage and the user inputs their details. Once the form is submitted, the push request is handled by the application where the model predicts for the values given by the user, The resultant prediction is displayed in the website.

# Output



# Conclusion

The Visa Approval Prediction is successfully implemented. The user can input their details in the website and the model predicts the Case Status for them. The resultant prediction can be viewed on the website.

# Future Scope

Nationality and other features can also be considered as factors if it is included in the dataset in the future. The model and the website can be deployed in a public server so that the accessibility increases.

# References

[1] D. Swain, K. Chakraborty, A. Dombe, A. Ashture and N. Valakunde, "Prediction of H1B Visa Using Machine Learning Algorithms," 2018 International Conference on Advanced Computation and Telecommunication (ICACAT), 2018, pp. 1-7, doi: 10.1109/ICACAT.2018.8933628.