PROJECT DOCUMENT

Topic: Predictive Modelling for H1b Visa Approval Using IBM Watson

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.

In the Guided Project, 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.

Overview

Companies file approximately 2 million visa applicants each year, however only 65000 of them are authorized. So, the idea is to look at all of the petitions that

have been submitted and their outcomes over the last six years, from 2011 to 2016, and see if there is a pattern that can be used to estimate the outcome using a predictive model constructed using Machine Learning techniques.

Purpose

The objective is to forecast the outcome of H-1B visa applications, which are filed by a large number of professional foreign nationals each year. In this instance, we framed the challenge as a classification problem and used it to anticipate the application's case state.

Existing problem

Due to the limited number of granted Visas, assessing the acceptance or denial of the same is challenging. As a result, we can easily predict the conclusion using machine learning techniques. H-1B visas are a category of employment-based, non-immigrant visas for temporary foreign workers in the United States.
For a foreign national to apply for an H1-B visa, a US employer must offer them a job and submit a petition for a H-1B visa to the US immigration department.

This is also the most common visa status applied for and held by international students once they complete college or higher education and begin working in a full-time position. Due to the limited number of granted Visas, assessing the acceptance or denial of the same is challenging. As a result, we can easily predict the conclusion using machine learning techniques.

Proposed solution

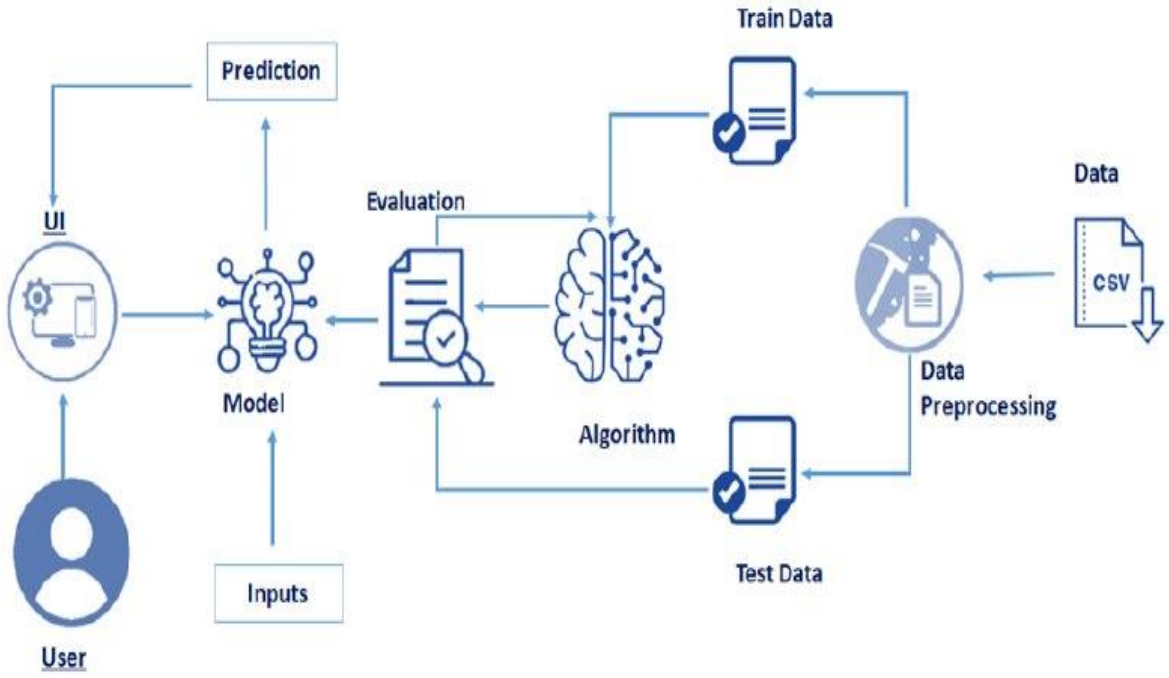
For prediction, we use conventional machine learning methods such as classification, random forest, and decision trees. For quick user interaction, the model will be deployed on Flask.

Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction.

The fundamental concept behind random forest is a simple but powerful one — the wisdom of crowds. In data science speak, the reason that the random forest model works so well is:

"A large number of relatively uncorrelated models (trees) operating as a committee will **outperform** any of the individual constituent models."

Block Diagram:



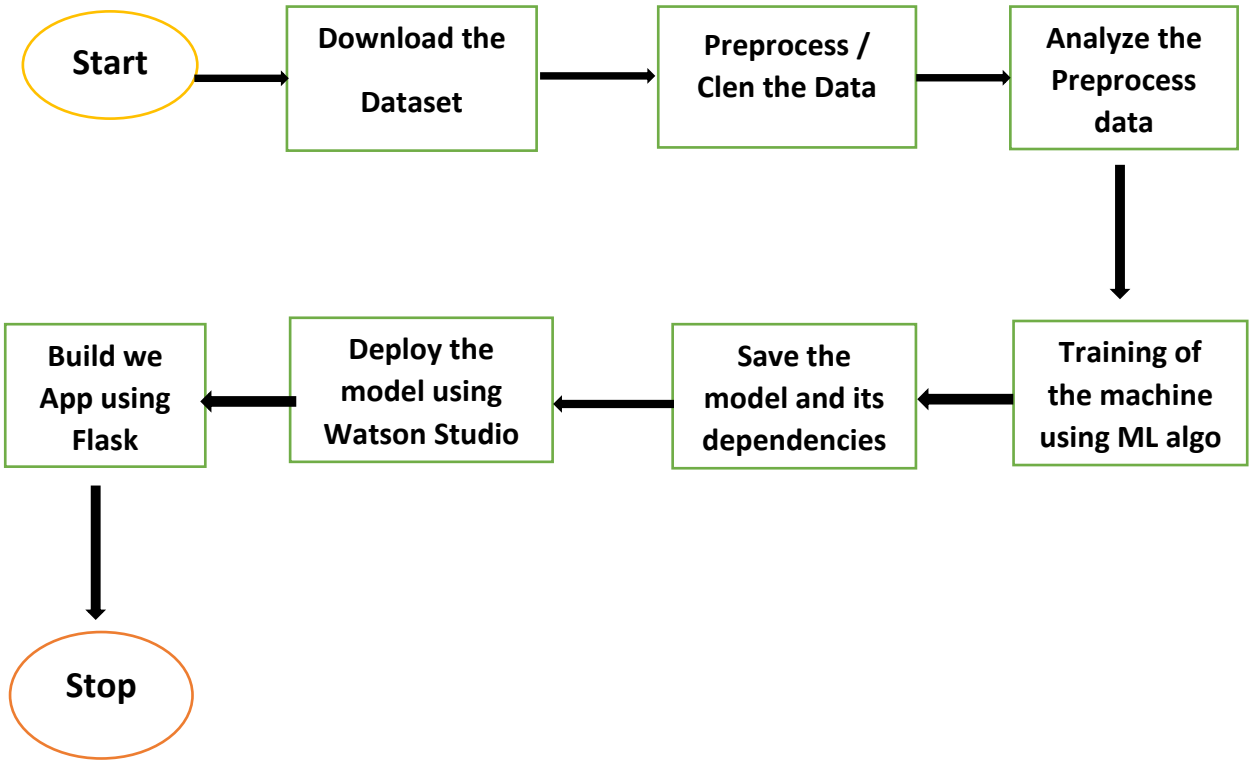
Flask is a **web development framework** developed in Python. Some features which make Flask an ideal framework for web application development are:

1. Flask provides a **development server** and a **debugger**.
2. It uses **Jinja2** templates.
3. It is compliant with **WSGI 1.0**.
4. It provides integrated support for **unit testing**.

5. Many extensions are available for Flask, which can be used to enhance its functionalities.

The reason why we deploy Machine Learning models is that we need them to be available for use to other people. The goal of building a Machine Learning model is to solve a problem, and a Machine Learning model can only do so if it is in production and is actively used by customers. In such cases, **deployment is as important as model building.**

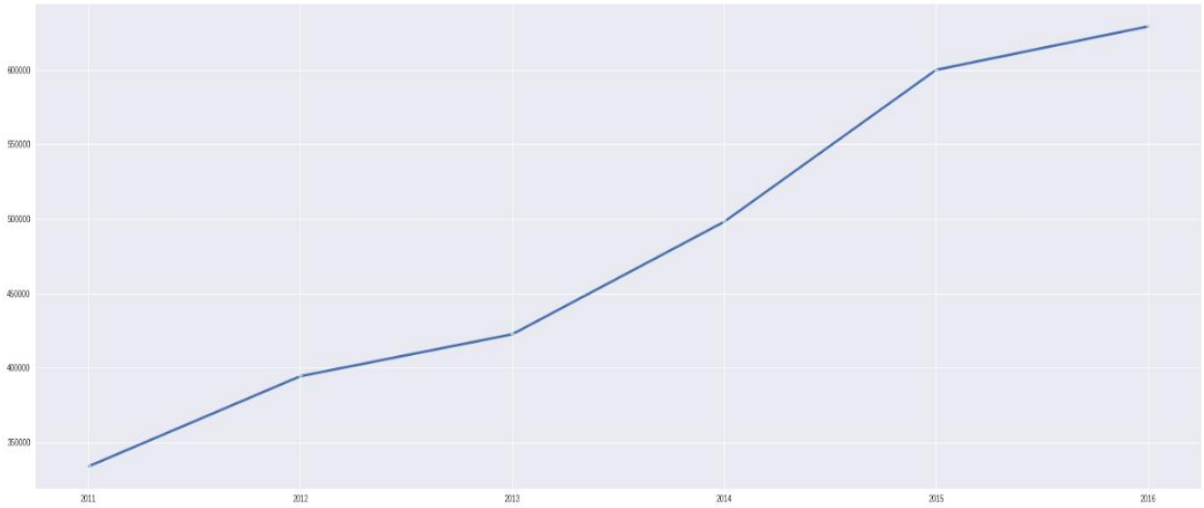
Flowchart:

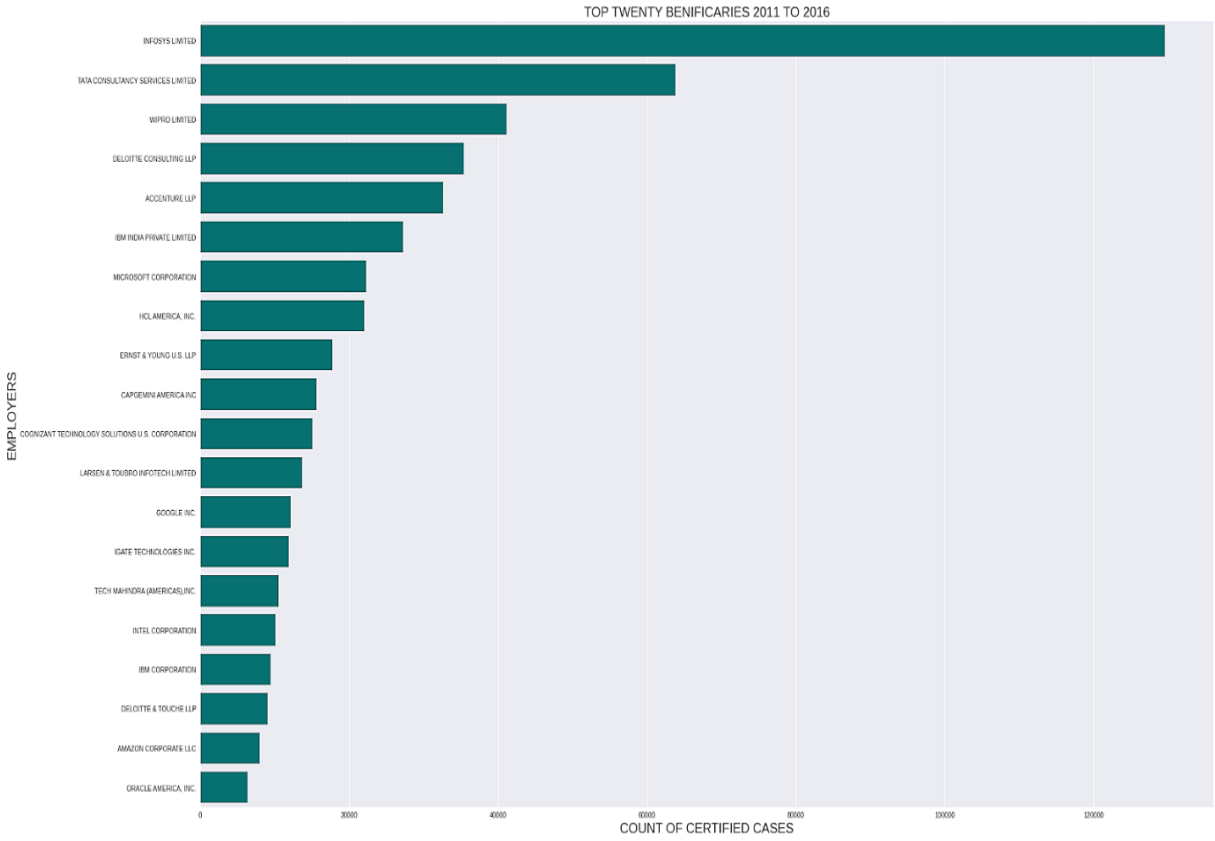


Experimental Investigations

- Out of the seven fields, certified visa is the most prevalent.

- The majority of visas granted are for full-time employment.
- During 2011 to 2016, we see a progressive rise in the number of accepted H1B applications.
- Infosys one of the biggest proponents of H1.
- Visa is an endorsement placed within a passport that grants the holder official permission to enter, leave or stay in a country for a specified time period.

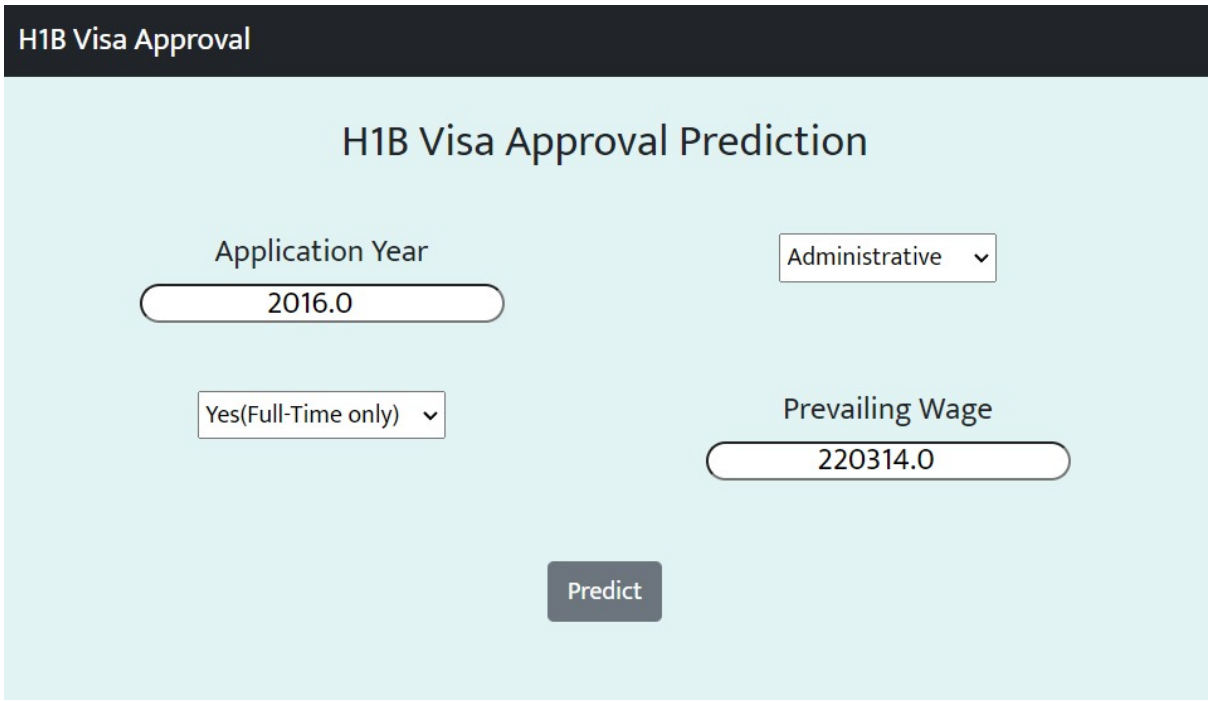




Result

A random forest model is used to predict visa status with 86 percent accuracy when we input information such as occupational code, position, prevailing wage, and year of application. The jupyter file Notebook is then integrated with the Watson Studio and then a flask model is made.

After the deployment we integrate it with the Flask and then we see the outcomes as follows in the screenshots.



File Structure

This project has 4 directories, namely:

1. **Dataset**: This folder contains the dataset that is used to create our Random-Forest model.
2. **Flask-App**: This folder contains the flask code that is required for deploying our Machine Learning model.
   The **app.py** file contains the main Flask code, which is run to start our server, which in-turn starts our Flask model. We have 2 routes in our Flask app, namely the home route and the prediction route
   The **home** route returns the rendered version of our index.html web page.
   The **predict** route takes in the values entered in the form, passes those values as parameters through our random-forest model, and then returns the rendered version of our result.html web page, including our prediction.
   The folder comprises 2 sub-folders; the **static, and templates**.
   The **static** folder is used to hold all of our styling and functionality code, that includes our **.css files and .js files.** The **templates** folder consists of our HTML pages, which is the skeleton of our flask model. We have 2 web-pages:
   1. **Index.html:** This page is our home page, and it contains the HTML Form from which the user puts the inputs.

   Result.html: This page contains the result of our model deployment.
1. **IBM:** This folder contains 2 files, **app_ibm.py and scoring_points.py**
   The **app_ibm.py** file is similar to the app.py file mentioned above, the only difference is that it doesn't use the model stored in .pkl form. Here, we have deployed our model on the IBM Cloud, and we use our keys and tokens to pass parameters to our deployed model.
   The **scoring_points.py** file is used for getting a response on our deployed model, as to see if we get our prediction when we pass arbitrary values into our model in IBM Cloud.

Training: This folder contains our training Jupyter file, which we used to explore our data and deploy it on IBM Cloud. The visualizations and the inferences we have got in this report are from the training file itself.

Advantages and Disadvantages

Knowing the status of the visa will encourage applicants in compiling documentation if it is to be approved, and in putting more effort to upgrade their application if it is to be denied.

The application's lack of robustness is a key drawback. As of now, our model has preset features like wage, year of application etc. Hence, it is not flexible to change in features,

Applications

The aim is to estimate the outcome of H-1B visa applications, which are filed regularly by a high number of professional foreign people. We presented the problem as a classification task in this example and utilized it to determine the application's case state.

Conclusion

To anticipate the visa status, a model that focuses on Random Forest is developed.

Future Scope

- Extending the dataset's capacity and introducing additional features.
- The user experience (UX) and User Interface (UI) can be upgraded.

Bibliography

https://smartinternz.com/

Appendix

Source Code: Here