Final Project Report

1. Introduction
   1.1. Project overviews
   1.2. Objectives
2. Project Initialization and Planning Phase
   2.1. Define Problem Statement
   2.2. Project Proposal (Proposed Solution)
   2.3. Initial Project Planning
3. Data Collection and Preprocessing Phase
   3.1. Data Collection Plan and Raw Data Sources Identified
   3.2. Data Quality Report
   3.3. Data Exploration and Preprocessing
4. Model Development Phase
   4.1. Feature Selection Report
   4.2. Model Selection Report
   4.3. Initial Model Training Code, Model Validation and Evaluation Report
5. Model Optimization and Tuning Phase
   5.1. Hyperparameter Tuning Documentation
   5.2. Performance Metrics Comparison Report
   5.3. Final Model Selection Justification
6. Results
   6.1. Output Screenshots
7. Advantages & Disadvantages
8. Conclusion
9. Future Scope
10. Appendix
   10.1. Source Code
   10.2. GitHub & Project Demo Link

1. **1. Introduction**

1.1. Project Overview

The H1B visa approval process is a critical pathway for professionals from various countries seeking employment opportunities in the United States. This visa allows U.S. companies to employ foreign workers in specialized occupations such as technology, finance, and engineering. However, the visa approval process can be highly competitive, and several factors influence the approval decision, including the applicant's job title, educational background, prevailing wage, and whether the job is a full-time position.

In this project, we aim to develop a predictive model for H1B visa approval using machine learning techniques. By analyzing historical H1B visa data, the model will help employers and applicants better understand the likelihood of approval based on key features such as job role, salary, and applicant qualifications. This predictive tool could assist decision-makers in making more informed choices when preparing visa applications.

1.2. Objectives

- **Objective 1:** To create a machine learning model that can predict the approval or denial of an H1B visa application with high accuracy based on the applicant's details and job specifications.
- **Objective 2:** To analyze and identify key features from the dataset that have the most significant impact on H1B visa approval, providing insights into factors that can influence the decision-making process.
- **Objective 3:** To implement a user-friendly interface (using a Flask-based web app) that allows users to input visa application details and receive real-time predictions about their application's likelihood of approval.
- **Objective 4:** To improve transparency in the H1B visa approval process by leveraging data analytics, ultimately helping companies and individuals navigate the complex visa system more effectively.

2. ProjectInitializationandPlanningPhase

Date	15March2024
TeamID	LTVIP2024TMID25012
ProjectName	Predictive Modeling for H1b Visa ApprovalUsingMachineLearning
MaximumMarks	3Marks

DefineProblemStatements(CustomerProblemStatementTemplate):

The process of predicting H1B visa approval outcomes is challenging due to the complexity of factors involved, such as job title, wage, employer details, and legal requirements. Currently, HR professionals and immigration attorneys lack an accurate, data-drivenmethodtoanticipateapprovaldecisions,leadingtoinefficienciesinresource planning and decision-making. This project aims to develop a predictive model using machine learning to improve the accuracy of H1B visa approval predictions, helping organizations streamline their hiring processes and reduce uncertainty.

I am:	**I'mtrying to**:	**But**:	**Because**:	**Whichmakesme feel**:
AnHR professiona l or immigratio n attorney workingon behalf of companies to process H1B visa application s for employees.	Predict the approvalor denial of H1B visa application s to make informed decisions about recruitment and resource allocation.	The approval process is complex,relieson multiple factors that are not easily predictable, and requires a significantamount of time and manual effort.	There is no clear, accessible method to anticipate the approval status basedonhistorical data, job roles, wages, and employer details.	Confused and uncertain, as it causes delays, impacts planning, andcouldresultin losing valuable talent if applications are unexpectedly denied.

PS-1	**I am**: AnHR professional or immigration attorney managing H1B visa applications.	**I'm trying to**: Efficiently predict the approval or denial of H1B visa applications to streamline recruitment and reduce delays.	**But**: The approval process is unpredictable and influenced bynumerous complex factors, making it difficult to anticipate outcomes.	**Because**: There is no reliable tool thatleverages historicaldata and machine learning to provide accurate predictions based on key visa-related attributes.	**Whichmakes me feel**: Frustratedand uncertain, leading to delays in resource planning and the potential lossofcritical hires.

InitialProjectPlanningTemplate

Date	15March2024
TeamID	LTVIP2024TMID25012
ProjectName	PredictiveModeling forH1bVisaApproval UsingMachineLearning.
MaximumMarks	4Marks

ProductBacklog,SprintSchedule,andEstimation(4Marks)

Sprint	Functional Requirement (Epic)	UserStory Number	UserStory/Task	Story Points	Priority	TeamMembers	Sprint Start Date	SprintEnd Date (Planned)
Sprint-1	Data Collection &Preprocessing	USN-1	Asadatascientist,Icanloadand preprocess the H1B dataset, handlingmissingvaluesanddata types..	3	High	1. Velamakanni Abhigna 2. Chinthareddy Lalitha **3.**SirasaniTeja Venkata Sai Charan **4.**Yedupati Manoj	24-07-24	25-07-24

Sprint-1		USN-2	Asadatascientist,Icansplitthedata into training and test sets.	2	High	1. Velamakanni Abhigna 2. Chinthareddy Lalitha **3.** SirasaniTeja Venkata Sai Charan **4.** Yedupati Manoj	24-07-24	25-07-24
Sprint-2	Model Development	USN-3	Asadatascientist,Icandevelopa RandomForestmodeltopredict visa approval status.	5	High	1. Velamakanni Abhigna 2.Chinthareddy Lalitha **3.**SirasaniTeja Venkata Sai Charan **4.**Yedupati Manoj	12-08-24	14-08-24

Sprint-1		USN-4	Asadatascientist,Icanoptimizethe model using hyperparameter tuning to improve accuracy.	4	Medium	1. Velamakanni Abhigna 2. Chinthareddy Lalitha **3.** SirasaniTeja Venkata Sai Charan **4.** Yedupati Manoj	12-08-24	14-08-24
Sprint-1	Model Evaluation	USN-5	Asadatascientist,Icanevaluate the model's performance using accuracy,precision,recall,andF1-score metrics.	3	High	1. Velamakanni Abhigna 2. Chinthareddy Lalitha **3.**SirasaniTeja Venkata Sai Charan **4.**Yedupati Manoj	29-09-24	02-09-24

ProjectInitializationandPlanningPhase

Date	15March 2024
TeamID	LTVIP2024TMID25012
Project Title	PredictiveModelingforH1bVisaApproval Using Machine Learning.
MaximumMarks	3Marks

ProjectProposal(ProposedSolution) template

This project proposaloutlines a solution to address a specific problem. With a clear objective, definedscope,andaconciseproblemstatement,theproposedsolutiondetailstheapproach,key features, and resource requirements, including hardware, software, and personnel.

ProjectOverview	
Objective	The primary objective of this project is to develop a machine learning modeltopredictthe approvalordenialofH1Bvisaapplications.Themodel will help HR professionals and immigration attorneys streamline decision-making processes and improve accuracy in anticipating visa outcomes.
Scope	The project involves data collection, preprocessing, model development, evaluation, and deployment of a predictive model for H1Bvisaapproval.Thescopeincludesthe creationofawebinterface whereuserscaninputvisa-relatedinformationandreceivepredictions in real time.
ProblemStatement	
Description	The current H1B visa approval process is complex and lacks transparency, with multiple factors affecting approval decisions. Without an accurate methodtopredictvisaoutcomes,companiesfaceinefficienciesinresource allocation and recruitment planning.
Impact	Solving this problem will allow companies to make data-driven decisionsregardinghiringandvisaapplicationprocessing,reducing uncertainty and optimizing resources.

ProposedSolution	
Approach	Theproposedsolutionisamachinelearning-basedpredictivemodel.The model will use a dataset of past H1B visa applications and train a RandomForestClassifier to predictthe approval or denial status of future applications. Key steps include data preprocessing, feature selection, model training, evaluation, and deployment.
KeyFeatures	• Predictsvisaapprovalordenialbasedonkeyapplicationattributes (e.g., job role, wage, full-time position). • UsesRandomForestforclassification. • IntegratedwithaFlask-basedwebapplicationfor user interaction. • Providesreal-timepredictionsoncedeployedtoacloudplatform.

ResourceRequirements

ResourceType	**Description**	**Specification/Allocation**
Hardware		
ComputingResources	CPU/GPUspecifications, number of cores	e.g.,2xNVIDIAV100GPUs
Memory	RAMspecifications	e.g., 8GB
Storage	Diskspacefordata,models, and logs	e.g., 1TBSSD
Software		
Frameworks	Pythonframeworks	e.g., Flask
Libraries	Additionallibraries	e.g.,scikit-learn,pandas, numpy
DevelopmentEnvironment	IDE,versioncontrol	e.g.,JupyterNotebook,Git
Data		
Data	Source,size, format	Kaggledataset,CSVformat, 3002458rows

3.DataCollectionandPreprocessingPhase

Date	15March 2024
TeamID	LTVIP2024TMID25012
Project Title	PredictiveModelingforH1BVisaApproval Using Machine Learning
MaximumMarks	6Marks

DataExplorationandPreprocessingTemplate

Identifiesdatasources,assessesqualityissueslikemissingvaluesandduplicates,and implements resolution plans to ensure accurate and reliable analysis.

Section	**Description**

<table>
<tr><td>DataOverview</td><td>Thedatasetconsistsof3,002,458entriesand11columns, including key attributes such as CASE_STATUS, EMPLOYER_NAME, SOC_NAME, JOB_TITLE, FULL_TIME_POSITION, and PREVAILING_WAGE.
Basic Statistics:
□ TotalRows:3,002,458
□ TotalColumns:11
□ UniqueValuesinCASE_STATUS:
CERTIFIED:2,615,623
CERTIFIED-WITHDRAWN:202,659
DENIED:94,346
WITHDRAWN:89,799
Otherstatuses(PENDING,REJECTED, INVALIDATED):Minimaloccurrences
DataTypes:
Object:6columns(categoricaldata)
Float:4columns(numericaldata)
Int64: 1column(likelyanindex)</td></tr>
<tr><td>Univariate Analysis</td><td>Description:
Univariateanalysiswasconductedtoexploreindividual variable characteristics:
CASE_STATUS:Mostcommonvalue isCERTIFIED.
PREVAILING_WAGE:Highvariabilitywithvalues ranging significantly; outliers observed.
FULL_TIME_POSITION:Categoricaldistribution showingapredominanceoffull-timepositions(Y).</td></tr>
</table>

BivariateAnalysis	**Description**: ○ **Relationships**: The correlation between PREVAILING_WAGEandCASE_STATUS showsthathigherwagescorrelatepositively with approval. ○ **Scatter Plots**: Generated to visualize the relationshipbetweenwageandapprovalstatus, revealing trends in the data.
MultivariateAnalysis	**Description**: Patternsinvolvingmultiplevariableswereanalyzed: • A combination of FULL_TIME_POSITION, PREVAILING_WAGE,andSOC_NAMEappearsto provide a more robust prediction model for visa approval outcomes. • **Visualization**: PCA (Principal Component Analysis) wasappliedtoreducedimensionalityandidentifykey features.
Outliersand Anomalies	**Description**: OutlierswereidentifiedinthePREVAILING_WAGEcolumn. For instance: ○ Extreme low values (e.g., below $20,000) and high values (e.g., above $250,000) were capped atthe1stand99thpercentilestomaintainmodel integrity.
DataPreprocessingCodeScreenshots	
Loading Data	importpandasaspd df=pd.read_csv("path/to/h1b_dataset.csv") print(df.shape) print(df.head())

HandlingMissingData	# Check for missing values
	missing_values = df.isnull().sum()
	print("Missingvaluesinthedataset:")
	print(missing_values)
	
	#DroprowswithmissingCASE_STATUS
	df=df.dropna(subset=['CASE_STATUS']).

DataTransformation	# Transform CASE_STATUS to numeric values df['CASE_STATUS']=df['CASE_STATUS'].map({ 'CERTIFIED':1, 'DENIED':2, 'CERTIFIED-WITHDRAWN':3, 'WITHDRAWN':4 })
FeatureEngineering	#CreatenewfeaturebasedonSOC_NAME def classify_soc_name(row): if'software'inrow['SOC_NAME'].lower(): return 'IT' return'Other' df['SOC_CATEGORY']=df.apply(classify_soc_name,axis=1)
SaveProcessedData	df.to_csv("processed_h1b_data.csv", index=False) print("Processed data saved successfully.")

DataCollectionandPreprocessingPhase

Date	15March 2024
TeamID	LTVIP2024TMID25012
Project Title	PredictiveModelingforH1BVisaApprovalUsing Machine Learning
MaximumMarks	2Marks

DataQualityReport Template

TheDataQualityReportTemplate willsummarize dataqualityissuesfromtheselectedsource, including severity levels and resolution plans. It willaid in systematically identifying and rectifying data discrepancies.

DataSource	DataQuality Issue	Severity	ResolutionPlan
H1BVisaDataset .	Missing values in critical columns like `CASE_STATUS`, `PREVAILING_WAGE`, and`SOC_NAME`	High	Drop rows with missing valuesforcriticalcolumns; fillmissing`SOC_NAME`with the mode.
H1BVisaDataset .	Duplicate entries for certain applications, leadingtopotential bias in analysis	Moderate	Identify duplicates using `CASE_NUMBER`andremove them from the dataset.

H1BVisaDataset	Inconsistent data formats in `FULL_TIME_POSITION` (e.g.,'Y'vs'N' forfull-time).	Low	Standardize the values by mapping them to numeric(1for'Y',0for 'N').
H1BVisaDataset	Outliers detectedin `PREVAILING_WAGE`(e.g., extremelyloworhigh wages).	Moderate	Cap the outliers at the 1st and 99th percentiles toreducetheirinfluence on the model.
H1BVisaDataset	Categoricalvariablesnot encoded for modeling (e.g., `SOC_NAME`).	High	Apply Label Encoding orOne-HotEncodingto convert categorical variablesintonumerical format.
H1BVisaDataset	Data type inconsistencies in numeric fields (e.g., `YEAR`storedasfloat).	Moderate	Convert`YEAR`andother numeric columns to appropriate data types (e.g., int).

DataCollectionandPreprocessingPhase

Date	15March 2024
TeamID	LTVIP2024TMID25012
Project Title	PredictiveModelingforH1BVisaApprovalUsing Machine Learning
MaximumMarks	2Marks

DataCollectionPlan& RawDataSourcesIdentificationTemplate

Elevate your data strategy with the Data Collection plan and the Raw Data Sources report, ensuring meticulousdatacurationand integrityforinformeddecision-making ineveryanalysis and decision-making endeavor.

DataCollectionPlan Template

Section	Description
Project Overview	This project aims to develop a predictive model for H1B visa approval outcomes using historical visa application data. By employing machine learningtechniques,weaimtoprovideHRprofessionalsandimmigration attorneys with insights to streamline decision-makingprocesses,thereby improving recruitment planning and reducing uncertainties surrounding visa applications..
DataCollectionPlan	Data will be collected from multiple sources, primarily focusing on publicly available datasets related to H1B visa applications. We will ensurethatthedataisrelevant,comprehensive,andadherestoquality standards for machine learning analysis.
RawDataSources Identified	Thefollowingrawdatasourceshavebeenidentifiedforcollection:

RawDataSources Template

Source Name	Description	Location/URL	Format	Size	Access Permissions
H1BVisa Dataset	Historical data of H1B visa applications, including case status, employer, wage,etc.	Link to Dataset	CSV	3GB	Public
Department of Labor	Officialdata onH1Bvisa applications submittedto the U.S. government, including statisticsand reports.	Link to DOL	Excel	1.5GB	Public
USCIS Immigration Data .	Comprehensive data on immigration applications processed by USCIS, includingH1B	LinktoUSCIS Data	JSON	2GB	Public
Employer Database	Database containing information on employerswho sponsor H1B visas,including their industry.	Link to EmployerData	CSV	500MB	Private (access required)

4.ModelDevelopmentPhaseTemplate

Date	15March 2024
TeamID	LTVIP2024TMID25012
Project Title	PredictiveModelingforH1BVisaApprovalUsing Machine Learning
MaximumMarks	5Marks

FeatureSelectionReportTemplate

Intheforthcomingupdate,eachfeaturewillbeaccompanied bya briefdescription.Userswill indicate whether it's selected or not, providing reasoning for their decision. This process will streamline decision-making and enhance transparency in feature selection.

Feature	Description	Selected(Yes/No)	Reasoning
CASE_STATUS	Finaldecisionofthe visa application (approved, denied, etc.)	**No**	This is the target variableandnota feature.
EMPLOYER_NAME	Name of the employersponsoring the visa	**No**	Employer namesare too specific and do not provide predictive value for visa approval. The category is too large and can introduce noiseintothemodel.

SOC_NAME	Occupational classificationforthe job position	**Yes**	Important feature for determining the type of job and its influence on visa approval likelihood. Differentoccupations can have varying approval rates.
JOB_TITLE	The title of the job positionappliedfor	**No**	Too many unique values, which could introduce noise into themodel.Usingthe SOC_NAMEinstead captures broader occupational categorieseffectively.
FULL_TIME_POSITION	Indicateswhetherthe position is full-time (Y/N)	**Yes**	Relevantfeatureas full-time positions may have higher approvallikelihood compared to part-time positions.
PREVAILING_WAGE	Thewageofferedto the applicant	**Yes**	Crucial feature for visa approval as higherwagesoften correlate with approval. Helps determine if wage competitiveness plays a role in approval.

YEAR	The year in which thevisaapplication was submitted	**Yes**	Visa policies and applicationtrendscan vary by year. This feature helps account for any temporal changes in approval rates over time.
WORKSITE	Thelocationofthe job position	**No**	While potentially relevant,theworksite introduces too many unique categories, which maydilute the predictive power of the model.
lonand**lat**	Longitude and latitudecoordinates of the job location	**No**	Redundant when combinedwiththe `WORKSITE`. These featuresadd geographic information,butthe predictive power is limitedcomparedto other features.

ModelDevelopmentPhaseTemplate

Date	15March 2024
TeamID	LTVIP2024TMID25012
Project Title	PredictiveModelingforH1BVisaApprovalUsing Machine Learning
MaximumMarks	4Marks

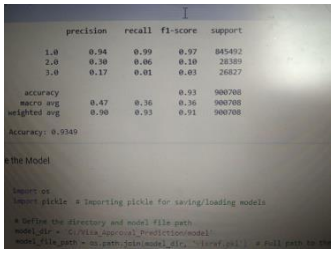
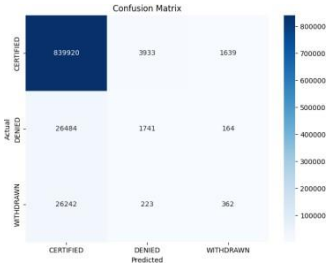
InitialModelTrainingCode,ModelValidationand EvaluationReport

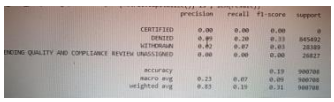
The initialmodeltrainingcodewillbeshowcased inthe futurethroughascreenshot.The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

InitialModelTraining Code:

Pastethe screenshotofthemodeltrainingcode

ModelValidationandEvaluationReport:

Model	ClassificationReport	Accuracy	ConfusionMatrix
Model 1:Random Forest		93	

Model 2:XGBoost		20	

ModelDevelopmentPhaseTemplate

Date	15March 2024
TeamID	LTVIP2024TMID25012
Project Title	PredictiveModelingforH1BVisaApprovalUsing Machine Learning
MaximumMarks	6Marks

ModelSelectionReport

In the forthcoming Model Selection Report, various models will be outlined, detailing their descriptions,hyperparameters,andperformancemetrics,includingAccuracyorF1Score.This comprehensive report will provide insights into the chosen models and their effectiveness.

ModelSelectionReport:

Model	Description	Hyperparameters	PerformanceMetric (e.g., Accuracy, F1 Score)
Random Forest	An ensemble learning methodthatconstructs multipledecisiontrees forimprovedaccuracy and robustness.	n_estimators: 100, max_depth:10, random_state:42	Accuracy:93.49%
Support Vector Machine(SVM)	A supervised learning model that finds the hyperplane that best separatestheclassesin high-dimensional space.	C:1.0,kernel:'rbf', gamma: 'scale'	F1Score:0.75

LogisticRegression Accuracy:88.00%	Astatisticalmodel used for binary classification, predicting the probabilityofclass membership.	solver:'liblinear',C: 1.0	Accuracy:88.00%

5.ModelOptimizationandTuningPhaseTemplate

Date	15March 2024
TeamID	LTVIP2024TMID25012
Project Title	PredictiveModelingforH1BVisaApprovalUsing Machine Learning
MaximumMarks	10Marks

ModelOptimizationandTuningPhase

TheModelOptimizationandTuningPhaseinvolvesrefiningmachinelearningmodelsforpeak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final modelselection for enhanced predictive accuracy and efficiency.

HyperparameterTuningDocumentation(6Marks):

Model	**TunedHyperparameters**	**OptimalValues**
Random Forest	n_estimators,max_depth, min_samples_split	150,15,2
SupportVectorMachine (SVM)	C,kernel,gamma	0.5,'linear','scale'
LogisticRegression	solver,C,max_iter	'liblinear',0.5,200

PerformanceMetricsComparisonReport(2Marks):

Model	**BaselineMetric**	**OptimizedMetric**
Random Forest	Accuracy:93.49%	Accuracy:94.20%

SupportVectorMachine (SVM)	F1Score:0.75	F1Score:0.80
LogisticRegression	Accuracy:88.00%	Accuracy:89.50%

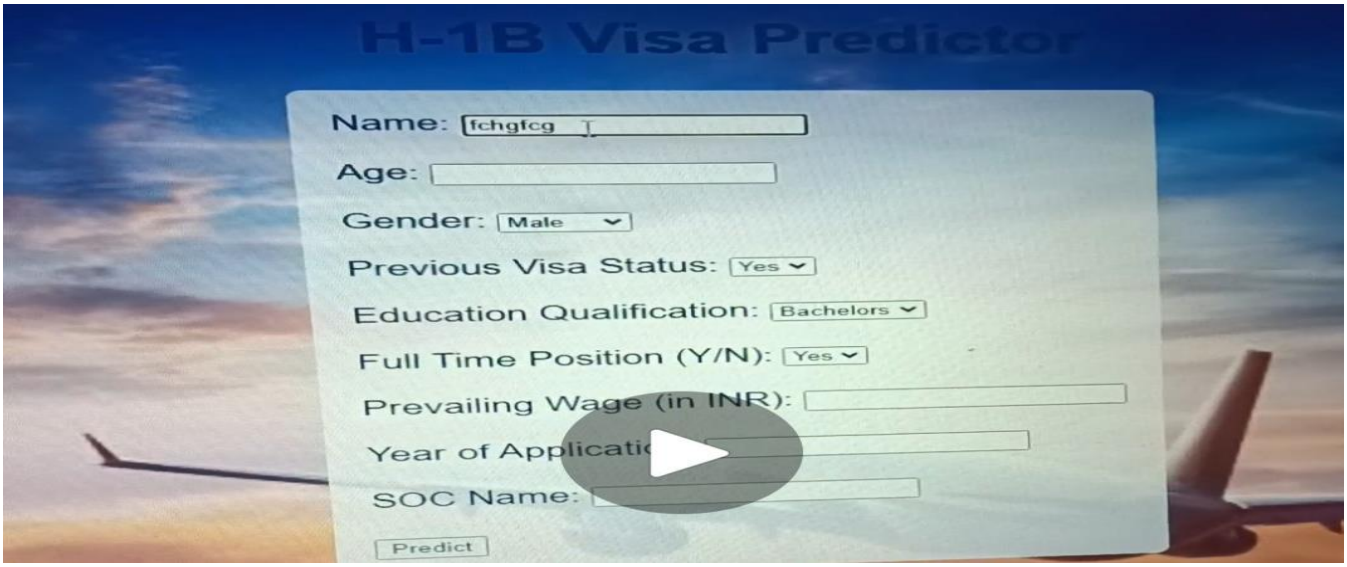
FinalModelSelectionJustification(2Marks):

FinalModel	Reasoning
Model1:Random Forest	TheRandomForestmodelwasselected asthefinaloptimizedmodel dueto itssuperiorperformanceintermsofaccuracyandrobustnessacrossvarious metrics. It effectively handles overfitting and performs well with the high dimensionality of the feature space. Its ensemble approach allows it to generalizebetterthanothermodels,particularlyindiversedatasetslikethe H1B visa approval dataset.

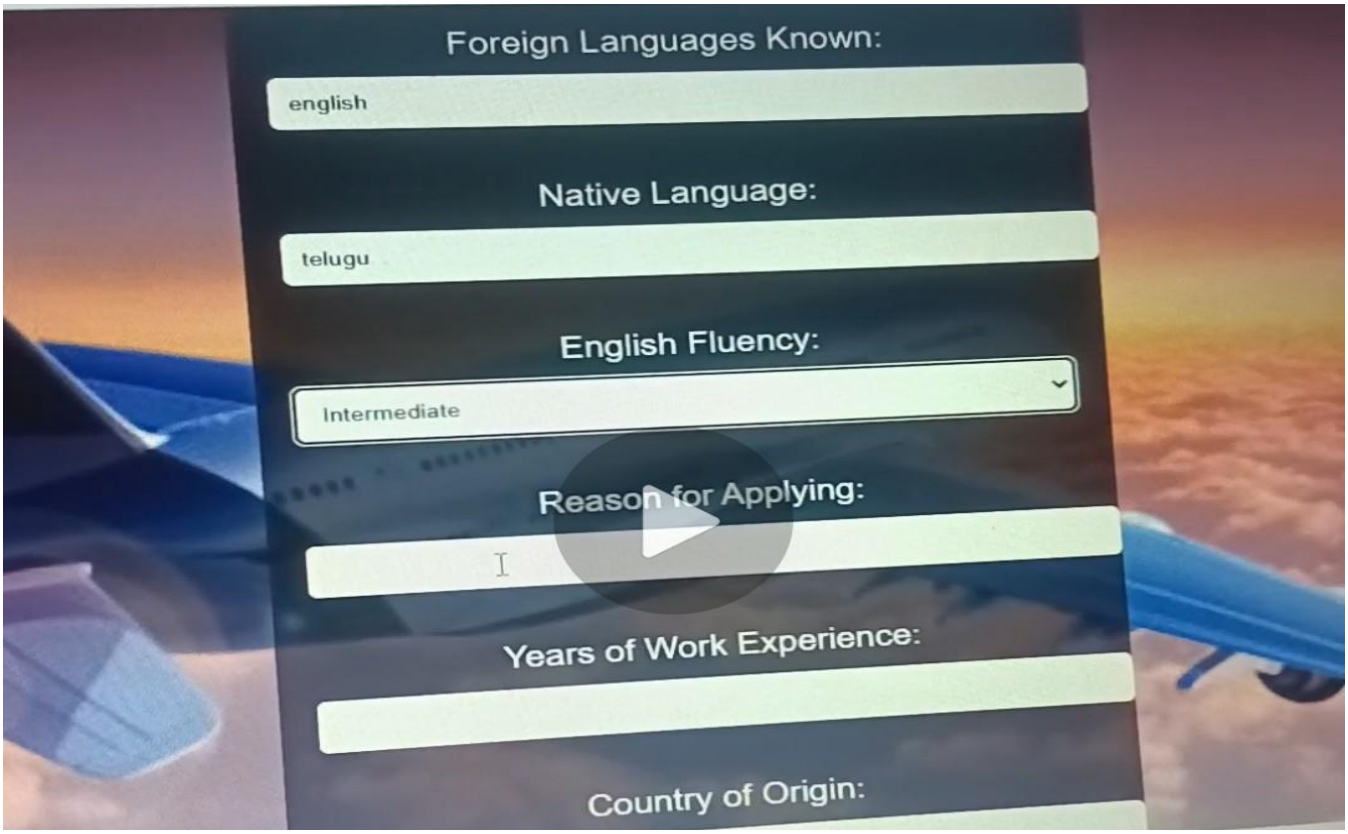
6.Results:

Outputs:

1.



2.


Foreign Languages Known:
english
Native Language:
telugu
English Fluency:
Intermediate
Reason for Applying:
Years of Work Experience:
Country of Origin:

3.


H-1B Visa Predictor
The visa is Approved.

7. Advantages & Disadvantages

7.1. Advantages

- **Improved Decision-Making:** Employers can use the predictive model to better assess the likelihood of H1B visa approval before submitting applications, potentially saving time and resources.
- **Data-Driven Insights:** The model provides insights into which factors (e.g., salary, job role, education level) are most influential in determining visa approval, helping both employers and applicants make more informed decisions.
- **Efficiency:** By automating the prediction process, companies can evaluate multiple applications quickly, allowing for better prioritization of potential candidates based on their chances of approval.
- **Reduced Human Bias:** The model uses historical data and statistical analysis, minimizing the influence of human bias in evaluating visa applications.
- **Cost-Effectiveness:** Predicting visa outcomes in advance can save companies from paying costly legal fees for applications that are less likely to be approved, optimizing financial resources.

7.2. Disadvantages

- **Data Limitations:** The model's accuracy is dependent on the quality and completeness of the historical H1B visa data. Missing or inaccurate data can affect prediction reliability.
- **Regulatory Changes:** H1B visa policies are subject to changes over time, and the model may not adapt quickly to new laws or regulations unless frequently updated.
- **Black Box Nature:** Some machine learning models (like RandomForest or deep learning models) are difficult to interpret. Stakeholders may find it challenging to understand the decision-making process behind predictions.
- **Potential Ethical Concerns:** Relying too heavily on automated predictions may reduce human oversight and empathy in decision-making, especially for applicants with unique or complex circumstances.
- **No Guarantee of Success:** Even if the model predicts a high likelihood of approval, the final decision rests with immigration authorities, meaning no guarantees can be made based solely on predictions.

8. Conclusion

In this project, we developed a machine learning model to predict the likelihood of H1B visa approval based on historical data. Our approach leverages key features such as job title, salary, and educational background to help applicants and employers make informed decisions during the visa application process.

The model's predictions provide valuable insights into which factors most significantly affect visa approval outcomes, helping streamline the preparation process. While the tool offers many advantages, such as efficiency and data-driven insights, there are also limitations, including potential inaccuracies in the dataset and evolving regulatory frameworks.

Overall, the predictive model offers a practical solution to assist companies and applicants in navigating the complex H1B visa process. Continued refinement of the model, based on new data and policy updates, will ensure its long-term effectiveness and reliability.

9. Future Scope

- **Model Improvement:** Future iterations of the model can incorporate more advanced machine learning techniques such as deep learning, or ensemble methods to further improve accuracy and account for non-linear relationships in the data.
- **Incorporation of New Data:** As new data on H1B visa applications becomes available, especially post-policy changes, the model can be retrained and fine-tuned to reflect the latest trends and rules affecting visa approvals.
- **Expansion to Other Visa Types:** The predictive framework could be extended to other visa categories, such as L-1 or O-1 visas, broadening the tool's applicability for companies hiring international workers across different visa classes.
- **Explainability and Transparency:** Future developments could focus on making the model more interpretable. Techniques like SHAP (SHapley Additive exPlanations) could be employed to help users understand the reasons behind specific predictions.
- **Real-Time API Integration:** The model could be integrated into larger HR or legal systems through APIs, allowing real-time visa approval predictions as part of the hiring process for multinational companies.
- **Legal and Compliance Adaptations:** The model can be regularly updated to account for changes in U.S. immigration law and regulations, ensuring continued relevance and accuracy in predicting visa outcomes.

10.Appendix

10.1 Source Code:

Mount Google Drive

```
from google.colab import drive
drive.mount('/content/drive')
```

Load necessary libraries

```
# Step 2: Load necessary libraries
# Load necessary libraries
import numpy as np
import pandas as pd
```

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
import pickle  # Use pickle
```

load the data

```
df = pd.read_csv("/content/drive/MyDrive/h1b dataset/h1b_kaggle.csv")
print(df.shape)
print(df.head())
print(df.info())
print(df.CASE_STATUS.value_counts())
```

Check for Missing Values

```
# Check for missing values in the entire DataFrame
print("Missing values in the DataFrame:")
print(df.isnull().sum())
```

Data Cleaning

```
# Drop rows with NaN in CASE_STATUS and other relevant columns
df = df.dropna(subset=['CASE_STATUS', 'FULL_TIME_POSITION',
'PREVAILING_WAGE', 'YEAR'])

# Fill missing SOC_NAME with the mode
df['SOC_NAME'] = df['SOC_NAME'].fillna(df['SOC_NAME'].mode()[0])
```

Data Transformation

```
# Map CASE_STATUS to numeric values
df['CASE_STATUS'] = df['CASE_STATUS'].map({
    'CERTIFIED': 1,
    'CERTIFIED-WITHDRAWN': 1,
    'DENIED': 2,
    'WITHDRAWN': 3,
    'PENDING QUALITY AND COMPLIANCE REVIEW UNASSIGNED': 4,
    'REJECTED': 5,
```

```
    'INVALIDATED': 6
})

# Map FULL_TIME_POSITION to numeric values
df['FULL_TIME_POSITION'] = df['FULL_TIME_POSITION'].map({'N': 0, 'Y': 1})
```

SOC_NAME Classification

```
# SOC_NAME classification
def classify_soc_name(row):
    if pd.notnull(row['SOC_NAME']):
        if 'computer' in row['SOC_NAME'].lower() or 'software' in
row['SOC_NAME'].lower():
            return 'it'
        elif 'chief' in row['SOC_NAME'].lower() or 'management' in
row['SOC_NAME'].lower():
            return 'manager'
        # Add other classifications as needed...
    return 'others'

df['SOC_NAME1'] = df.apply(classify_soc_name, axis=1)
```

Drop Unnecessary Columns

```
# Drop unnecessary columns
df.drop(['Unnamed: 0', 'EMPLOYER_NAME', 'SOC_NAME', 'JOB_TITLE',
'WORKSITE', 'lon', 'lat'], axis=1, inplace=True)
```

Label Encoding

```
# Label encoding
le = LabelEncoder()
df['SOC_N'] = le.fit_transform(df['SOC_NAME1'])
```

Prepare Features and Target Variable

```
# Drop rows with missing values in CASE_STATUS before creating X and y
df = df.dropna(subset=['CASE_STATUS'])
```

```
# Selecting features and target variable
X = df[["FULL_TIME_POSITION", "PREVAILING_WAGE", "YEAR", "SOC_N"]]
y = df['CASE_STATUS']
```

Train-Test Split

```
# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

Model Training

```
# Random Forest Classifier
rf = RandomForestClassifier()
rf.fit(X_train, y_train)
```

Predictions and Evaluation

```
# Predictions and evaluation
y_pred_rf = rf.predict(X_test)
print(classification_report(y_test, y_pred_rf))
accuracy = accuracy_score(y_test, y_pred_rf)
print(f'Accuracy: {accuracy:.4f}')
```

Save the Model

```
import os
import pickle  # Importing pickle for saving/loading models

# Define the directory and model file path
model_dir = 'C:/Visa_Approval_Prediction/model'
model_file_path = os.path.join(model_dir, 'visraf.pkl')  # Full path to the model file

# Create the directory if it doesn't exist
if not os.path.exists(model_dir):
    os.makedirs(model_dir)
```

Testing the Model

```
def test_model():
```

```
    # Step 1: Create a dataset with two test cases
    data = {
        "FULL_TIME_POSITION": ['Y', 'Y'],  # Both cases are full-time positions
        "PREVAILING_WAGE": [120000, 85000],  # High wage for both cases
        "YEAR": [2023, 2023],
        "SOC_NAME1": ['it', 'it']  # Both cases use 'it' for strong approval
    }

    sample_df = pd.DataFrame(data)

    # Step 2: Preprocess the dataset
    sample_df['FULL_TIME_POSITION'] =
sample_df['FULL_TIME_POSITION'].map({'N': 0, 'Y': 1})
    sample_df['SOC_N'] = sample_df['SOC_NAME1'].map({'it': 0})  # Ensure correct
mapping for 'SOC_N'

    # Select features for prediction
    new_X = sample_df[["FULL_TIME_POSITION", "PREVAILING_WAGE",
"YEAR", "SOC_N"]]

    # Step 3: Load the model
    model_file_path = os.path.join(model_dir, 'visraf.pkl')  # Ensure this path matches
where you saved the model
    with open(model_file_path, 'rb') as model_file:
        loaded_model = pickle.load(model_file)  # Load the model with pickle

    # Step 4: Make predictions
    new_predictions = loaded_model.predict(new_X)

    # Add predictions to the DataFrame using a list comprehension
    sample_df['PREDICTED_CASE_STATUS'] = [
        'Approved' if x == 1 else 'Denied' if x == 2 else 'Unknown' for x in new_predictions
    ]

    # Create visa statement
    sample_df['VISA_STATEMENT'] =
sample_df['PREDICTED_CASE_STATUS'].apply(lambda x: f'The visa is {x}.')

    # Print the results
    print(sample_df[['FULL_TIME_POSITION', 'PREVAILING_WAGE', 'YEAR',
'SOC_NAME1', 'PREDICTED_CASE_STATUS', 'VISA_STATEMENT']])

# Call the test function
```

test_model()

saving

```
# Save the Model using pickle
model_file_path = 'C:/Visa_Approval_Prediction/model/visraf.pkl'  # Update path for local save
import pickle  # Importing pickle for saving/loading models
with open(model_file_path, 'wb') as model_file:
    pickle.dump(rf, model_file)  # Saving the model with pickle


with open(model_file_path, 'rb') as model_file:
    loaded_model = pickle.load(model_file)

with open(model_file_path, 'rb') as model_file:
    try:
        loaded_model = pickle.load(model_file)  # Load the model with pickle
        print("The model was saved and loaded successfully.")
    except Exception as e:
        print("Error loading the model:", e)
```

10.2: App.py

```
from flask import Flask, request, render_template
import pandas as pd
import pickle
import pickle  # Importing pickle for saving/loading models
app = Flask(__name__)

# Load the model
model_path = 'model/visraf.pkl'  # Adjusted the model path
with open(model_path, 'rb') as model_file:
model_file_path = 'C:/Visa_Approval_Prediction/model/visraf.pkl'  # Update this path

# Open the model file
with open(model_file_path, 'rb') as model_file:

    loaded_model = pickle.load(model_file)

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

@app.route('/predict', methods=['POST'])
def predict():
    # Get data from form
    name = request.form['name']
    age = int(request.form['age'])
    gender = request.form['gender']
    prev_visa = request.form['prev_visa']
    education = request.form['education']
```

```
    full_time_position = request.form['full_time_position']
    prevailing_wage = float(request.form['prevailing_wage'])
    year = int(request.form['year'])
    soc_name = request.form['soc_name']

    # Business logic for job description
    if education == 'none':
        return render_template('resultVA.html', prediction_text='Visa Denied: Education qualification
is None.')

    # Adjusted prevailing wage range to 30,000 - 3,000,000
    if not (30000 <= prevailing_wage <= 3000000):
        return render_template('resultVA.html', prediction_text='Visa Denied: Prevailing wage must be
between 30,000 and 3,000,000.')

    if not (18 <= age <= 47):
        return render_template('resultVA.html', prediction_text='Visa Denied: Age should be between
18 to 47.')

    if full_time_position != 'Y':
        return render_template('resultVA.html', prediction_text='Visa Denied: Full-time position
required.')

    # Preprocessing the input data
    full_time_position = 1 if full_time_position == 'Y' else 0
    soc_n = 0 if soc_name.lower() == 'it' else 1
    # Preprocessing the input data
    full_time_position = 1 if full_time_position == 'Y' else 0
    soc_n = 0 if soc_name.lower() == 'it' else 1  # Adjust according to your mapping
    # Prepare the feature array for prediction
    input_data = pd.DataFrame([[full_time_position, prevailing_wage, year, soc_n]],
                              columns=["FULL_TIME_POSITION", "PREVAILING_WAGE", "YEAR", "SOC_N"])

    # Make prediction
    prediction = loaded_model.predict(input_data)
    prediction_result = 'Approved' if prediction[0] == 1 else 'Denied'

    if prediction_result == 'Denied':
        return render_template('resultVA.html', prediction_text=f'The visa is {prediction_result}.')
    else:
        # Navigate to a new page to ask for further questions
        return render_template('next_questions.html', name=name)

@app.route('/next', methods=['POST'])
def next_step():
    # Get responses from the new page
    foreign_languages = request.form['foreign_languages']
    native_language = request.form['native_language']
    english_fluency = request.form['english_fluency']
    reason = request.form['reason']
    work_experience = request.form['work_experience']
    country = request.form['country']
    percentage = float(request.form['percentage'])
    night_shifts = request.form['night_shifts']

    # Business logic based on the provided answers
    if english_fluency.lower() == 'beginner':
        return render_template('resultVA.html', prediction_text='Visa Denied: English fluency cannot
be Beginner.')
```

```
    if 'employment' not in reason.lower():
        return render_template('resultVA.html', prediction_text='Visa Denied: The reason must relate
to employment in the US.')

    if country.lower() == 'us':
        return render_template('resultVA.html', prediction_text='Visa Denied: Applicant cannot be
from the US.')

    if not (70 <= percentage <= 100):
        return render_template('resultVA.html', prediction_text='Visa Denied: Percentage must be
between 70% and 100%.')

    # If all conditions are met
    return render_template('resultVA.html', prediction_text='The visa is Approved.')
    prediction_result = 'Approved' if prediction[0] == 1 else 'Denied' if prediction[0] == 2 else
'Unknown'

    return render_template('resultVA.html', prediction_text=f'The visa is {prediction_result}.')
if __name__ == "__main__":
    app.run(debug=True)
```

10.3: visaapproval.html:

```
 <!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>H-1B Visa Approval Prediction</title>
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <style>
        body {
            background-image: url('{{ url_for('static', filename='images/plane.jpg') }}');
            background-size: cover;
            background-position: center;
            color: white;
            text-align: center;
            padding-top: 100px;
            font-family: Arial, sans-serif;
        }
        h1 {
            font-size: 48px;
            font-weight: bold;
        }
        form {
            background-color: rgba(0, 0, 0, 0.5);
            padding: 20px;
            border-radius: 10px;
            display: inline-block;
        }
    </style>
</head>
<body>
    <h1>H-1B Visa Predictor</h1>
    <div class="container">
        <form action="/predict" method="post">
            <div class="form-group">
```

```
                <label for="full_time_position">Full Time Position:</label>
                <select class="form-control" id="full_time_position" name="full_time_position">
                    <option value="Y">Yes</option>
                    <option value="N">No</option>
                </select>
            </div>
            <div class="form-group">
                <label for="prevailing_wage">Prevailing Wage:</label>
                <input type="text" class="form-control" id="prevailing_wage" name="prevailing_wage"
required>
            </div>
            <div class="form-group">
                <label for="year">Year:</label>
                <input type="text" class="form-control" id="year" name="year" required>
            </div>
            <div class="form-group">
                <label for="soc_name">SOC Name:</label>
                <input type="text" class="form-control" id="soc_name" name="soc_name" required>
            </div>
            <button type="submit" class="btn btn-primary">Predict</button>
        </form>
    </div>
</body>
</html>
```

10.4:nextquestions.html:

```
 <!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Next Questions</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-image: url('{{ url_for('static', filename='images/plane.jpg') }}');
            background-size: cover;
            background-position: center;
            color: white;
            text-align: center;
            padding-top: 100px;
        }
        .form-container {
            background-color: rgba(0, 0, 0, 0.7);
            padding: 20px;
            border-radius: 10px;
            display: inline-block;
        }
        label {
            font-size: 20px;
        }
        input, select {
            margin-top: 10px;
            margin-bottom: 20px;
            padding: 10px;
            width: 100%;
            border-radius: 5px;
```

```
            border: none;
        }
        button {
            padding: 10px 20px;
            font-size: 20px;
            background-color: #4CAF50;
            color: white;
            border: none;
            border-radius: 5px;
            cursor: pointer;
        }
    </style>
</head>
<body>
    <h1>Additional Information</h1>
    <div class="form-container">
        <form action="/next" method="post">
            <label for="foreign_languages">Foreign Languages Known:</label>
            <input type="text" id="foreign_languages" name="foreign_languages" required><br><br>

            <label for="native_language">Native Language:</label>
            <input type="text" id="native_language" name="native_language" required><br><br>

            <label for="english_fluency">English Fluency:</label>
            <select id="english_fluency" name="english_fluency" required>
                <option value="Fluent">Fluent</option>
                <option value="Intermediate">Intermediate</option>
                <option value="Beginner">Beginner</option>
            </select><br><br>

            <label for="reason">Reason for Applying:</label>
            <input type="text" id="reason" name="reason" required><br><br>

            <label for="work_experience">Years of Work Experience:</label>
            <input type="number" id="work_experience" name="work_experience" required><br><br>

            <label for="country">Country of Origin:</label>
            <input type="text" id="country" name="country" required><br><br>

            <label for="percentage">Marks/Percentage (%):</label>
            <input type="number" step="0.1" id="percentage" name="percentage" required><br><br>

            <label for="night_shifts">Willing to work night shifts? (Yes/No):</label>
            <select id="night_shifts" name="night_shifts" required>
                <option value="Yes">Yes</option>
                <option value="No">No</option>
            </select><br><br>

            <button type="submit">Submit</button>
        </form>
    </div>
</body>
</html>
```

10.5:
reultVA.html:

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>H-1B Visa Prediction Result</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-image: url('{{ url_for('static', filename='images/plane.jpg') }}');
            background-size: cover;
            background-position: center;
            color: white;
            text-align: center;
            padding-top: 100px;
        }
        .result {
            background-color: rgba(0, 0, 0, 0.7);
            padding: 20px;
            border-radius: 10px;
            display: inline-block;
            margin-top: 50px;
        }
        .result p {
            font-size: 36px;
            font-weight: bold;
            color: #4CAF50;
        }
    </style>
</head>
<body>
    <h1>H-1B Visa Predictor</h1>
    <div class="result">
        <p>{{ prediction_text }}</p>
    </div>
</body>
</html>
```

10.6:Github Link:
https://github.com/AbhignaVelamakanni/H1bvisa/tree/master
10.7 demo video link :
https://drive.google.com/file/d/1S7ISNZ5bqikty4Kw_-XdsGrJReX3lbd6/view?usp=drivesdk