H-1B Visa Labor Condition Application Approval Prediction

A team project report submitted for

BIA 678-C - Big Data Technologies

by

Akshat Jain (CWID 2000 8931)
Jaykumar Patel (CWID 2000 8512)
Juilee Thakur (CWID 2000 8598)
Prathamesh Desai (CWID 1047 4461)

(Group No.: CD - 10)

Under the guidance of
Prof. David Belanger

STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®

Stevens Institute of Technology
Hoboken, New Jersey
December 09, 2022

Table of Contents:

1. Introduction

Labor Conditional Application, or LCA, is a required document that before filing the H-1B application with USCIS for whichever non-immigrant worker, Employers must apply with the US Department of Labor. LCA is essential to ensuring that you are paid a fair wage as a foreign employee and are not exploited by US businesses. Among the first stages to obtaining an H1B work permit in the US is having an LCA approved.

An H1B Labor Condition Application (LCA) form contains all the pertinent details about the position being offered to the foreign worker, including the wage and location information. To submit an H1B LCA, the H1B Sponsor must use the US DOL's online system known as "Foreign Labor Certification Gateway (FLAG)" to submit ETA Form 9035 / 9035E electronically. Employers sponsoring additional visa classes, such as "H-1--B1," which are used by citizens of Chile and Singapore, and "E--3," which are used by Australians, also utilize the same LCA form, ETA 9035E.

This demonstrates how several categorization models may be implemented and their performance compared to forecast the results of an LCA application. The study also discusses how different models perform when used on local machines and the cloud.

The document has three different goals. To start, find relationships between the dataset's labels and additional attributes.

Secondly, the alteration of Unbalanced datasets is those that contain classes and are therefore skewed or biased. Finally, we compare how well different classification models perform. This study makes use of PySpark, PySpark ML pn Databricks to analyze the performance of different models on different computers.

2. Body

2.1. Dataset

The dataset used throughout the experiment is captured from Data World which contains various observations about H-1B LCA applications filed in the year 2017. It includes administrative information from Labor Condition for Employers Applications and certification decisions conducted by the Department's Office of Foreign Labor Certification, Employment and Training Administration where the determination date was released on or before June 30, 2017, and on or even after October 1, 2016. All information was taken from the iCERT Visa Portal System, an electronic system for filing and processing applications run by the Office of Foreign Labor Certification of petitions for H-1B non-immigrant workers from employers. The figures illustrate the column information of the dataset.

FIELD NAME	DESCRIPTION
CASE_NUMBER	Unique identifier assigned to each application submitted for processing to the Chicago National Processing Center.
CASE_STATUS	Status associated with the last significant event or decision. Valid values include “Certified,” “Certified-Withdrawn,” Denied,” and “Withdrawn”.
CASE_SUBMITTED	Date and time the application was submitted.
DECISION_DATE	Date on which the last significant event or decision was recorded by the Chicago National Processing Center.
VISA_CLASS	Indicates the type of temporary application submitted for processing. R = H-1B; A = E-3 Australian; C = H-1B1 Chile; S = H-1B1 Singapore. Also referred to as “Program” in prior years.
EMPLOYMENT_START_DATE	Beginning date of employment
EMPLOYMENT_END_DATE	Ending date of employment
EMPLOYER_NAME	Name of employer submitting labor condition application.
EMPLOYER_ADDRESS	Contact information of the Employer requesting temporary labor certification
EMPLOYER_CITY	
EMPLOYER_STATE	
EMPLOYER_POSTAL_CODE	
EMPLOYER_COUNTRY	
EMPLOYER_PROVINCE	
EMPLOYER_PHONE	
EMPLOYER_PHONE_EXT	
AGENT_ATTORNEY_NAME	Name of Agent or Attorney filing an H-1B application on behalf of the employer.
AGENT_ATTORNEY_CITY	City information for the Agent or Attorney filing an H-1B application on behalf of the employer.
AGENT_ATTORNEY_STATE	State information for the Agent or Attorney filing an H-1B application on behalf of the employer.

FIELD NAME	DESCRIPTION
JOB_TITLE	Title of the job
SOC_CODE	Occupational code associated with the job being requested for temporary labor condition, as classified by the Standard Occupational Classification (SOC) System.
SOC_NAME	Occupational name associated with the SOC_CODE
NAICS_CODE	Industry code associated with the employer requesting permanent labor condition, as classified by the North American Industrial Classification System (NAICS)
TOTAL_WORKERS	Total number of foreign workers requested by the Employer(s)
FULL_TIME_POSITION	Y = Full Time Position; N = Part Time Position
PREVAILING_WAGE	Prevailing Wage for the job being requested for temporary labor condition.
PW_UNIT_OF_PAY	Unit of Pay. Valid values include “Daily (DAI),” “Hourly (HR),” “Bi-weekly (BI),” “Weekly (WK),” “Monthly (MTH),” and “Yearly (YR)”
PW_SOURCE	Variables include "OES", "CBA", "DBA", "SCA" or "Other"
PW_SOURCE_YEAR	Year the Prevailing Wage Source was Issued
PW_SOURCE_OTHER	If "Other Wage Source", provide the source of wage
WAGE_RATE_OF_PAY_FROM	Employer’s proposed wage rate
WAGE_RATE_OF_PAY_TO	Maximum proposed wage rate
WAGE_UNIT_OF_PAY	Unit of pay. Valid values include “Hour", "Week", "Bi-Weekly", "Month", or "Year"
H-1B_DEPENDENT	Y = Employer is H-1B Dependent; N = Employer is not H-1B Dependent
WILLFUL_VIOLATOR	Y = Employer has been previously found to be a Willful Violator; N = Employer has not been considered a Willful Violator
WORKSITE_CITY	City information of the foreign worker's intended area of employment
WORKSITE_COUNTY	County information of the foreign worker's intended area of employment
WORKSITE_STATE	State information of the foreign worker's intended area of employment
WORKSITE_POSTAL_CODE	Zip Code information of the foreign worker's intended area of employment
ORIGINAL_CERT_DATE	Original Certification Date for a Certified_Withdrawn application

The dataset contains ~528000 observations and 40 columns before any data pre-processing is performed. The attributes are a combination of both numerical data and categorical information.

2.2. Pre-processing

The dataset used from Data World is unprocessed, has noise, and missing data. Useless format, unusable values, and inability to directly feed machine learning models. To improve the precision and effectiveness of a machine learning model, data must be cleaned and made acceptable for the model. In order to prepare the data for the model, we further divided the data pre-processing stage into six parts.

We created a sparks session and checked for the Nan and values in the first stage and removed attributes that are not necessary for analysis and classification. We created a correlation matrix of all the attributes for this and compared to the objective variable and selected key characteristics that would improve categorized the dataset's

observations. After creating a schema of the required variables, what we obtained was as below:

```
Cmd 8
1  pd.DataFrame(df.take(5), columns=df.columns)
```

▸ (1) Spark Jobs

	CASE_STATUS	EMPLOYER_NAME	EMPLOYER_CITY	EMPLOYER_STATE	JOB_TITLE	SOC_NAME	TOTAL_WORKERS	FULL_TIME_POSITION	PREVAILING_WAGE	H-1B_DEPENDENT	WILLFUL_VIOLATOR
0	CERTIFIED-WITHDRAWN	DISCOVER PRODUCTS INC.	RIVERWOODS	IL	ASSOCIATE DATA INTEGRATION	COMPUTER SYSTEMS ANALYSTS	1	Y	59,197.00	N	N
1	CERTIFIED-WITHDRAWN	DFS SERVICES LLC	RIVERWOODS	IL	SENIOR ASSOCIATE	OPERATIONS RESEARCH ANALYSTS	1	Y	49,800.00	N	N
2	CERTIFIED-WITHDRAWN	EASTBANC TECHNOLOGIES LLC	WASHINGTON	DC	.NET SOFTWARE PROGRAMMER	COMPUTER PROGRAMMERS	2	Y	76,502.00	Y	N
3	WITHDRAWN	INFO SERVICES LLC	LIVONIA	MI	PROJECT MANAGER	COMPUTER OCCUPATIONS, ALL OTHER	1	Y	90,376.00	Y	N
4	CERTIFIED-WITHDRAWN	BB&T CORPORATION	WILSON	NC	ASSOCIATE - ESOTERIC ASSET BACKED SECURITIES	CREDIT ANALYSTS	1	Y	116,605.00	N	N

Command took 1.04 seconds -- by jthakur3@stevens.edu at 12/9/2022, 9:42:04 PM on project

We then required the variables: FULL_TIME_POSITION, H-1B_DEPENDENT and WILLFUL_VIOLATOR variables to be numeric so we applied the Boolean logic for Yes and No as '1' and '0' respectively. Below is the results that we obtained.

Cmd 9

```
1  df = df.withColumn("FULL_TIME_POSITION", func.when(df["FULL_TIME_POSITION"] == 'Y', 1).otherwise(0))
2  df = df.withColumn("H-1B_DEPENDENT", func.when(df["H-1B_DEPENDENT"] == 'Y', 1).otherwise(0))
3  df = df.withColumn("WILLFUL_VIOLATOR", func.when(df["WILLFUL_VIOLATOR"] == 'Y', 1).otherwise(0))
```

Command took 0.21 seconds -- by jthakur3@stevens.edu at 12/9/2022, 9:42:04 PM on project

```
Cmd 11
1  pd.DataFrame(df.take(5), columns=df.columns)
```

▸ (1) Spark Jobs

	CASE_STATUS	EMPLOYER_NAME	EMPLOYER_CITY	EMPLOYER_STATE	JOB_TITLE	SOC_NAME	TOTAL_WORKERS	FULL_TIME_POSITION	PREVAILING_WAGE	H-1B_DEPENDENT	WILLFUL_VIOLATOR
0	CERTIFIED-WITHDRAWN	DISCOVER PRODUCTS INC.	RIVERWOODS	IL	ASSOCIATE DATA INTEGRATION	COMPUTER SYSTEMS ANALYSTS	1	1	59,197.00	0	0
1	CERTIFIED-WITHDRAWN	DFS SERVICES LLC	RIVERWOODS	IL	SENIOR ASSOCIATE	OPERATIONS RESEARCH ANALYSTS	1	1	49,800.00	0	0
2	CERTIFIED-WITHDRAWN	EASTBANC TECHNOLOGIES LLC	WASHINGTON	DC	.NET SOFTWARE PROGRAMMER	COMPUTER PROGRAMMERS	2	1	76,502.00	1	0
3	WITHDRAWN	INFO SERVICES LLC	LIVONIA	MI	PROJECT MANAGER	COMPUTER OCCUPATIONS, ALL OTHER	1	1	90,376.00	1	0
4	CERTIFIED-WITHDRAWN	BB&T CORPORATION	WILSON	NC	ASSOCIATE - ESOTERIC ASSET BACKED SECURITIES	CREDIT ANALYSTS	1	1	116,605.00	0	0

Command took 0.77 seconds -- by jthakur3@stevens.edu at 12/9/2022, 9:42:04 PM on project

We, described and summarised the PREVAILING WAGES as below:

	0	1	2	3	4
summary	count	mean	stddev	min	max
PREVAILING_WAGE	400609	74283.97675382979	23865.46989362525	37794.0	144961.0

Now, as CASE_STATUS was the objective variable, we checked for the count of the same in data frame. Below shows the count:

```
1  df.groupBy('CASE_STATUS').count().show()
```

▸ (2) Spark Jobs

```
+--------------------+------+
|         CASE_STATUS| count|
+--------------------+------+
|           CERTIFIED|468970|
|CERTIFIED-WITHDRAWN| 36171|
|           WITHDRAWN| 16016|
|              DENIED|  6989|
+--------------------+------+
```

Command took 15.31 seconds -- by jthakur3@stevens.edu at 12/9/2022, 9:42:04 PM on project

Second, we created a multi-class target variable that was a variable of the binary class. Four categories make up the goal variable: CERTIFIED, CERTIFIED-WITHDRAWN, WITHDRAWN, and DENIED. We consider the active ones so we took the CERTIFIED and DENIED data to get a ratio of the same and the ratio we obtained was 67 as below.

Cmd 23

```
1  major_df = df.filter(func.col("CASE_STATUS") == 'CERTIFIED')
2  minor_df = df.filter(func.col("CASE_STATUS") == 'DENIED')
3  ratio = int(major_df.count()/minor_df.count())
4  print("ratio: {}".format(ratio))
```

▸ (4) Spark Jobs

```
ratio: 67
```

Command took 15.62 seconds -- by jthakur3@stevens.edu at 12/9/2022, 10:21:02 PM on project

The third phase involved reclassifying and re-evaluating the values of several properties. By simply taking into account the H-1B visa class and the USA as the employing country, we rescaled the wages into annual wages.

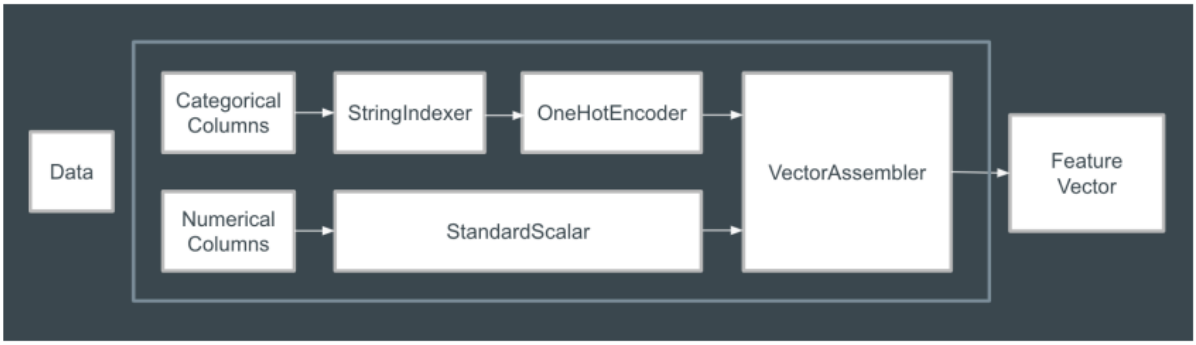
We deal with the dataset's missing data in step four. We got rid of observations.

that lacked an employer name, and we used the mode of the appropriate characteristics to fill in the gaps left by missing values in other attributes. The columns that were no longer needed because they only had solitary values were subsequently deleted. In step five, we use PySpark's StringIndexer and OneHotEncoder functions to

convert category variables into numerical representations. One Hot Encoder transforms categorical data into a binary vector, while String Indexer translates a string column of labels to a column of label indices.

We used Standard Scalar to scale the data for all of the numerical properties so that all values fall between 0 and 1. We made use of Vector Assembler, which merges both the collection of attributes and the features into a single vector column.

The entire process is as follows:



The final set of attributes are:

```
Cmd 22

1  selectedCols = ['label', 'features'] + cols
2  df = df.select(selectedCols)
3  df.printSchema()
```

```
root
 |-- label: double (nullable = false)
 |-- features: vector (nullable = true)
 |-- CASE_STATUS: string (nullable = true)
 |-- EMPLOYER_NAME: string (nullable = true)
 |-- EMPLOYER_CITY: string (nullable = true)
 |-- EMPLOYER_STATE: string (nullable = true)
 |-- JOB_TITLE: string (nullable = true)
 |-- SOC_NAME: string (nullable = true)
 |-- TOTAL_WORKERS: double (nullable = true)
 |-- FULL_TIME_POSITION: integer (nullable = false)
 |-- PREVAILING_WAGE: double (nullable = true)
 |-- H-1B_DEPENDENT: integer (nullable = false)
 |-- WILLFUL_VIOLATOR: integer (nullable = false)

Command took 0.14 seconds -- by jthakur3@stevens.edu at 12/9/2022, 10:21:02 PM on project
```

Handling data that is unbalanced is the last phase. We under-sampled and over-sampled the dataset to achieve this. To match the number of observations in the

majority class, we reproduced the minority class data during oversampling. In an attempt to equal the number of observations in the minority class, we chose random samples from the majority class.

We would also be measuring the performance of the machine learning models, as well as the computing performance using three types of datasets: non-sampled, over-sampled, and under-sampled as below:

Cmd 24

```
a = range(ratio)

# duplicate the minority rows
oversampled_df = minor_df.withColumn("dummy", func.explode(func.array([func.lit(x) for x in a]))).drop('dummy')

# combine both oversampled minority rows and previous majority rows
df_os = major_df.unionAll(oversampled_df)

```

Command took 0.91 seconds -- by jthakur3@stevens.edu at 12/9/2022, 10:21:02 PM on project

Cmd 25

```
df_os.groupBy('CASE_STATUS').count().show()
```

▸ (2) Spark Jobs

```
+------------+------+
|CASE_STATUS| count|
+------------+------+
|  CERTIFIED|468970|
|     DENIED|468263|
+------------+------+
```

Command took 16.48 seconds -- by jthakur3@stevens.edu at 12/9/2022, 10:21:02 PM on project

Cmd 26

```
sampled_majority_df = major_df.sample(False, 1/ratio)
df_us = sampled_majority_df.unionAll(minor_df)

```

Command took 0.13 seconds -- by jthakur3@stevens.edu at 12/9/2022, 10:21:03 PM on project

Cmd 27

```
df_us.groupBy('CASE_STATUS').count().show()
```

▸ (2) Spark Jobs

```
+------------+-----+
|CASE_STATUS|count|
+------------+-----+
|  CERTIFIED| 6966|
|     DENIED| 6989|
+------------+-----+
```

Command took 17.90 seconds -- by jthakur3@stevens.edu at 12/9/2022, 10:21:03 PM on project

2.3. Model Implementation

We have applied three machine learning models for classification: Logistic Regression Classifier, Tree Classifier, and Naive Bayes Classifier. Wet hen assessed the performance based on the F-1 score, then assessed the performance of calculation on

a local machine and Databricks. The section that follows provides information about various models and we have set hyper-parameters for each model.

2.3.1. Decision Tree Classification

A non-parametric supervised learning approach, the decision tree model is used largely for categorization. This model's primary objective is to generate a target value for a specific target variable using straightforward decision-based rules derived from dataset attributes. For this experiment, we employed the "entropy" criterion and set the "depth" hyper-parameter to 10.

2.3.2. Logistic Regression

A statistical model called the Logistic Regression (LR) model is primarily used to categorize variables with dichotomies. This indicates that for the logistic regression models to function optimally, our predictor variables must be binary. Logistic regression has historically scaled well with an increase in data size, which is one of its key advantages. After the oversampling procedure, our dataset contained over 800,000 records, hence another requirement of logistic regression was met. In order to provide enough iterations for analysis, we have set the maximum iterations at "10".

2.3.3. Naive Bayes Classifier

The Bayes Theorem serves as the foundation for Naive Bayes, which aids in the Given that event B has already happened, one can calculate the likelihood that event A will also occur. Furthermore, in the Naive Bayes model, we make the false assumption that none of the model's features or columns are reliant on one another. Here, we can see that the model does not look for any connections between the column characteristics, and as a result, we get one of the key benefits of the Naive Bayes classifier: its performance over time. The

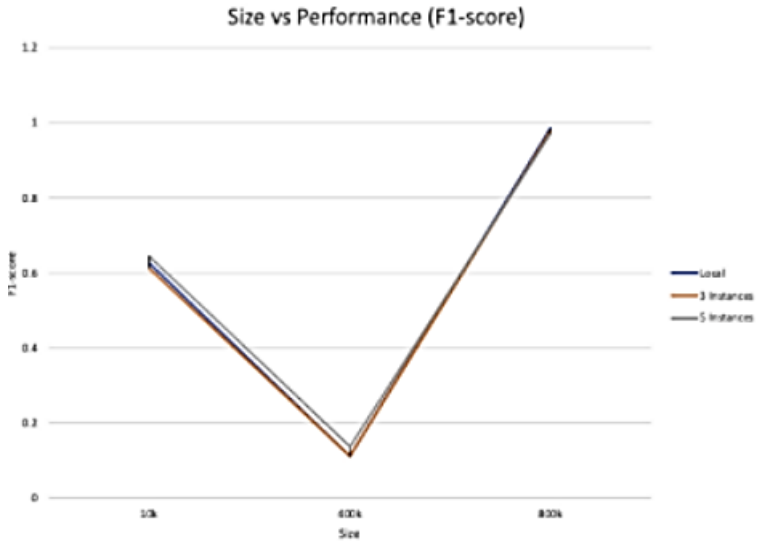
model operates much faster because it's less complex compared to the aforementioned techniques, and frequently works well for huge datasets.

2.4. Performance evaluation

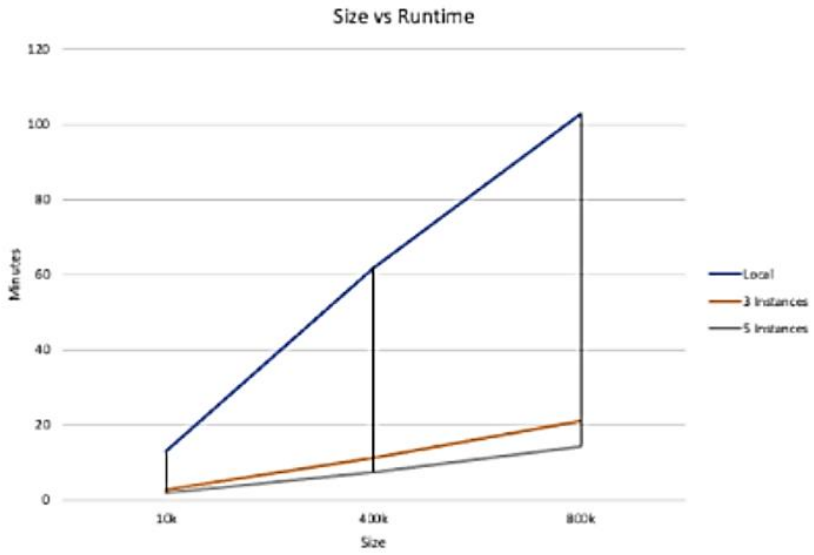
2.4.1. Decision Tree Classifier

First, we tested the model with under-sampled data. The F1-score achieved using this model on the under-sampled dataset containing almost 10,000 records.

The F-1 score is:



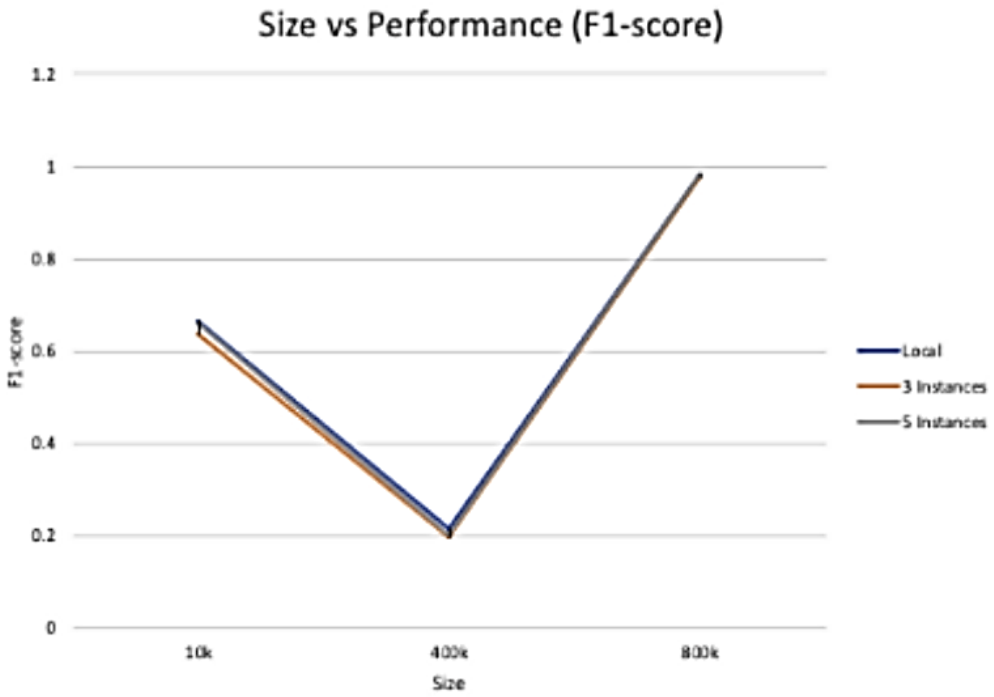
The time performance is:



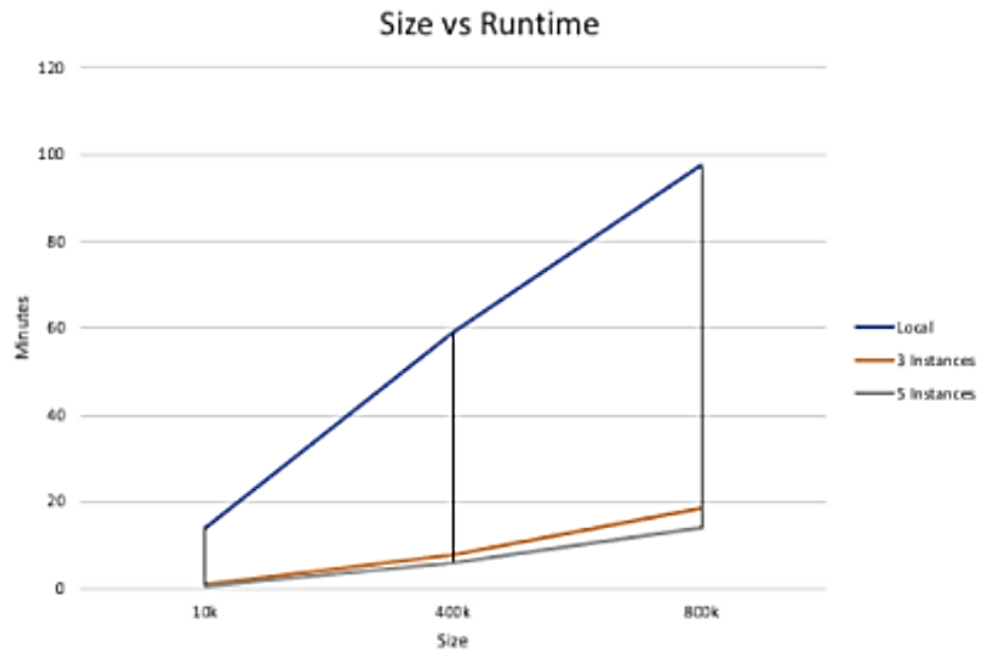
2.4.2. Logistic Regression

First, we tested the model with under-sampled data. First, we tested the model with under-sampled data. Finally, we did for the over-sampled data.

The F-1 score is:



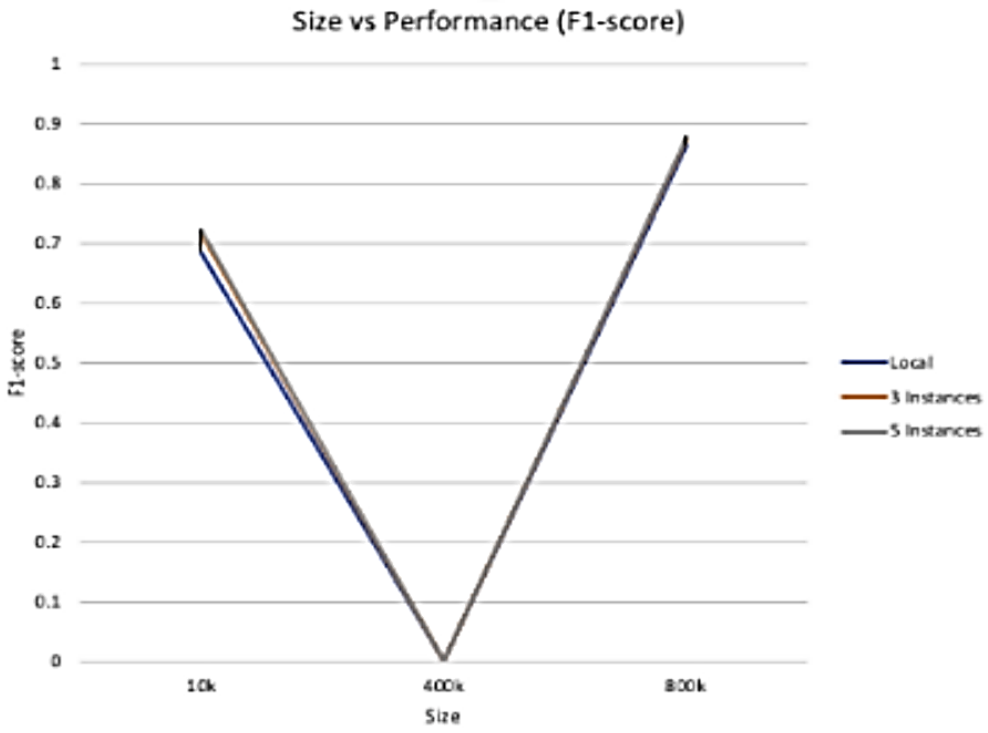
The time-performance is:



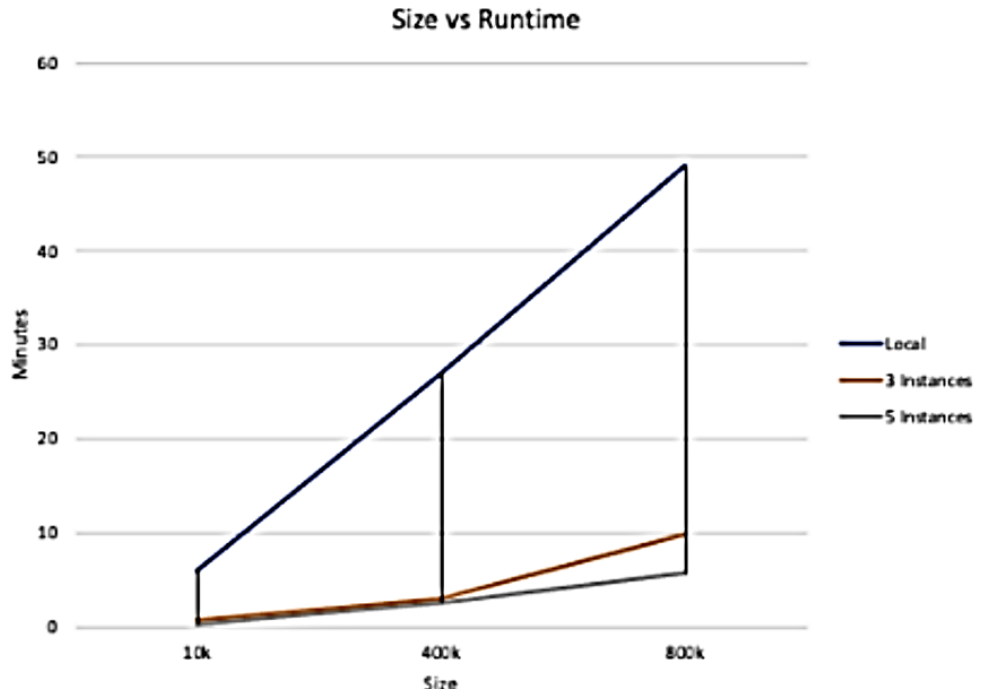
2.4.3. Naive Bayes

First, we tested the model with under-sampled data and then over-sampled. We observed that the logistic regression model gave similar results. We further analyzed the time performance for the decision tree model when running it.

The F-1 score is:



The time performance is:



For all the models, we observed that the change in the number of instances does not drastically impact the F1-score performance, however, it makes a huge impact in terms of the time performance. We then compared the time performance of all the models with respect to time and the number of instances used as well and noted that there is a gradual decrease in the time taken as we increase the number of clusters regardless of the model being applied to the dataset and on which data set it is being applied to.

3. Conclusion

We have derived the following conclusions from the study and testing done above. First, when it came to classifying fresh data, the models performed very similarly. Second, relative to unbalanced and under-sampled data, all models perform better when the data is oversampled.

To compare service performances in the future, we want to create models on more cloud infrastructures including Google Cloud Platform, Microsoft Azure, and IBM Watson. Additionally, we may put into practice several models like XGBoost, SVM, etc., and evaluate how well they perform.

4. References and Appendix

1. Ian Greenleigh, “Data world”, https://data.world/ian/h-1-b-disclosure-data-fy-17/workspace/file?filename=H-1B_FY17_Record_Layout.pdf, 2017.
2. Kumar, “What is H1B LCA ? Why file it? Salary, Processing times – DOL”, February 2022, https://redbus2us.com/what-is-h1b-lca-why-file-it-salary-processing-times-dol/