INST 737 - Intro To Data Science

Milestone 2

Bhavesh Bellara

Gaurav Hasija

Kanishka Jain

Danish Mir

Research Question:

1. Predict the **time duration in days** it takes for a case to be certified or denied.
2. Predict the **case decision.**
3. Predict whether the **agent is present** or not for the case (Random forest).
4. Predict **Threshold Hourly Salary** based on Employee hourly salary and Occupation.

Data Cleaning:

Following the feedback received on Milestone 1, we realized that there were some important changes that needed to be made to our existing dataset to make it ready for the next stage of analysis. Following major changes were made in the aftermath of Milestone 1.

1. At the end of milestone 1, we had our H1B dataset divided into 5 parts based on years. Our first goal was to combine the data under these 5 different datasets into a single dataset to make it easier in conducting further studies on it. While combining, we identified there were a lot of uncommon rows and rows with the same data but different column names. We standardized this and combined the entire data under a single dataset.
2. Next stage involved cleaning the data of redundancies like missing values, incorrect entries, etc. We used techniques like mean, mode to identify these redundancies and ultimately dropped these values to have a clean dataset.
3. The final step in getting the dataset ready for analysis was encoding the data to a standard format to make it easier to answer research questions. For example, we had to standardize the wage column as the entries were of different types like bi-weekly, monthly and annual wage.

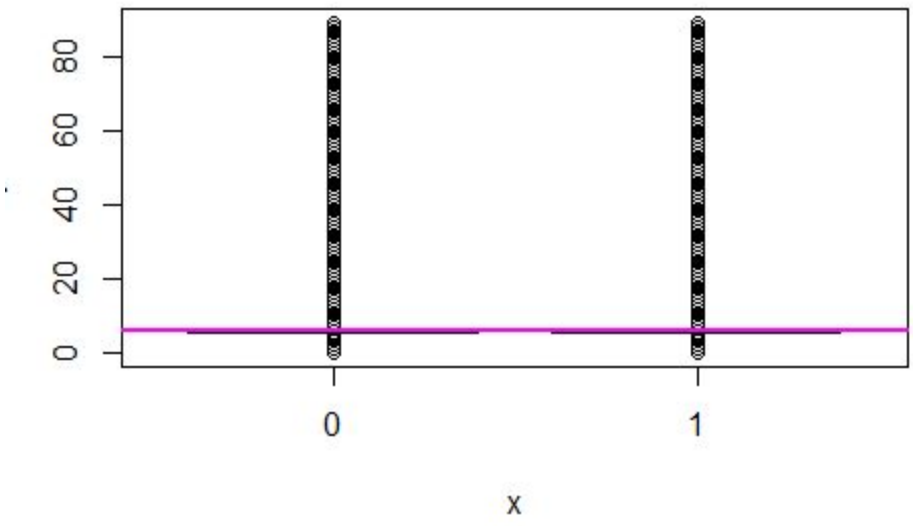
After completing these steps our data was finally ready for further analysis.

Question 1a: Linear Regressions:

As we have 2 numeric dependent variables in our dataset, we tried to apply linear regressions on both of them. We have 5 independent variables that we used i.e. Agent_Present, Wage_Rate_Pay_From_Hour, Occupation, Case_Status. We also used two dependent variables with one another.

First Dependent variable is **Duration**.

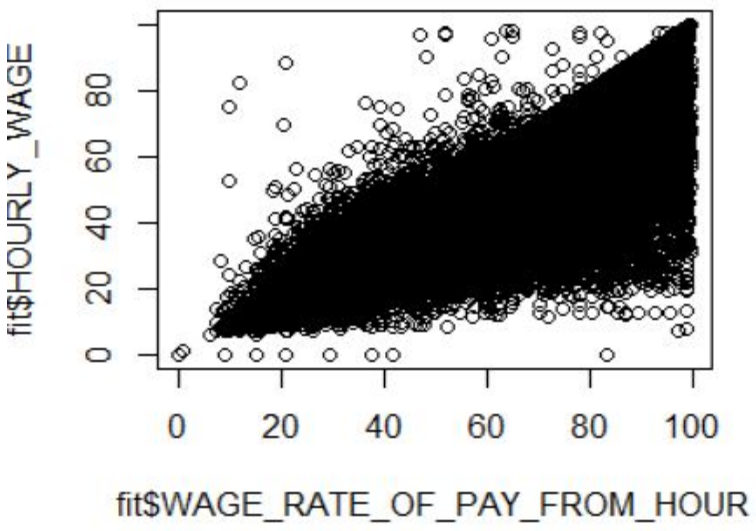
Independent variable	Multiple R-squared
Agent_Present	0.001054
WAGE_RATE_PAY_FROM_HOUR	1.312e-05
HOURLY_WAGE	2.249e-05
OCCUPATION	0.0003591
CASE_STATUS	0.01283



Graph of DURATION(On Y) vs AGENT_PRESENT on X

On analysis we found that none of the independent variables generated accurate predictions.

Second Dependent variable is **Hourly_Wage**. On analysis we found that Wage_Rate_Of_Pay_From_Hour could be used to generate accurate predictions. This can be seen from graphs below.



fitted.value	residual
32.23394	2.56606443
29.49592	-2.19592334
45.31555	6.68445039
34.66772	-1.06772423
36.64518	1.65482248
32.53816	-8.23815916
57.78872	10.61128352
41.74092	5.55907748
36.11279	-5.01278625
27.51847	0.08152995

1b. Multivariate Regression:

After trying out many combinations, we were able to identify that the prediction results improved when we combined Independent variables **Occupation** and **Agent_Present** with Dependent variable **Hourly_Wage**.

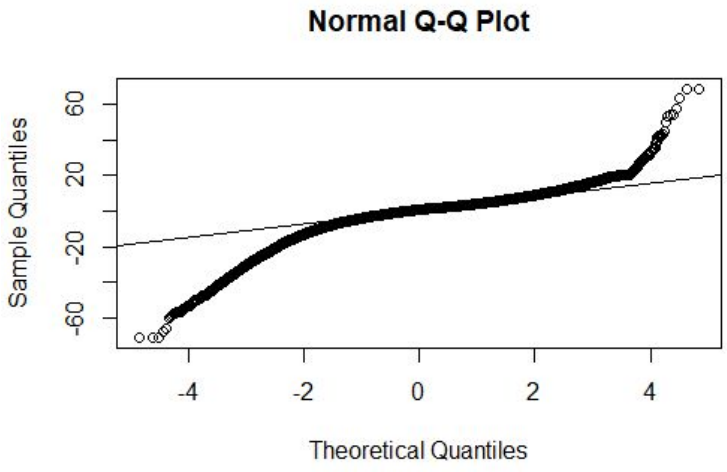
Following were the results that we generated:

Intercept: 2.9555454
Residual standard error: 5.246
Multiple R-squared: 0.7942

The **correlation** between the predicted and actual data for Hourly_Wage is 0.89

The **root mean square error** is 27.25222

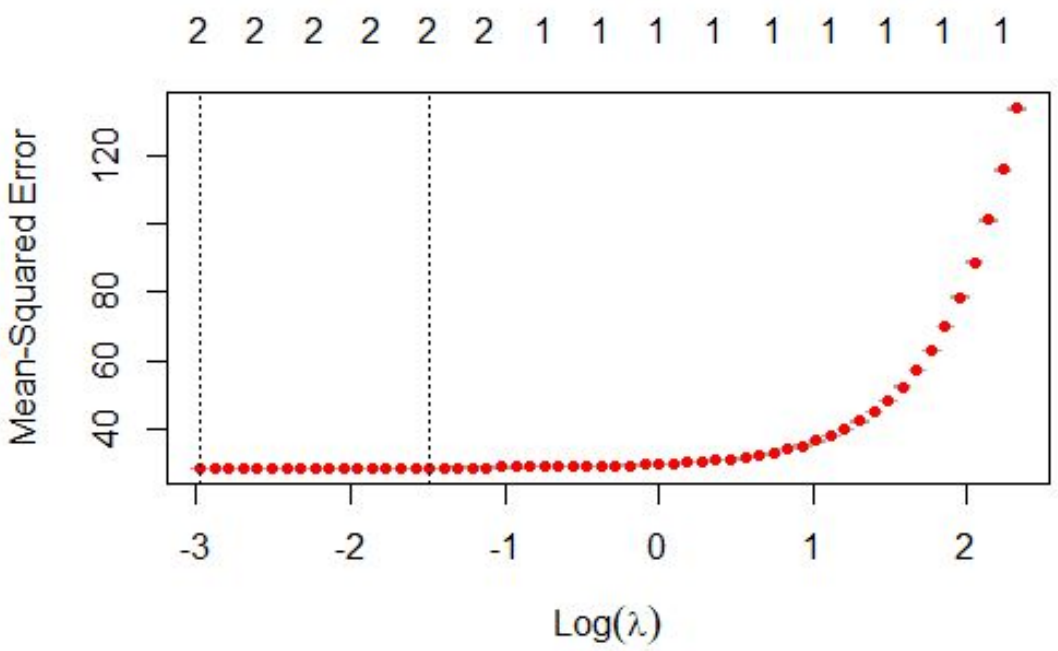
Parameter	Value (Before)	Value (After)
Multiple R-Squared	0.7861	0.7942
RMSE	28.59	27.25



fitted.value	residual
30.55055	-5.4505498
49.65520	-5.9551955
36.41733	3.2826692
30.02404	0.6759562
73.53927	9.3607288
20.36716	2.0328408
35.74039	3.0596055
32.77267	-1.2726650
50.54056	10.4594446
45.70271	-3.1027098
57.47757	-17.7775701

1c: Regularization:

Moving ahead with our analysis, we proceed to repeat the previous two experiments (Q1a & 1b) by adding regularization. We were able to conclude the following:



Even after adding regularization, the correlation remained the same. This suggests that although regularization allowed us to build a simpler model in which we used a few predictive features, the model did not improve. Highlighting that the other features are indeed not very predictive. An important insight obtained from this is that **Duration** did not help us in predicting **Hourly Wage**.

1d: Repeat a-c multiple times with different randomly selected training and testing sets and report differences or similarities across runs.

Train-Test Set	Model	
1	modelWFH <- lm(train$**HOURLY_WAGE**~train$**WAGE_RATE_OF_PAY_FROM_HOUR**)	Multiple R-squared: 0.7861 Root mean square error: 28.59337
1	modelMul2 <- lm(**HOURLY_WAGE~.**,data=train)	Multiple R-squared: 0.7962 Root mean square error” 27.25222
2	modelWFH <- lm(train$**HOURLY_WAGE**~train$**WAGE_RATE_OF_PAY_FROM_HOUR**)	Multiple R-squared: 0.786 Root mean square error: 28.63629
2	modelMul2 <- lm(**HOURLY_WAGE~.**,data=train)	Multiple R-squared: 0.796 Root mean square error” 27.29494
3	modelWFH <- lm(train$**HOURLY_WAGE**~train$**WAGE_RATE_OF_PAY_FROM_HOUR**)	Multiple R-squared: 0.786 Root mean square error: 28.6455
3	modelMul2 <- lm(**HOURLY_WAGE~.**,data=train)	Multiple R-squared: 0.796 Root mean square error”

		27.29956

Different colors indicate different train-test sets. We changed the set.seed parameter to obtain different train-test sets. We can see that after choosing random test and training samples, the results are almost the same.

Question 2: Logistic Regression and NB:

2a: With the knowledge gathered from question 1(b), compute a logistic regression model with respect to different sets of independent features on your training dataset and report.

Solution: Out of four research questions, for the two of those we would be using logistic regression and Naive Bayes as the dependent variable is categorical.

With the knowledge from linear regression model in question 1, we have decided to take these below mentioned independent variables for our logistic regression model with different sets of variables everytime and discard as per significance.

S NO.	INDEPENDENT VARIABLE
1	AGENT_PRESENT_1.0
2	HOURLY_WAGE
3	WAGE_RATE_OF_PAY_FROM_HOUR
4	DURATION
5	CASE_STATUS_1.0

6	OCCUPATION

We split the data set into parts Train (80,000 records) and Test (20,000 records).

First we start with a research question to predict the “case status” based on other features.

Logistic Regression Test 1 for CASE_STATUS:

Dependent Variable: CASE_STATUS

Independent Variables: Intercept: -1.588469

S NO.	INDEPENDENT VARIABLE	DATA TYPE	Coefficients
1	AGENT_PRESENT_1.0	Factor	0.228641
2	HOURLY_WAGE	Numeric	-0.024687
3	WAGE_RATE_OF_PAY_FROM_HOUR	Numeric	0.025982
4	DURATION	Numeric	0.854265
5	OCCUPATION	Factor	Screenshot Below

Coefficients for Occupation:
OCCUPATION Architecture & Engineering -0.155667
OCCUPATION Business Occupation -0.932151
OCCUPATION Computer Occupations 0.382887

OCCUPATION Education Occupations -0.675906	
OCCUPATION Financial Occupation	-0.212068
OCCUPATION Management Occupation 0.383384	
OCCUPATION Marketing Occupation	-1.539586
OCCUPATION Mathematical Occupations	-0.757987
OCCUPATION Medical Occupations	-0.533633

Logistic Regression Test 2 for case_status :

After multiple round of testing with different variables, we verified the p values and determined the significant features and ran the below test with only significant features to predict the probabilities:

Dependent Variable: CASE_STATUS

Independent Variables :

Intercept: -1.498456

Below are the most **significant features**:

S NO.	INDEPENDENT VARIABLE	DATA TYPE	Coefficients
1	HOURLY_WAGE	Numeric	-0.024687
2	WAGE_RATE_OF_PAY_FROM_HOUR	Numeric	0.025982
3	DURATION	Numeric	0.854265

Log Odds:

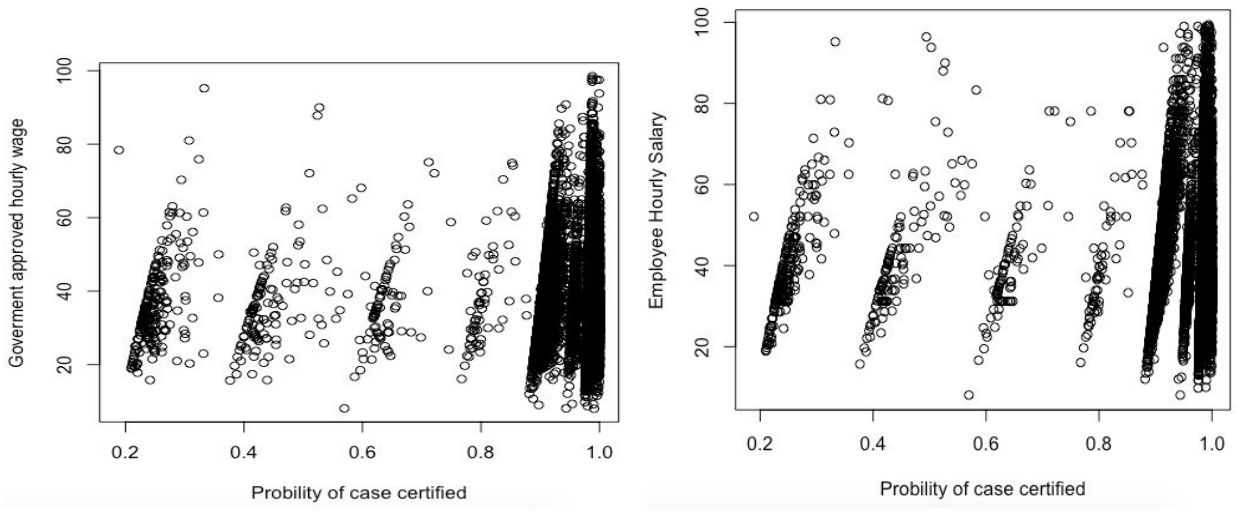
(Intercept)	DURATION	WAGE_RATE_OF_PAY_FROM_HOUR	HOURLY_WAGE
-1.48081118	0.84735533	0.02368765	-0.01543423

Odd ratios:

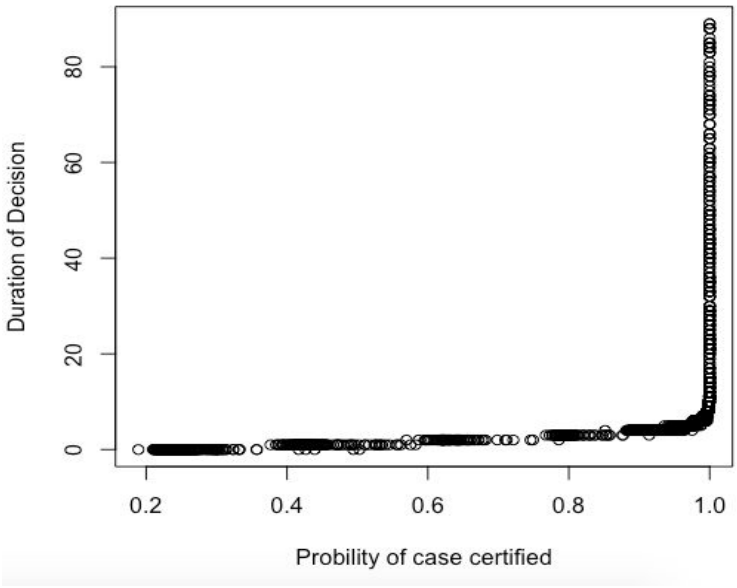
(Intercept)	DURATION	WAGE_RATE_OF_PAY_FROM_HOUR	HOURLY_WAGE
0.2274531	2.3334674	1.0239704	0.9846843

After running the above model for the significant feature and predicting on the test data below are the **test results:**

According to the data collected it was observed as 4% cases are denied and 96% cases are certified. To present the results of the logistic regression model for predicting the case status we will plot the significant predictors against the probability obtained by prediction.



According to the above plots, probability of a case being certified is loosely linked to hourly salary of employee and Government approved threshold salary. It is observed from above that in some cases probability increase with employee's hourly salary.



From the above plot, it can be concluded that if the duration is very low(<10 days) then the chances of getting a H1B visa certified is very less.

Secondly we start with a research question to predict the “Agent present” based on other features.

Logistic Regression Test 1 for AGENT_PRESENT:

Dependent Variable: **AGENT_PRESENT**

Independent Variables :

Intercept: -1.937799

S NO.	INDEPENDENT VARIABLE	DATA TYPE	Coefficients
1	CASE_STATUS_1.0	Factor	0.255145
2	HOURLY_WAGE	Numeric	0.012125
3	WAGE_RATE_OF_PAY_FROM_HOUR	Numeric	0.032415

4	DURATION	Numeric	-0.013284
5	OCCUPATION	Factor	Screenshot Below

Coefficients:	
OCCUPATIONArchitecture & Engineering	1.818540
OCCUPATIONBusiness Occupation	1.383377
OCCUPATIONcomputer occupations	0.218135
OCCUPATIONEducation Occupations	0.954075
OCCUPATIONFinancial Occupation	3.220862
OCCUPATIONManagement Occupation	0.978802
OCCUPATIONMarketing Occupation	3.260425
OCCUPATIONMathematical Occupations	0.568814
OCCUPATIONMedical Occupations	0.429234
OCCUPATIONOthers	1.851894

Logistic Regression Test 2 for AGENT_PRESENT :

After multiple round of testing with different variables, we verified the **p values** and determined the significant features and ran the below test with only significant features to predict the probabilities:

Dependent Variable: **AGENT_PRESENT**

Independent Variables :

Intercept: -0.7739658

Below are the most **significant features**:

S NO.	INDEPENDENT VARIABLE	DATA TYPE	Coefficients
1	WAGE_RATE_OF_PAY_FROM_HOUR	Numeric	-0.0121678
2	DURATION	Numeric	0.0333271

Log odds:

```
(Intercept)                DURATION WAGE_RATE_OF_PAY_FROM_HOUR
-0.77396584          -0.01216780                   0.03332707
```

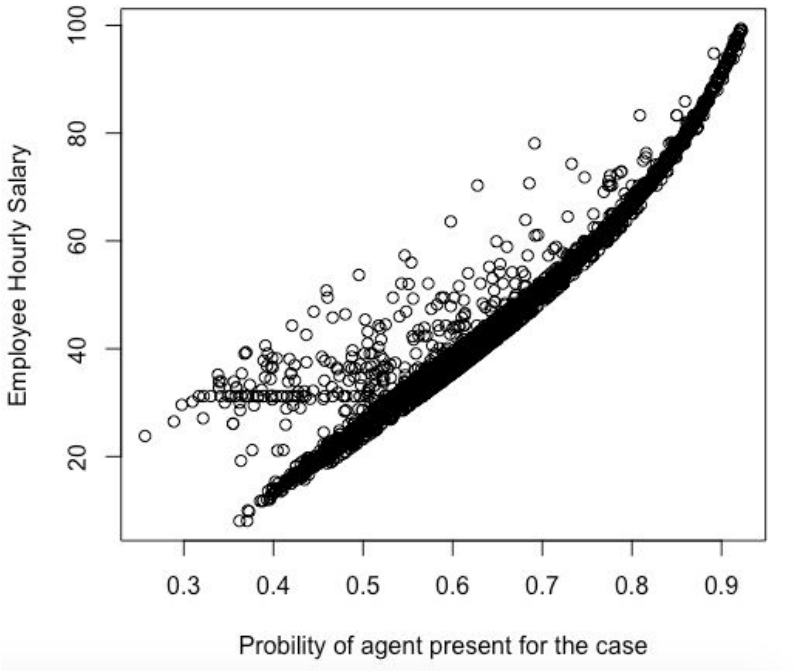
Odd ratios:

```
(Intercept)                DURATION WAGE_RATE_OF_PAY_FROM_HOUR
  0.4611805          0.9879059                   1.0338886
```

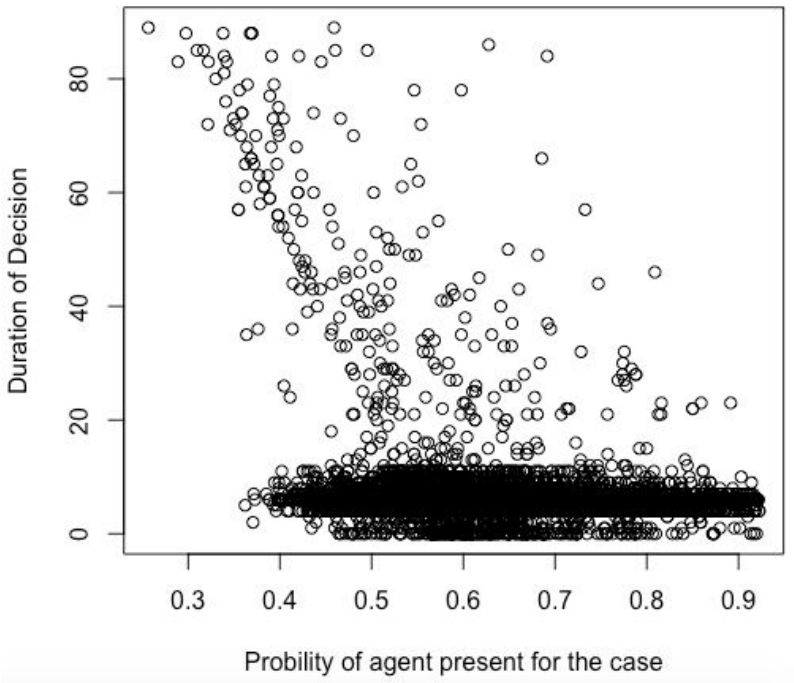
After running the above model for the significant feature and predicting on the test data below are the **test results:**

To present the results of the logistic regression model for predicting the case status we will plot the significant predictors against the probability obtained by prediction.

1) Probability of agents being present increases with the employees hourly wage, it shows **high paid employees are usually given attorneys by the company for their H-1B case.**



2) Probability of agent being present decreases with the duration, it shows **to get the results of H-1B quickly, agent needs to be hired for the case.**



2b: Next, we are going to report classification results using Naïve Bayes. Remember to categorize all your variables before running the classifier.

Solution: Now we apply the naive bayes model to the data for determining our two dependent variable for our two research question:

1) Predict the case status.

2) Predict if the agent is present or not.

- We split the data set into parts Train (80,000 records) and Test (20,000 records).
- We convert all the numeric variables to categorical by dividing them into a range of intervals from min to max.

HOURLY_RATE_RANGE: min=0, max=100 interval by 10.

WAGE_RANGE: min=0, max=100 interval by 10.

DURATION_RANGE: min=0, max=90 interval by 10.

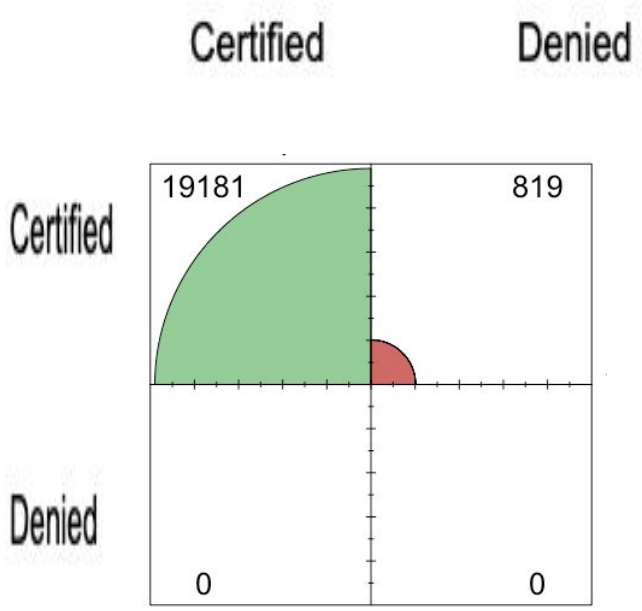
After running the naive bayes model for classifying the case status and agent present below are our observed confusion matrix.

Independent Variables for classifying case status:

S NO.	INDEPENDENT VARIABLE	DATA TYPE
1	AGENT_PRESENT_1.0	Factor
2	HOURLY_WAGE_RANGE	Factor
3	WAGE_RATE_OF_PAY_FROM_HOUR_RANGE	Factor
4	DURATION_RANGE	Factor
5	OCCUPATION	Factor

After taking the above variables, we apply the NB model using our train data and predict the probabilities on the test data to obtain the result of the model.

Confusion Matrix for classifying the CASE_STATUS:



As data contains 95 % of certified cases and only 4 percent of Denied cases, our model is not a perfect fit for case _status prediction.

Independent Variables for classifying AGENT PRESENT:

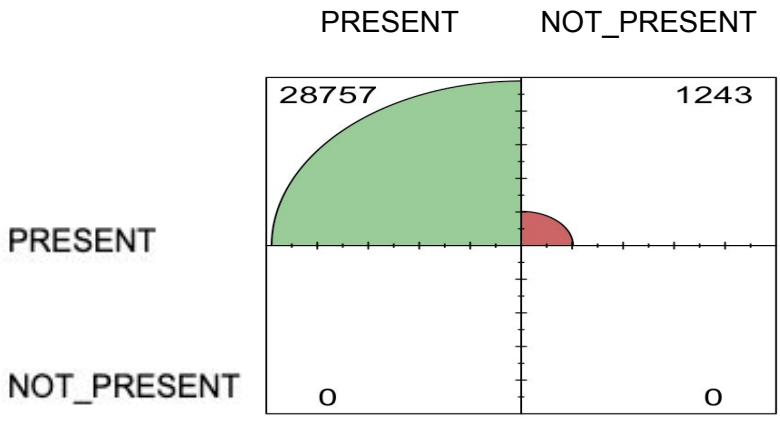
S NO.	INDEPENDENT VARIABLE	DATA TYPE
1	CASE_STATUS_1.0	Factor
2	HOURLY_WAGE_RANGE	Factor
3	WAGE_RATE_OF_PAY_FROM_HOUR_RANGE	Factor
4	DURATION_RANGE	Factor

5	OCCUPATION	Factor

After taking the above variables in all sets of combinations to maximize the classification. We predict the probabilities on the test data to obtain the result.

Train data is of 70,000 records. Test data is of 30,000 records.

Confusion Matrix for classifying the AGENT_PRESENT:



All the conditional Probabilities are low hence it is suspected that the model is not showing expected results.

To improve the efficiency of model predicting the **CASE STATUS:**

We tried Undersampling the dataset by selecting the same number of **certified** and **denied** cases. It did not improve the efficiency and below confusion matrices were observed after under sampling.

Confusion Matrix:

predmodelCS	CERTIFIED	DENIED
CERTIFIED	22500	22500
DENIED	0	0

To improve the efficiency of model predicting the **AGENT PRESENT:**

We tried Undersampling the dataset by selecting the same number of **certified** and **denied** cases. It did not improve the efficiency and below confusion matrices were observed after under sampling.

Confusion Matrix:

predmodelAP		PRESENT	NOT_PRESENT
	PRESENT	15000	15000
	NOT_PRESENT	0	0

Adding the Laplace Estimator did not improve the efficiency of the model and results remain the same.

Question 3: Decision Trees and Random Forests:

Solution: We will use DecisionTree Classifier to predict the **CASE_STATUS** in our dataset, which is a categorical value with Values as Certified or Denied.

Our dataset has 921389 Certified cases and 40557 Denied cases.

We will start with variable 'DURATION', "HOURLY_WAGE", 'AGENT_PRESENT_0.0', 'AGENT_PRESENT_1.0' to create a Decision Tree Classifier for CASE_STATUS_0.0 and do further prediction.

```
#setting up decision tree
from sklearn.model_selection import train_test_split
X=np.array(cdf[['DURATION', "HOURLY_WAGE", 'AGENT_PRESENT_0.0', 'AGENT_PRESENT_1.0']])
y=cdf['CASE_STATUS_0.0']
X_trainset,X_testset,y_trainset,y_testset=train_test_split(X,y,test_size=0.3,random_state=6)
```

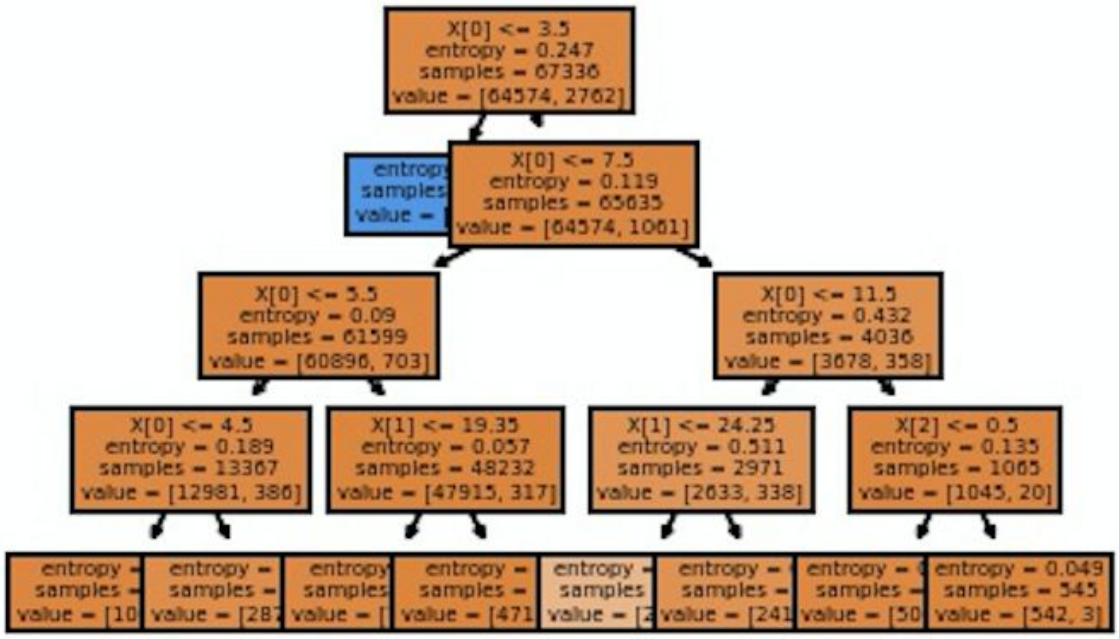
Python package “from sklearn.model_selection import train_test_split” was used to set up a decision tree and create testset and trainset with test_size=0.3, and random_state=6).

Next we created the Training and Test Data set. In order to avoid the potential issue of over

fitting, we used a split function with test size 0.3 to create our data sets.

We created our tree using DecisionTreeClassifier

```
#modeling
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
CASE_STATUS_tree=DecisionTreeClassifier(criterion="entropy", max_depth=4)
```



We next used our model to make predictions.

```
#Prediction
predTree = CASE_STATUS_tree.predict(X_testset)
print(predTree[0:5])
print(y_testset[0:5])
```

```
['Denied' 'Denied' 'Denied' 'Denied' 'Denied']
252327    Denied
267090    Denied
738335    Denied
760390    Denied
930954    Denied
Name: CASE_STATUS, dtype: category
Categories (2, object): [Certified, Denied]
```

This was followed by evaluation

```
#Evaluation
from sklearn import metrics
import matplotlib.pyplot as plt
print("DecisionTree's Accuracy:", metrics.accuracy_score(y_testset,predTree))
```

```
DecisionTree's Accuracy: 0.9837832218718597
```

Our DecisionTree's Accuracy came to be 0.9837832218718597

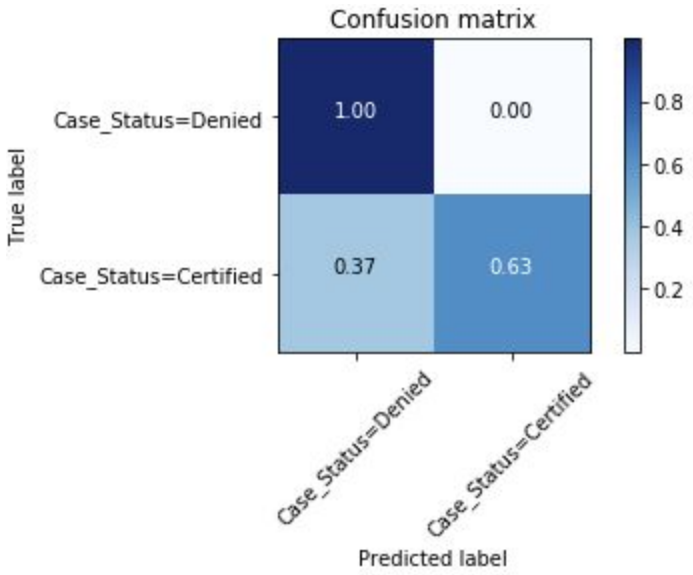
The Accuracy classification score computes subset accuracy: the set of labels predicted for a sample must exactly match the corresponding set of labels in y_true.

In multilabel classification, the function returns the subset accuracy. If the entire set of predicted labels for a sample strictly matches with the true set of labels, then the subset accuracy is 1.0; otherwise it is 0.0.)

Confusion matrix

```
# Compute confusion matrix
cnf_matrix = confusion_matrix(y_testset, predTree, labels=["Certified", "Denied"])
np.set_printoptions(precision=2)
# Plot normalized confusion matrix
plt.figure()
plot_confusion_matrix(cnf_matrix, classes=['Case_Status=Denied','Case_Status=Certified'],normalize= True,  title='Confusion matrix')
```

```
Normalized confusion matrix
[[1.   0.  ]
 [0.37 0.63]]
```



The confusion matrix showcases our model is able to predict 100% for Denied cases and 63% for Certified cases, which can be considered a decent result for prediction.

Further boosting was used to improve the model. We used AdaBoosterClassifier from sklearn Library in Python.

```
from sklearn.model_selection import cross_val_score
from sklearn.ensemble import AdaBoostClassifier
```

```
clf = AdaBoostClassifier(n_estimators=100)# applying booster
```

```
scores = cross_val_score(clf, X, y, cv=5)
scores.mean()
```

```
bdt_real = AdaBoostClassifier(
    DecisionTreeClassifier(max_depth=2),
    n_estimators=600,
    learning_rate=1)
bdt_discrete = AdaBoostClassifier(
    DecisionTreeClassifier(max_depth=2),
    n_estimators=600,
    learning_rate=1.5,
    algorithm="SAMME")
```

As we furthered with training/test model for our boosted data, we saw accuracy improving from 0.9825357774004643 to 0.9838525243424928

```
bdt_real.fit(X_trainset, y_trainset)
bdt_discrete.fit(X_trainset, y_trainset)
```

```
print("Real DecisionTree's Accuracy:", metrics.accuracy_score(real_test_predict, y_testset))
```

```
Real DecisionTree's Accuracy: 0.9825357774004643
```

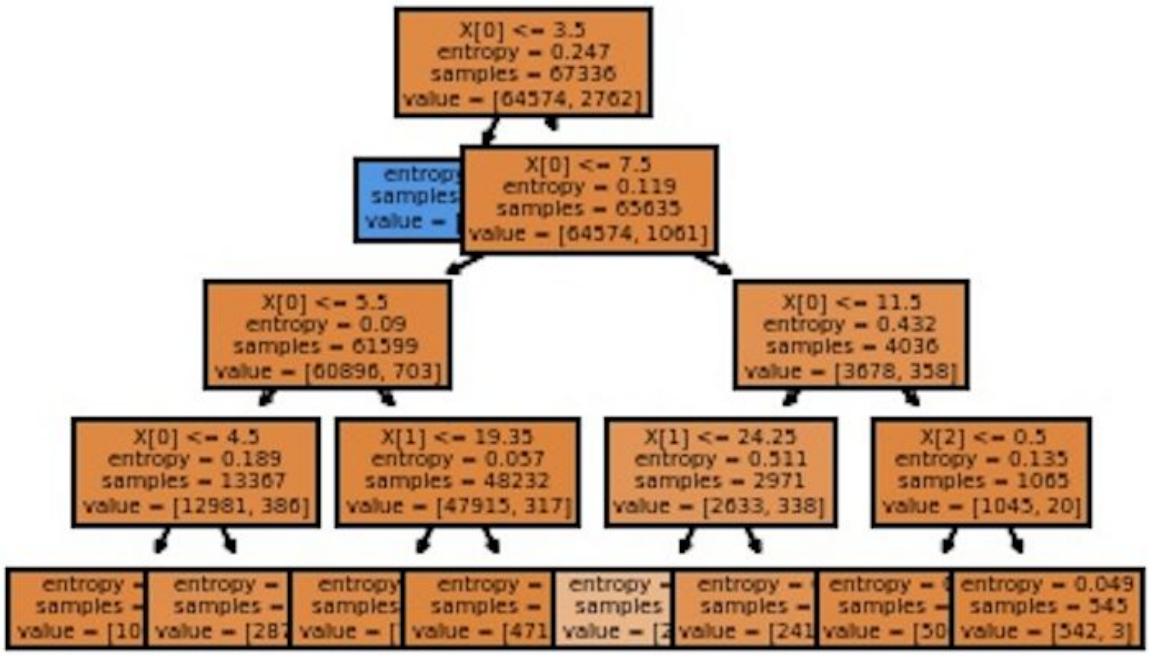
```
print("ADAboost DecisionTree's Accuracy:", metrics.accuracy_score(discrete_train_predict, y_testset))
```

```
ADAboost DecisionTree's Accuracy: 0.9838525243424928
```

We also want to predict the Agent present or absent in our research questions.

We again created a testset and trainset for Agent Present. We used 'DURATION', "HOURLY_WAGE", 'CASE_STATUS_0.0', 'CASE_STATUS_1.0' to predict 'AGENT_Presrent_0.0'

```
#setting up decision tree
from sklearn.model_selection import train_test_split
X=np.array(cdf[['DURATION', "HOURLY_WAGE", 'CASE_STATUS_0.0', 'CASE_STATUS_1.0']])
y=cdf['AGENT_PRESENT_0.0']
X_trainset,X_testset,y_trainset,y_testset=train_test_split(X,y,test_size=0.3,random_state=6)
```



Prediction:

```
#Prediction
predTree = AGENT_PRESENT_tree.predict(X_testset)
print(predTree[0:5])
print(y_testset[0:5])
```

```
['Absent' 'Absent' 'Absent' 'Absent' 'Absent']
569824     Absent
84152      Absent
26099     Present
468201     Absent
546271     Absent
Name: AGENT_PRESENT_0.0, dtype: category
Categories (2, object): [Absent, Present]
```

Evaluation: (DecisionTree's Accuracy: 0.6259009425244871)

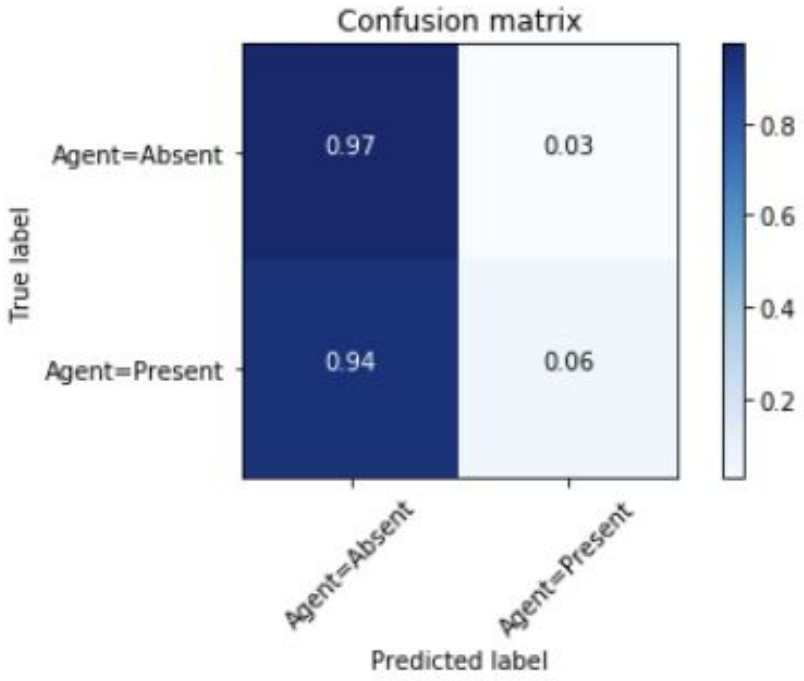
```
#Evaluation
from sklearn import metrics
import matplotlib.pyplot as plt
print("DecisionTree's Accuracy:", metrics.accuracy_score(y_testset,predTree))
```

```
DecisionTree's Accuracy: 0.6259009425244871
```

Confusion matrix:

```
# Compute confusion matrix
cnf_matrix = confusion_matrix(y_testset, predTree, labels=["Absent", "Present"])
np.set_printoptions(precision=2)
# Plot normalized confusion matrix
plt.figure()
plot_confusion_matrix(cnf_matrix, classes=['Agent=Absent','Agent=Present'],normalize= True,  title='Confusion matrix')
```

```
Normalized confusion matrix
[[0.97 0.03]
 [0.94 0.06]]
```



The confusion matrix showcases our model is able to predict 97% for Absent cases and 6% for Present cases, which perhaps can not be considered a good enough result.

Boosting:

We used AdaBoosterClassifier on our data set and created a train / test sets.

```
bdt_real = AdaBoostClassifier(
    DecisionTreeClassifier(max_depth=2),
    n_estimators=600,
    learning_rate=1)
bdt_discrete = AdaBoostClassifier(
    DecisionTreeClassifier(max_depth=2),
    n_estimators=600,
    learning_rate=1.5,
    algorithm="SAMME")
```

```
bdt_real.fit(X_trainset, y_trainset)
bdt_discrete.fit(X_trainset, y_trainset)
```

```
AdaBoostClassifier(algorithm='SAMME',
                   base_estimator=DecisionTreeClassifier(ccp_alpha=0.0,
                                                         class_weight=None,
                                                         criterion='gini',
                                                         max_depth=2,
                                                         max_features=None,
                                                         max_leaf_nodes=None,
                                                         min_impurity_decrease=0.0,
                                                         min_impurity_split=None,
                                                         min_samples_leaf=1,
                                                         min_samples_split=2,
                                                         min_weight_fraction_leaf=0.0,
                                                         presort='deprecated',
                                                         random_state=None,
                                                         splitter='best'),
                   learning_rate=1.5, n_estimators=600, random_state=None)
```

As we furthered with training/test model for our boosted data, we saw accuracy reducing from 0.6395536869340233 to 0.6294816115320643, this can be attributed to low efficcacy of model in this case.

```
print("Real DecisionTree's Accuracy:", metrics.accuracy_score(real_test_predict, y_testset))
```

```
Real DecisionTree's Accuracy: 0.6395536869340233
```

```
print("ADAboost DecisionTree's Accuracy:", metrics.accuracy_score(discrete_train_predict, y_testset))
```

```
ADAboost DecisionTree's Accuracy: 0.6294816115320643
```

Question 4: Comparative Analysis.

For comparative analysis on classifiers, we will take below two research questions which we have answered based on the classifiers in question 2 and 3.

1. Predict the **case decision.**
2. Predict whether the **agent is present** or not for the case (Random forest)

Logistic regression for case Status:

- From the graphs plotted in question (2) part a, it can be observed that probability of being certified **increases** with **employee hourly salary** and **government threshold salary**.
- Though wages are loosely linked with case status, we have "**time duration to decision"** which has good significance with the **probability of case being certified** and **shows that extremely lower days to duration will lead to case being denied.**

Logistic regression for Agent Present:

- Probability of agent being present increases with the employees hourly wage, **it shows high paid employees are usually given attorneys by the company for their H-1B case.**
- Probability of agent being present **decreases** with the duration, **it shows to get the results of H-1B quickly, agent needs to be hired for the case.**

Naive Bayes for CASE STATUS:

As per the confusion matrix reported in question 2 part b, model is unable to classify the case status correctly. As all the classification leads to majority class.

Undersampling was performed by equally dividing the true and false sets which did not improve classification efficiency.

Naive Bayes for AGENT PRESENT:

As per the confusion matrix reported in question 2 part b, model is unable to classify the agent present correctly. As all the classification leads to majority class.

Undersampling was performed by equally dividing the true and false sets which did not improve classification efficiency.

Decision tree:

Case Status

The independent dependent variables for this case include: 'DURATION', "HOURLY_WAGE", 'AGENT_PRESENT_0.0', 'AGENT_PRESENT_1.0', and the dependent variable is ‘CASE_STATUS_0.0.’

We made test and train data sets with test_size = 0.3. The separation of test and training data sets helped us to avoid the problem of overfitting. As we created our DecisionTreeClassifier, we played around with tree depth, and realised best results come for max_depth = 4, hence we choose this as our max_depth.

Our DecisionTree’s accuracy came to be 0.9837832218718597. The confusion matrix showcases our model is able to predict 100% for Denied cases and 63% for Certified cases, which can be considered a decent result for prediction.

We further used AdaBoostClassifier and saw an improvement in accuracy from 0.9825357774004643 to 0.9838525243424928. Comparing this DecisionTreeClassifier with

other classifiers used for predicting Case_Status, we conclude DecisionTreeClassifier is perhaps showcases the best results for Case_status.

Agent_present:

The independent dependent variables for this case include: 'DURATION', "HOURLY_WAGE", 'CASE_STATUS_0.0', 'AGENT_PRESENT_1.0', and the dependent variable is ‘AGENT_PRESENT_0.0.’

Again, we made test and train data sets with test_size = 0.3. As we created our DecisionTreeClassifier, we played around with tree depth, and realised best results come for max_depth = 4, hence we choose this as our max_depth.

In this case our, our DecisionTree's Accuracy came out to be 0.6259009425244871. The confusion matrix showcases our model is able to predict 97% for Absent cases and 6% for Present cases, which perhaps can not be considered a good enough result.

We further used AdaBoostClassifier and saw an improvement in accuracy from 0.6395536869340233 to 0.6294816115320643. Comparing this DecisionTreeClassifier with other classifiers used for predicting Agent Present, we conclude Naive Bayes Classifier showcased better results.

Based on above results we have concluded to use following models for our research questions:

1. Predict the case decision- **DECISION TREE**
2. Predict whether the agent is present or not for the case- **LOGISTIC REGRESSION.**
3. Predict **Hourly Wage - Linear Regression**
4. Predict **Duration - No proper predictor**