US Permanent Visa Case Study

Chandramouli Yalamanchili

Case Study: Perform analysis to identify different important factors that could impact the US permanent visa application. Also build a model to predict the approval of the US permanent visa application.

I have used the US permanent visa dataset and run various graph analysis on the selected features to see how each of them impact the outcome of the US permanent visa application.

Goal:

- To understand how different variables/features impact the decision of the US permanent visa applications using different types of graph analysis techniques.

Dataset:

- Original dataset has been taken from https://www.kaggle.com/jboysen/us-perm-visas.
- In previous classes, I have applied several data preparation techniques on this dataset by combining few columns and by normalizing the data in few columns.
- For this exercise I have used the the final CSV file created out of my previous exercises.
- The input file used is attached with this post – *us_perm_visas_final.csv.*

Features used:

Below are the features I have extracted and used from the dataset.

- case_status - This is the status of the US Permanent visa application.
- entry_visa – Type of visa that the candidate entered into USA with.
- citizenship - Country of citizenship of the candidate.
- no_of_employees - Number of employees under the employer who filed petition for the candidate.
- state - USA state where employer is located.
- job_level - Level of the job role, or expertise level of the candidate.
- year - Year of the application decision.
- salary - Salary offered to the candidate for the position.

Below are some of the other variables present in the input dataset that I dropped for this case study as I didn't see them fit.

- employer_name
- job_info_work_city
- pw_job_title_9089
- pw_soc_title
- birth_country

The step-by-step instructions to perform the graph analysis:

1. Load the data from the 'us_perm_visas_final.csv' into pandas data frame. Displayed the initial dataframe structure as part of this step to understand the data I have currently in the dataframe. I didn't drop rows with missing values yet. These are the details of raw dataset.

```
The dimension of the table is:  (374362, 17)

Top 5 rows      case_number case_status class_of_admission country_of_citizenship  \
0  A-07323-97014   Certified                J-1                ARMENIA
1  A-07332-99439      Denied                B-2                 POLAND
```

```
2  A-07333-99643   Certified               H-1B                   INDIA
3  A-07339-01930   Certified                B-2             SOUTH KOREA
4  A-07345-03565   Certified                L-1                  CANADA

  decision_date               employer_name  employer_num_employees  \
0    2012-02-01             NETSOFT USA INC.                     NaN
1    2011-12-21  PINNACLE ENVIRONEMNTAL CORP                     NaN
2    2011-12-01   SCHNABEL ENGINEERING, INC.                     NaN
3    2011-12-01      EBENEZER MISSION CHURCH                     NaN
4    2012-01-26   ALBANY INTERNATIONAL CORP.                     NaN

               employer_name.1 employer_state  \
0            NETSOFT USA INC.             NY
1  PINNACLE ENVIRONEMNTAL CORP             NY
2   SCHNABEL ENGINEERING, INC.             VA
3      EBENEZER MISSION CHURCH             NY
4   ALBANY INTERNATIONAL CORP.             NY

  foreign_worker_info_birth_country job_info_work_city job_info_work_state  \
0                               NaN          New York                   NY
1                               NaN          New York                   NY
2                               NaN        Lutherville                   MD
3                               NaN           Flushing                   NY
4                               NaN             Albany                   NY

                          pw_job_title_9089 pw_level_9089  \
0  Computer Software Engineers, Applications      Level II
1                          ASBESTOS HANDLER       Level I
2                            Civil Engineer       Level I
3                                File Clerk      Level II
4                  Sales & Service Engineer      Level IV

                               pw_soc_title pw_amount_9089  \
0  Computer Software Engineers, Applications        75629.0
1        Hazardous Materials Removal Workers        37024.0
2                           Civil Engineers        47923.0
3                               File Clerks          10.97
4                           Sales Engineers        94890.0

  pw_unit_of_pay_9089
0                  yr
1                  yr
2                  yr
3                  hr
4                  yr

Rows with missing data by column:
 case_number                             0
case_status                              0
class_of_admission                   22845
country_of_citizenship                  59
decision_date                            0
employer_name                           12
employer_num_employees              135349
employer_name.1                         12
employer_state                          42
foreign_worker_info_birth_country   135300
job_info_work_city                     102
job_info_work_state                    103
pw_job_title_9089                      392
pw_level_9089                        27627
pw_soc_title                          2336
pw_amount_9089                        2216
pw_unit_of_pay_9089                   1572
```

2. Data clean up and preparation as needed for graph analysis. I have performed below data clean up steps to extract the data in the format needed for graph analysis. Majority of the changes as part of case study part 2 are done as part of this step.
   - Dropped the rows that have NA values in the columns - 'class_of_admission', 'country_of_citizenship', 'employer_state', and 'pw_unit_of_pay_9089'.
   - Selected only few interested columns and dropped the rest of the columns. Also, renamed the column names.
   - Field extraction - extracted salary field using two fields, pw_amount_9089 & pw_unit_of_pay_9089, I have extracted the yearly salary for all rows.
   - Selected rows from years 2014, 2015 & 2016 years only to reduce the dataset size.
   - Transformed job_level data from text into numeric by assigning unique value for each job level.
   - At this point I had all of the selected rows, so I have done reset_index to reset the index on the dataframe.
   - Below is how dataset looks like after step 2:

```
Dataset state after step 2 - feature and data selection

The dimension of the table is:  (279365, 8)

Top 5 rows          case_status entry_visa  citizenship  no_of_employees
state  \
0  Certified-Expired       H-1B        INDIA              NaN  MASSACHUSETTS
1  Certified-Expired       H-1B        INDIA              NaN       ARKANSAS
2          Certified       H-1B        INDIA              NaN       NEW YORK
3  Certified-Expired       H-1B  SOUTH KOREA              NaN     CALIFORNIA
4          Certified       H-1B        INDIA              NaN      WISCONSIN

   job_level        year    salary
0   Level IV  2014-02-21  116542.4
1    Level I  2014-01-08   42973.0
2  Level III  2014-05-22  101629.0
3   Level II  2014-03-28   60445.0
4   Level IV  2014-05-28   92414.0

Rows with missing data by column:
 case_status            0
entry_visa              0
citizenship             0
no_of_employees     57167
state                   0
job_level           19805
year                    0
salary                  0
dtype: int64
```

3. Modify the feature values on selected rows - in this step I have modified the extracted feature values either to normalize them, or to fill missing values or to extract more meaningful value.
   - Filling the missing values in the dataset - as you have seen above, after step 2 we still have some missing fields, I have filled those as below:
     1. No_of_employees - used the median to fill the missing values.
     2. Job_level - used the most frequent value to fill the missing values.
   - Value normalization - I have used log normalization to normalize the salary & no_of_employees features to remove the skewed data we have seen last week. Below is the data before and after log normalization:

```
Top 5 rows after log transformation for salary & no_of_employees along with the or
iginal values
      salary  salary_log  no_of_employees  no_of_employees_log
0  116542.4   11.666019           1634.0             7.399398
1   42973.0   10.668351           1634.0             7.399398
2  101629.0   11.529094           1634.0             7.399398
3   60445.0   11.009506           1634.0             7.399398
4   92414.0   11.434045           1634.0             7.399398
```

- Year - derived the 4 digit year from the date.
- Below is how the dataset looks after step 3:

```
Dataset state after step 3 - modifying some features and filling missing values

The dimension of the table is:  (279365, 10)

Top 5 rows          case_status entry_visa  citizenship  no_of_employees
state  \
0  Certified-Expired       H-1B        INDIA           1634.0  MASSACHUSETTS
1  Certified-Expired       H-1B        INDIA           1634.0       ARKANSAS
2          Certified       H-1B        INDIA           1634.0       NEW YORK
3  Certified-Expired       H-1B  SOUTH KOREA           1634.0     CALIFORNIA
4          Certified       H-1B        INDIA           1634.0      WISCONSIN

   job_level  year    salary  salary_log  no_of_employees_log
0        4.0  2014  116542.4   11.666019             7.399398
1        1.0  2014   42973.0   10.668351             7.399398
2        3.0  2014  101629.0   11.529094             7.399398
3        2.0  2014   60445.0   11.009506             7.399398
4        4.0  2014   92414.0   11.434045             7.399398

Rows with missing data by column:
 case_status            0
entry_visa              0
citizenship             0
no_of_employees         0
state                   0
job_level               0
year                    0
salary                  0
salary_log              0
no_of_employees_log     0
```

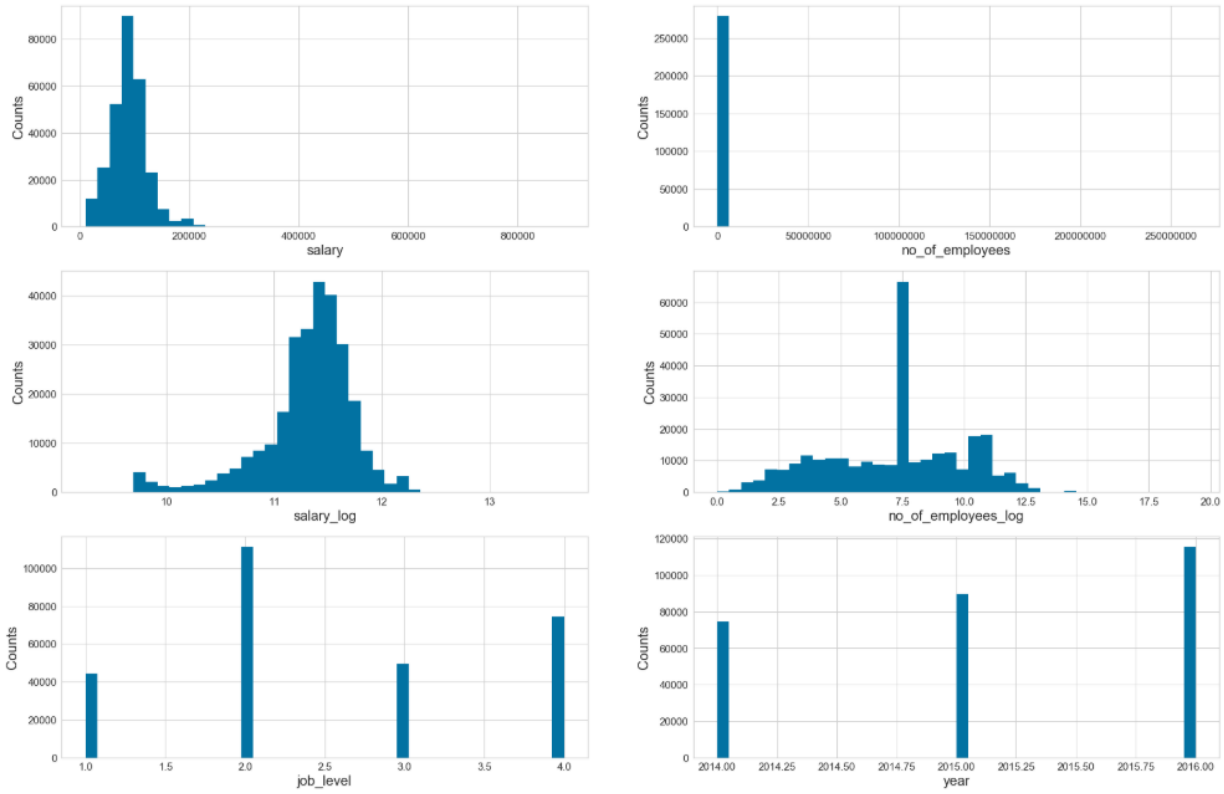
4. Understand different variable types we have in the dataset, I ran describe and summary commands on the dataset.

```
Describe Data
       no_of_employees      job_level           year         salary  \
count     2.793650e+05  279365.000000  279365.000000  279365.000000
mean      1.992288e+04       2.551426    2015.146586   88646.609885
std       5.044350e+05       1.047190       0.811143   31965.935855
min       0.000000e+00       1.000000    2014.000000   10400.000000
25%       1.700000e+02       2.000000    2014.000000   71074.000000
50%       1.634000e+03       2.000000    2015.000000   88254.000000
75%       1.080000e+04       4.000000    2016.000000  106288.000000
max       2.635506e+08       4.000000    2016.000000  885666.000000

          salary_log  no_of_employees_log
count  279365.000000        279365.000000
mean       11.313593             7.199814
std         0.431814             2.754739
min         9.249657             0.000000
25%        11.171491             5.141664
50%        11.387986             7.399398
```

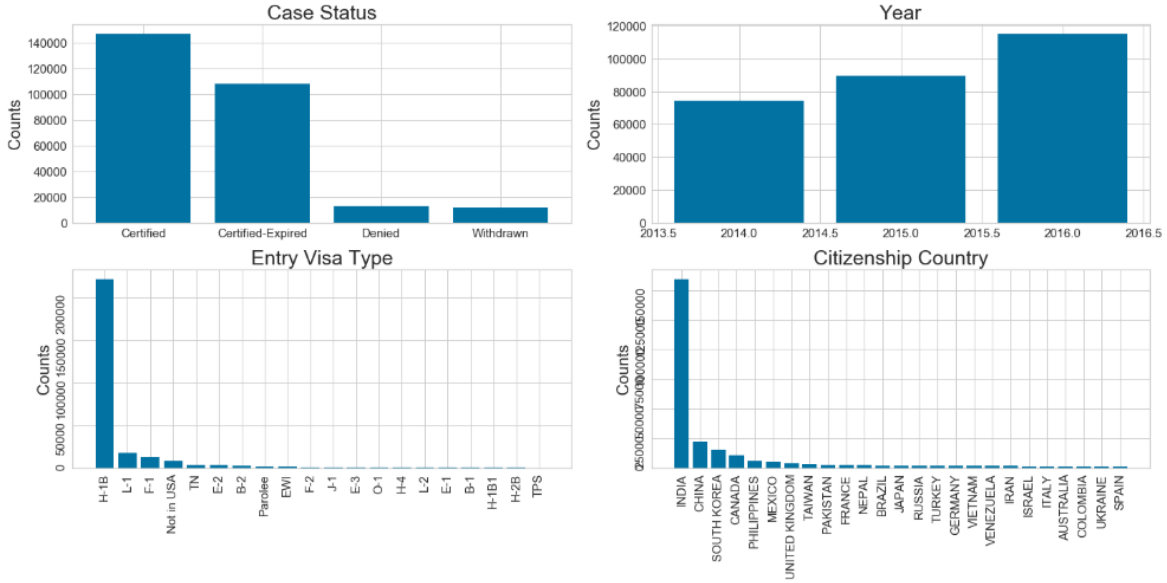
```
75%         11.573917              9.287394
max         13.694096             19.389756

Summarized Data
       case_status entry_visa citizenship   state
count       279365     279365      279365  279365
unique           4         54         197     112
top      Certified       H-1B       INDIA      CA
freq        147213     222234      159643   36454
```

5. As part of this step, I plotted some histograms to understand the data from different perspectives. Below are some of my observations from the histograms plotted.
   - Salary - Due to normalization, the graph looks more distributed now, removing the left skewness I had earlier. I have left the picture from original analysis as well to show the difference. Based on the initial analysis, as can think most of the applications seems to be around $100,000, so I didn't find any surprising findings here other than a small and interesting spike at $190,000.
   - Number of employees - Due to normalization, the graph looks more distributed now, removing the left skewness I had earlier. I have left the picture from original analysis as well to show the difference. Based on the initial analysis, I was definitely not expecting more than 1000 companies having around 50,000 employees so that is an interesting finding.
   - Job level - As H1B visa contributes to most of the permanent visa candidates that could be contributing here reflecting that level 4 candidates are more in number.
   - Year - this one is pretty straight forward, we had increasing number of cases over last few years, so this is in line with what I was expecting to see.

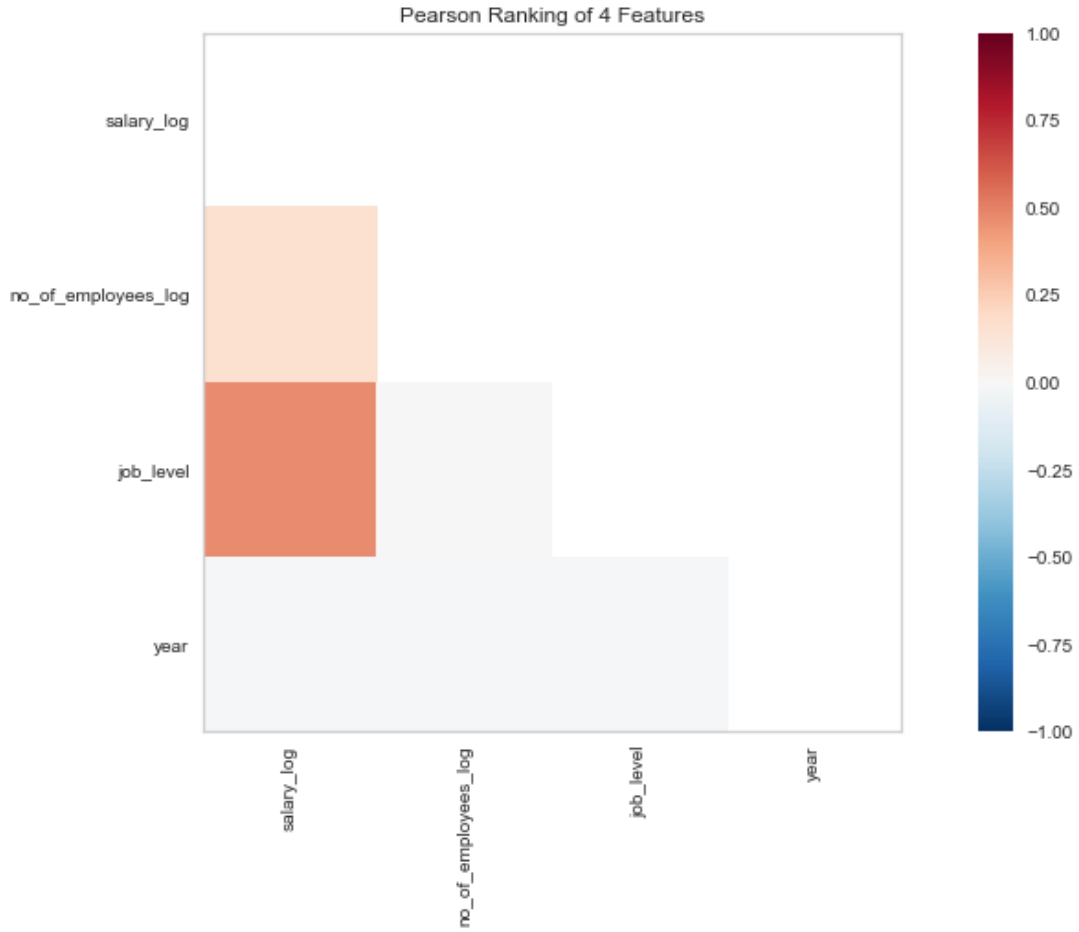
   Histogram charts including both raw and normalized values for salary & no_of_employees:



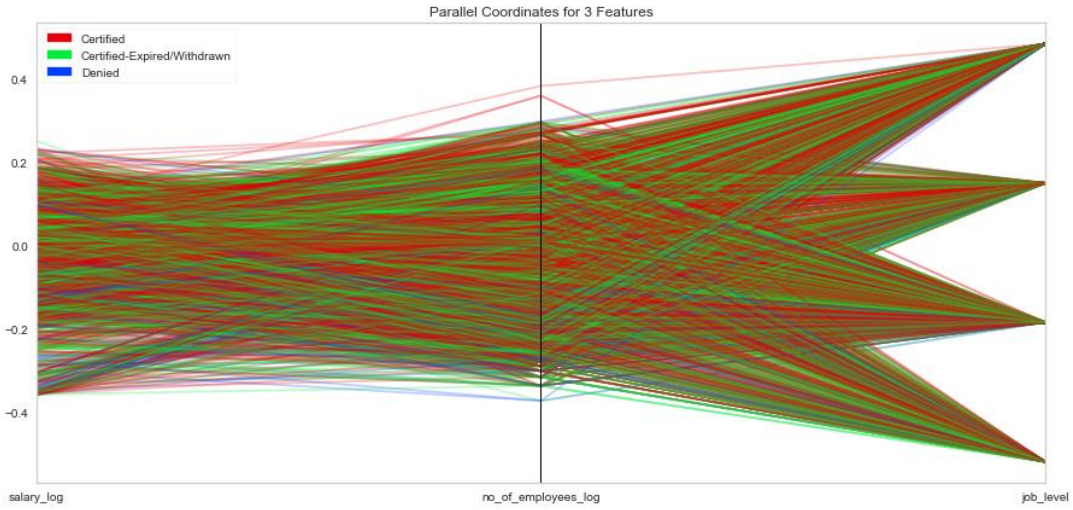
6. I have plotted bar charts using some of the other features, once again to gain understanding of the data from a different perspective. Below are my observations from the bar charts plotted.
    - Case status - I was expecting to see more number of certified or approved cases, which we see in the below chart, but what surprised me was the certified-expired, I was not expecting to see so many of the expired cases.
    - Year - As we have seen before with histogram, bar chart also shows increased number of cases by year.
    - Entry visa type - once again, the chart here meets my expectations and proves my understanding to be correct. Most of the cases are H1B cases.
    - Country - I knew India will be at the top of the list, but wasn't expecting this much difference with other countries. This is an interesting finding for me.



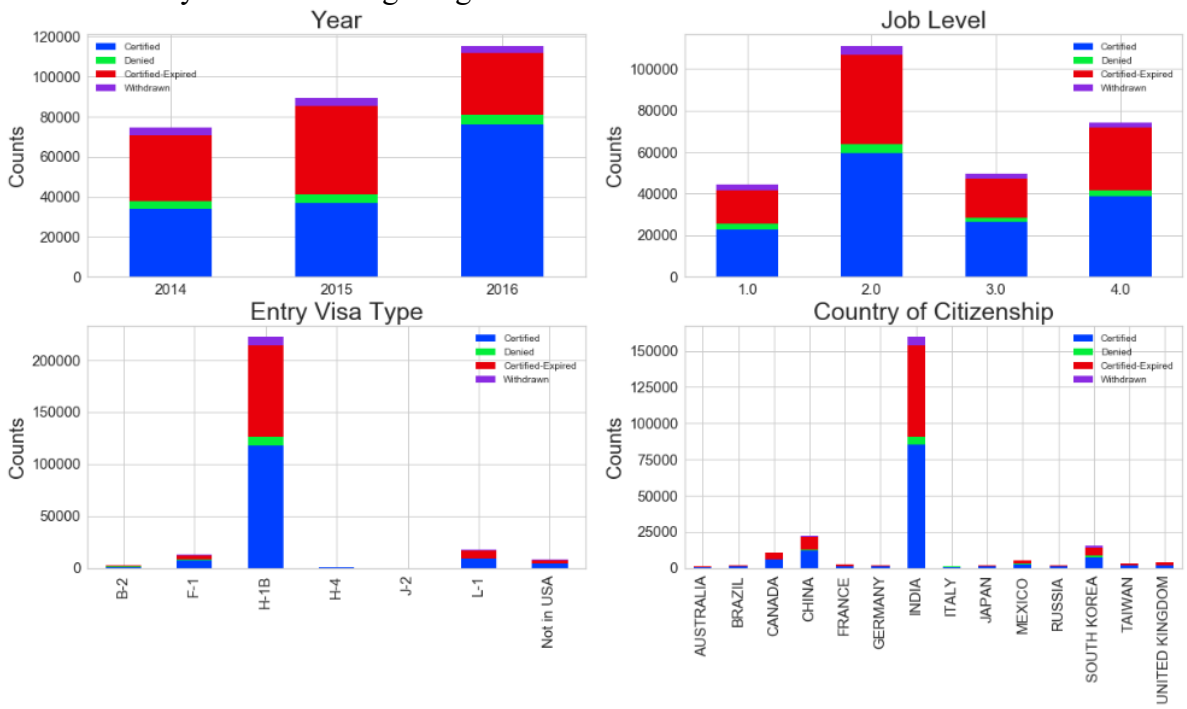
7. As part of this step I have used visualizer to find out the relationship between different features using Pearson ranking. Below are my observations:
    - I see high correlation between salary and no_of_employees.
    - Salary and year are positively related, that's what we always hope in reality to have increased wages as the year changes.
    - Salary and job level are positively related as well, which would make sense, a person with higher expertise would demand higher salary.
    - All of the remaining parameters are negatively related.
    - But, one of the thing surprised me was not having any strong relations between these fields.



8. As part of this step, I have compared several numeric parameters in the data using the parallel coordinates plot.
   - I did not get any meaningful or clear insights out of the parallel coordinates plot. Only thing we can see is with year, as the year increases we see stronger certified line and in the older years the number of certified-expired cases is high.
   - As part of part 2, I removed the year and arrived at below graph with normalized data for salary and no_of_employees.



9. As the next step, I have compared various features using the stacked bar chart with respect to case status counts for each feature. Below are my observations out of this step.
   - Year - It is interesting to see we have more approved cases in 2016 compared to 2015, by visual comparison.
   - Job level - One more surprising fact, once again for me is to see those many expired cases.
   - Entry visa - once again we can see H1B leading the chart at a very high margin from other types of visa.
   - Country - India is leading the chart here and seeing a good number of approved cases for all countries.
   - One common thing this chart clearly shows us is that most of the cases are getting approved, seems like very less cases are getting denied.



10. As the next step, I have converted few categorical features I have in the dataset to numerical using one hot technique.
    Below are the columns that are converted:
    entry_visa
    citizenship
    state
    As part of this step, I have also generated data_temp data frame that contains only the rows with either 'certified' or 'denied' rows. As the primary goal for me is to identify if a particular case would be either certified or denied.

```
Data before conversion:
    entry_visa citizenship          state
2         H-1B       INDIA       NEW YORK
4         H-1B       INDIA      WISCONSIN
7         H-1B       INDIA       NEW YORK
23        H-1B       INDIA       MICHIGAN
24        H-1B       INDIA     CALIFORNIA
26         E-3   AUSTRALIA  NORTH CAROLINA
34        H-1B       INDIA        GEORGIA
```

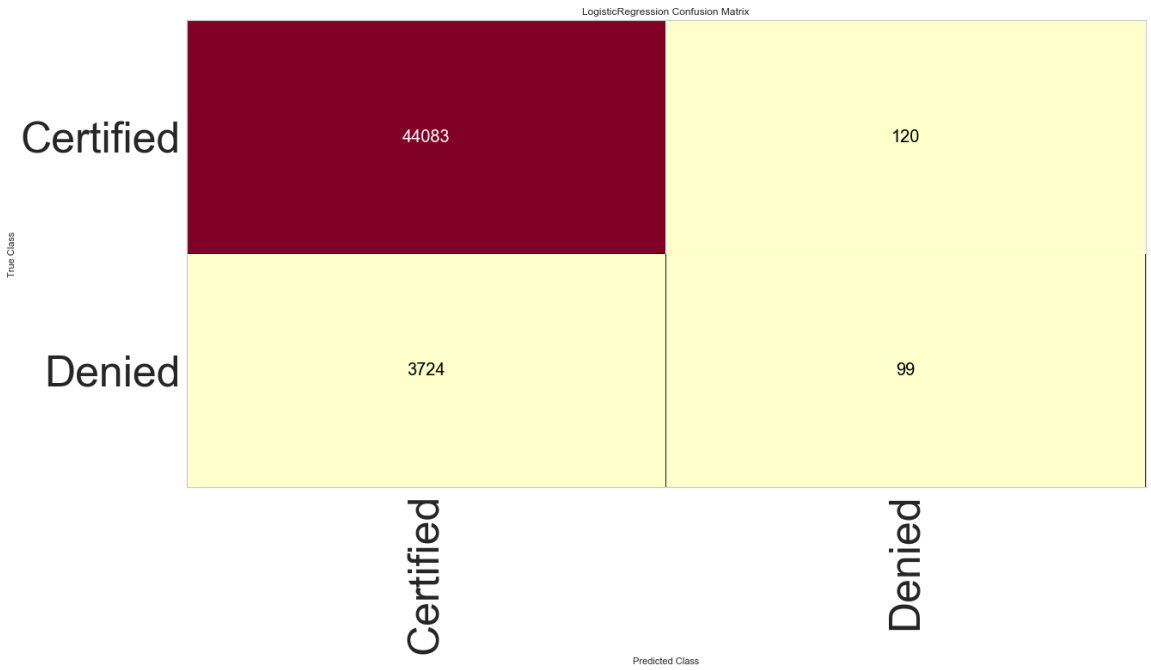
```
35       H-1B       INDIA        NEW YORK

Data after conversion:
     entry_visa_A-3  entry_visa_A1/A2  entry_visa_B-1  entry_visa_B-2  \
2                 0                 0               0               0
4                 0                 0               0               0
7                 0                 0               0               0
23                0                 0               0               0
24                0                 0               0               0
26                0                 0               0               0
34                0                 0               0               0
35                0                 0               0               0

    entry_visa_C-1  entry_visa_C-3  entry_visa_D-1  entry_visa_E-1  \
2                0               0               0               0
4                0               0               0               0
7                0               0               0               0
23               0               0               0               0
24               0               0               0               0
26               0               0               0               0
34               0               0               0               0
35               0               0               0               0

    entry_visa_E-2  entry_visa_E-3  ...  state_VIRGINIA  state_VT  state_
WA  \
2                0               0  ...               0         0
 0
4                0               0  ...               0         0
 0
7                0               0  ...               0         0
 0
23               0               0  ...               0         0
 0
24               0               0  ...               0         0
 0
26               0               1  ...               0         0
 0
34               0               0  ...               0         0
 0
35               0               0  ...               0         0
 0

    state_WASHINGTON  state_WEST VIRGINIA  state_WI  state_WISCONSIN  \
2                  0                    0         0                0
4                  0                    0         0                1
7                  0                    0         0                0
23                 0                    0         0                0
24                 0                    0         0                0
26                 0                    0         0                0
34                 0                    0         0                0
35                 0                    0         0                0

    state_WV  state_WY  state_WYOMING
2          0         0              0
4          0         0              0
7          0         0              0
```

```
23          0          0                0
24          0          0                0
26          0          0                0
34          0          0                0
35          0          0                0

[8 rows x 348 columns]
```

11. Create final feature datasets that can be used for train and validation.
    As part of this step I have combined the categorical variables converted to numbers with other features I have and generated the X and Y data frames needed for logistic regression model.
    I have also separated data frame into two sets, one for training the model and the other for testing the model.
    Below are the details from training and testing sets:

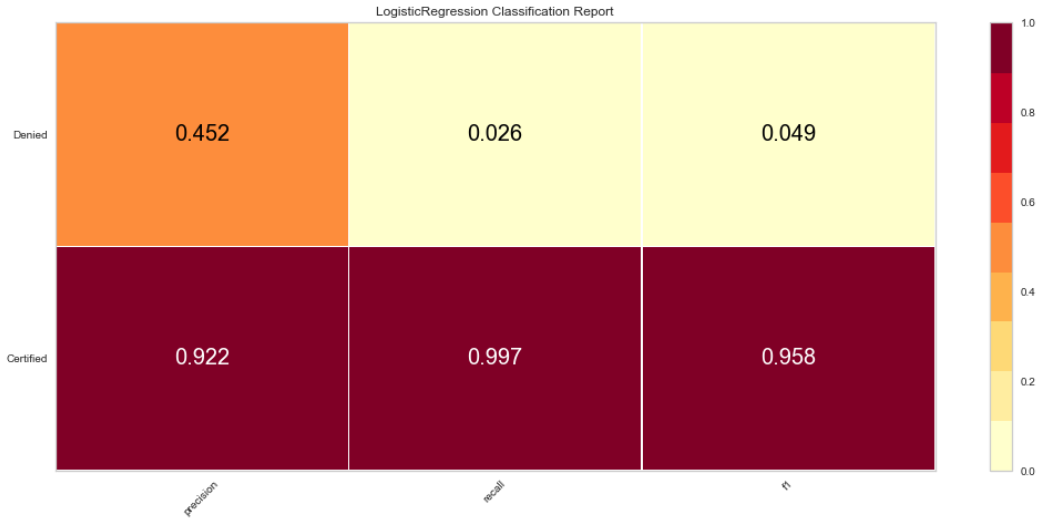
```
No. of samples in training set:  112060
No. of samples in validation set: 48026


Look at different case_status values in the training set:
Certified    103010
Denied         9050
Name: case_status, dtype: int64


Look at different case_status values in the validation set:
Certified    44203
Denied        3823
Name: case_status, dtype: int64
```

12. Create the logistic regression - as part of this step, I have created logistic regression model and ran several evaluations on the model to see how the model is performing.
    Confusion Matrix:
    - As you can see below, TP (True Positives) are high, but model failed to identify the denied cases accurately, only 99 cases (out of total 3823) denied cases were correctly predicted.



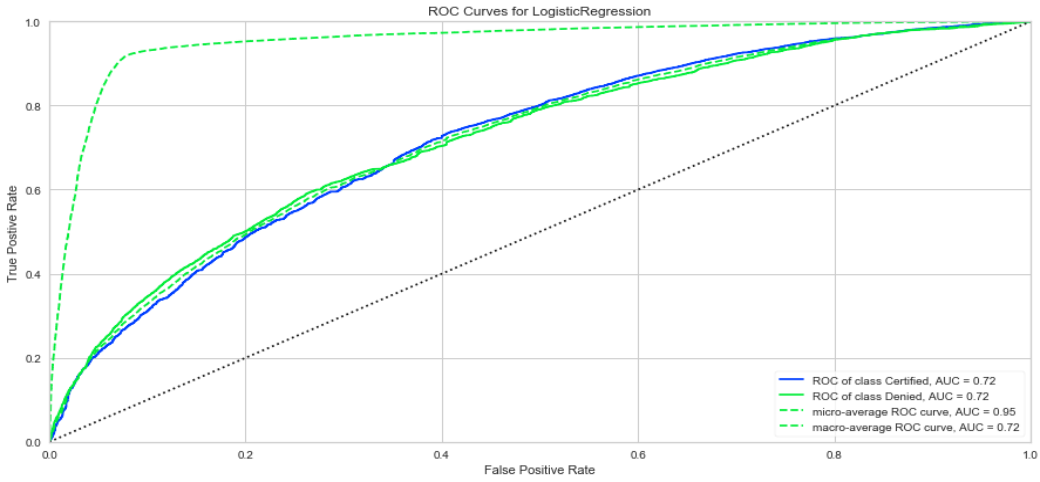
Logistic Regression Classification Report:

- Similar to what we have seen in confusion matrix, the other evaluation parameters show the poor performance of the model when it comes to denied class, as shown below.



ROC curves for Logistic Regression:

- ROC curves show a better performance of the model as all of the curves are above the dotted line, which is randomly guessed.



Conclusion:

The graph analysis on the US permanent visa applications dataset has given me very good insight into the dataset, helped me in understanding this dataset in different perspective. It also helped me to realize some interesting facts. One of such fact being the very high number of approved, but expired cases. Also one good thing I see out of this analysis is that there are very less number of denials.

As part of part 2 of case study, I learned couple of lessons that it is better to apply the normalization after we complete the graph analysis to understand the data and before we feed data into any models. I have also noticed that conversion of categorical features into numeric through one hot technique is probably not ideal when we have many possible values like in my case. So, depending on how I use this data as part of case study part 3, I will probably have to adopt a different technique.

As part of part 3 of case study, I have built a logistic regression model to predict if a US permanent visa will be granted based on provided data or not. Overall the model I have built seems to be predicting the certified cases well, but predicting too many of the denied cases as certified as well.