Data Wrangling

Link to Jupyter Notebook on Github:
https://github.com/sopaturi/Springboard/blob/master/Capstone%20Project/Capstone%20Project-Data%20Wrangling.ipynb

The first step to examining the dataset involves data wrangling or the process of converting the data into a more valuable format that can be used for analysis. The current dataset was obtained from https://www.foreignlaborcert.doleta.gov/performancedata.cfm. The dataset contains H1B-LCA application data from the 2017 fiscal year.

The data wrangling for this dataset can be split into and defined by these four steps: deletion of unnecessary columns, deletion of rows that contain missing values, deletion of rows with incorrect characters that do not represent the field accurately, and checking for outlier values.
First, the dataset needs to be uploaded into a pandas dataframe so that data wrangling can start. The dataset was originally not encoded in utf-8. In order for pandas to be able to read the .xlsx file, the file was converted to a csv file and encoded to utf-8.

The irrelevant columns that do not help us with exploratory data analysis or answering the overall question of predicting the likelihood of an accepted application will be removed. A justification is provided in the notebook for each field deletion so that columns with vital information are not accidentally taken out.

There are 624,650 rows in the dataset. A lot of these rows contain missing values. When all rows with any missing values are deleted, only 98,155 rows remain. Therefore, only the columns with absolutely crucial information for predicting H1B-LCA application status will have the entries for their corresponding missing values removed. There should not be critical fields with missing values in the final data frame because it will construe the data and make it harder to do analysis. Only columns with crucial information that affect H1B-LCA application outcome will have entries for their corresponding missing values removed in order to preserve a substantial row count for later analysis.

The columns considered to be crucial and ready to have the corresponding entries containing the null value in the column removed are the employer_name, employer_address, employer_city, job_title, soc_name, full_time_position, prevailing_wage, pw_unit_of_pay, wage_rate_of_pay_to, wage_unit_of_pay, H1B dependent, willful violator, support h1b, labor_con_agree, and worksite_city. The columns considered to be crucial are the columns that can possibly affect the outcome of the H1B-LCA application. These columns will have the entries containing null values of this column deleted.

Now that the entries with missing values in critical fields have been removed, values that are irregularities need to be replaced with more understandable notation. For example in the AGENT_ATTORNEY_NAME field, a comma is in many rows that do not have an attorney name.

The comma can be replaced with the value None so that it is easily identifiable if an application had an attorney or not.

Currently, the industry from which the job is in is represented by a NAICS_CODE. The NAICS code needs to be converted to an industry name. The code in the jupyter notebook retrieves the corresponding industry name from the formatted url and adds unique values to a dictionary. Unique values are first replaced in the dataframe with industry names then the duplicate values are populated. Replacing the numerical codes with industry names is more readable and easy to understand.

Next, the case status column will be examined in order to remove irrelevant rows. The case status column identifies whether or not the H1B-LCA application was certified, denied, withdrawn, or certified-withdraw. Because the goal is to predict the likelihood of an applicant receiving a certified status, the rows containing applications that were withdrawn can be removed as they give no insight on whether an application was certified or not.

A possible reason for the declined H1B-LCA application is the difference between the prevailing wage and the wage offered by the employer. If the wage difference is large, it may be reason for the application receiving a rejection because a lower wage than the prevailing wage is considered unfair. A calculated field for wage difference was created. The units for the prevailing wage and the offered wage may be in different units such as monthly, hourly, and yearly. The units for wage need to be converted so they are identical for each application. For example if prevailing wage is in yearly units but the wage offered is in monthly, it needs to be converted to yearly.

Another aspect of data wrangling involves finding the outliers in a field that may lead to faulty analysis later on. In order to make sure the prevailing wage column and the wage rate offered by the employer contain values that are valid, a scatter plot can be constructed to make sure the salaries fall within a reasonable range.

Python lists for graphing wage offered vs. prevailing wage by different units of time were created so that a scatter plot could be made.

It is clear from the scatter plot describing wage offered by employer vs prevailing wage that there are outlier values in the prevailing wage axis. There are some entries with a prevailing wage greater than 50 million. These are not correct prevailing wages and should be removed from the dataset. However, when looking at the case status of these applications with a prevailing wage greater than 50 million below, it is seen that all the application are denied. It can be hypothesized that an unrealistic prevailing wage listed can lead to an application denial. Removing these rows would remove rows that received a denial status and make the dataset even more unbalanced towards the certified entries. Therefore, the outliers are kept because prevailing wage may be a large factor in the application case status.

After these data wrangling steps, the dataset should be ready for exploratory data analysis and solving the main problem of predicting the likelihood of whether an H1B-LCA application will be approved.