***IST 707 Applied Machine Learning***

**Classification Of Fake/Real Jobs And Job Recommendation System**

**Dishank Solanki**

***dsolanki@syr.edu***

**Classification Of Fake/Real Jobs And Job Recommendation System**

Final Project Report: Spring 2020, IST 707 Applied Machine Learning

Author: Dishank Solanki Under Supervision Of Prof. Yang Yang

1. **Introduction:**

Every graduate student is excited to start their first job right after their college. And if one doesn’t get a job soon, then they become desparate to start the job. Ad student debts add extra burden on the students to start the job as soon as possible. According to CNBC job scams in 2018 had doubled as compared to 2017 [1]. Not only graduates, but situation like COVID pandemic also creates the environment where lot of individuals have to loose their job. All this factors create desperation in an individual to look for new jobs. Because of this, individuals don’t do much of background check on job postings. This creates a perfect opportunity for job scammer who take advantage of this situations and scam individuals. My own friend who in her last semester for her master’s study had become victim of such a scam. And when I came across this dataset about job scam classification, I found it very relatable. What if job portals like Linkedin, Indeed etc. come up with algorithms to classify the job whether they are scams or not and add an red alert or something which notifies the users that this job posting can be an scam and apply for this job on your own risk. This will greatly reduce job scams.

I have three important sections in my project and they are exploratory data analysis to understand the behavior of job scammers discussed in section 5, Machine Learning Models to build a real-fake job classifier discussed in section 6 and content based job recommendation system discussed in section 7. But before starting analysis, I have pre – processed the dataset which is discussed in section 4. And at the end, I have listed my conclusions. Let’s start our project discussion with objective of the project and data description of dataset used.

1. **Objective:**

The main objective for this project is to build the models that classifies the job posting as fake or real. And then, recommend the jobs to the user using the dataset. I found this dataset [2] from Kaggle. I decided to use Logistic Regression, Naïve Bayes, Decision Tree, Random Forest and SVM to classify the job postings. Also, try to understand various parameters that job scammers target via exploratory data analysis. Using cosine similarity, I will try to build recommendation system to recommend the jobs to the user by capturing the inputs like job title, job type, function and country of interest from them.

1. **Data Description:**

The dataset was taken from Kaggle. The dataset consist of 17780 data rows and 18 attributes. Out of which two are binary columns which are has\_company\_logo and has\_questions. Two are numeric that is job\_id and salary range. Target variable is Fraudulent which is binary (1 = job is fraudulent, 0 = job is real). Rest 13 are textual columns.

1. **Data Cleaning & Pre-processing:**

We check for %missing data in each column. So, two attributes, department and salary\_range has more that 50% of missing columns. And this kind of attributes do not contribute much to data visualization or for classification models. So, we drop those columns. Then we drop data rows among dataset which has about 50% data missing. This leaves us with 16712 data rows and 15 independent attributes. We split the location attribute to extract the country name from it.

1. **Preprocessing for exploratory data analysis:** Our data set still has missing values. So for Data visualization, we substitute the missing values with “Not Mentioned”.
2. **Pre-processing for classification models:** For classification models, we replace missing values with blank space that is “ “. Also, the dataset is highly imbalance that is number of non-fraudulent jobs are 15939 and fraudulent jobs are just 773. So to remove the unbalance from our dataset, we randomly select 1227 rows from non-fraudulent data so as to make our dataset of 2000 data rows and 15 attributes. The textual columns(11 attributes) were merged together and then were tokenized, stop words were removed and then were normalized using TF-IDF method. Then standard scaler (standardizes features by removing mean and scaling to unit variance [3]) was used for scaling the vectorized attributes for all the classification models except Naïve bayes where we used Min Max Scaler because the attribute values are required to be between 0 and 1 for naïve bayes algorithm. The we split our dataset into training and testing dataset. 25% dataset is testing and 75% dataset is training.
3. **Pre-processing for recommendation system:** Unwanted columns 'company\_profile', 'description', 'requirements', 'benefits', ‘telecommuting' ,'has\_company\_logo', 'has\_questions' ,'department','salary\_range' are dropped. Country name is extracted from Location column and then Location column is dropped as well. Missing values are replaced by “ “. Remaining columns are merged and then vectorized using TFIDF.
4. **Exploratory Data Analysis:**

Initially all the visualization contained 16712 rows. Out of which employment\_type has 13%, required\_education contains 38.8%, required experience contains 32.9%, industry & Function columns contains 21% and 29% missing values which were not considered for visualization.

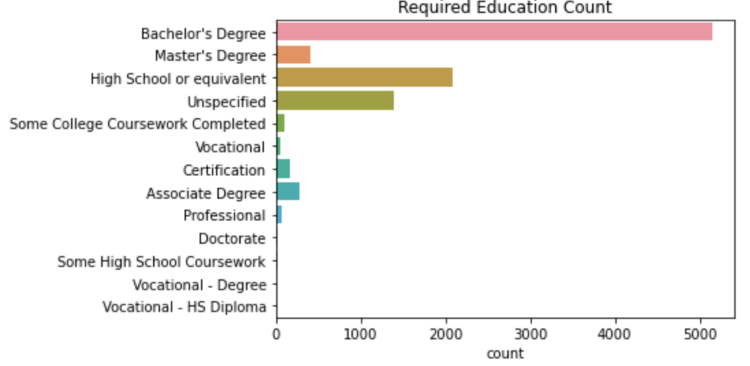
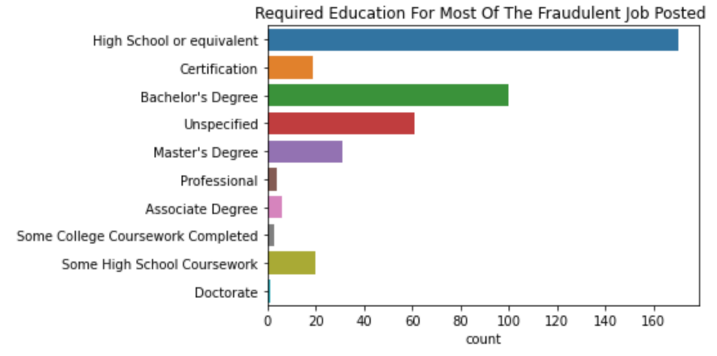


Fig 5.1 Min Required Education In Every Job Posted Fig 5.2 Min Required Education In Fraudulent Jobs

As we can see from the title of the images, Fig 5.1 represents the required education for job posted. And fig.5.2 represents the required education from fraudulent data. We can see that normally, most of the jobs requirement mention at least bachelor’s degree. But for fraudulent jobs, only high school or equivalent education is required.

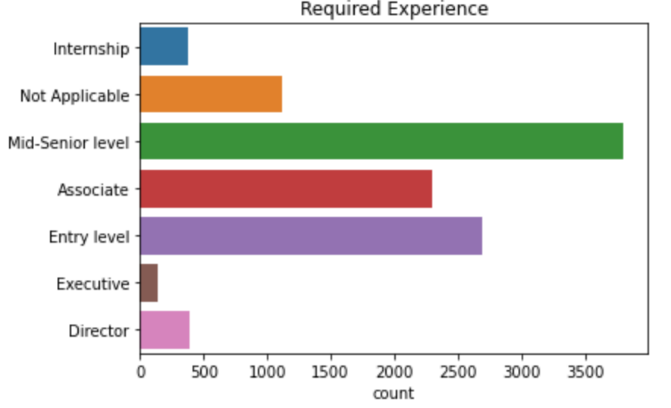
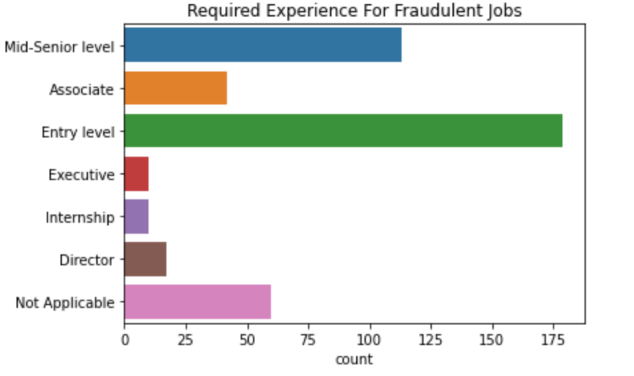


Fig 5.3 Jobs Posted For Each Designation Level Fig 5.4 Fraudulent Jobs For Each Designation Level

Image Fig 5.3 represents the required experience for job posted. And fig.5.4 represents the required experience from fraudulent data. We can see that normally, most of the jobs requirement mention at least mid – senior level experience. But most of the fraudulent jobs are entry-level.

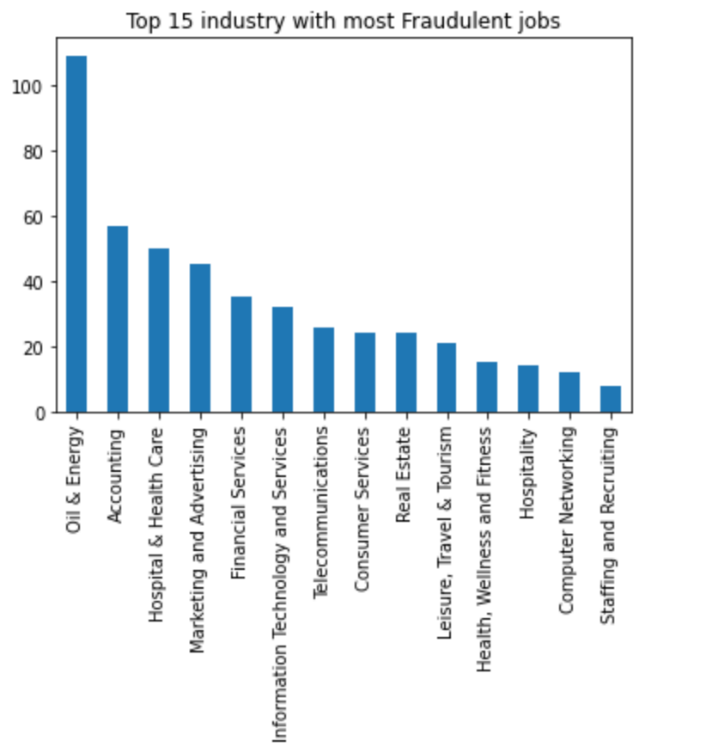
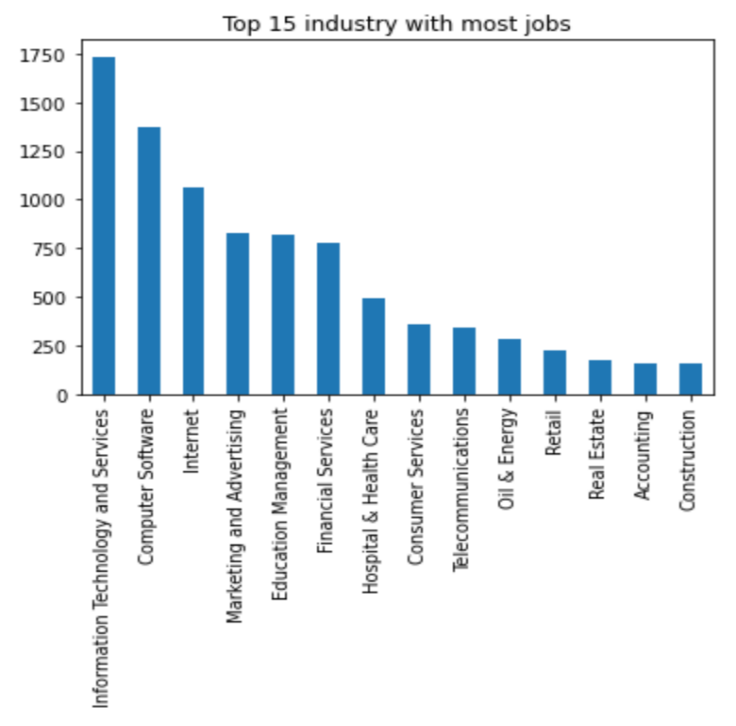


Fig 5.5 Job Postings For Top 10 Industry Fig 5.6 Fraudulent Job Postings For Top 10 Industry

Image Fig 5.5 represents the industry with most job posted. And fig.5.6 represents the industry with most fraudulent job posted. We can see that normally, most of the jobs are from IT & Services industry. But most of the fraudulent jobs are from Oil & Energy industry.

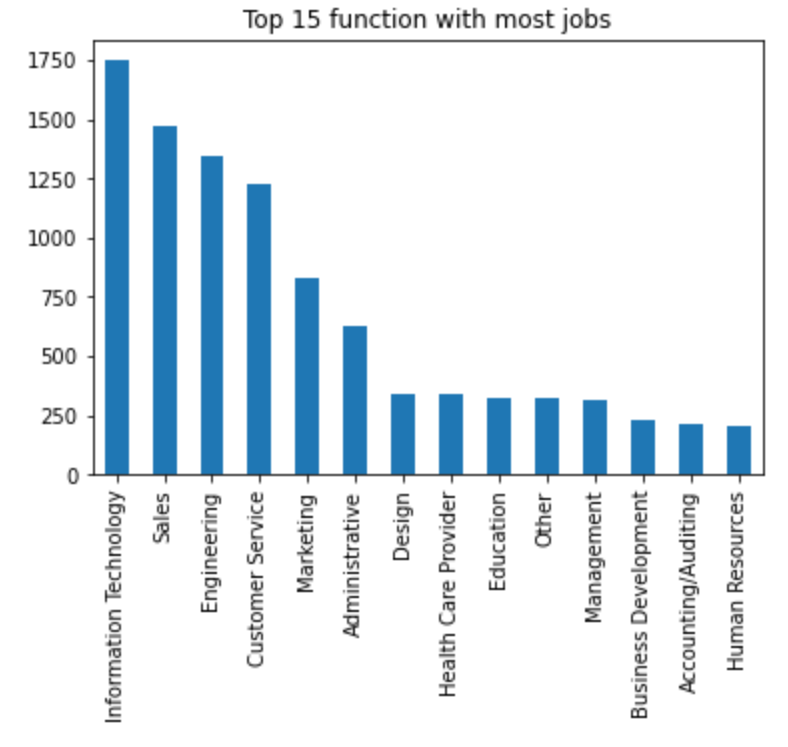
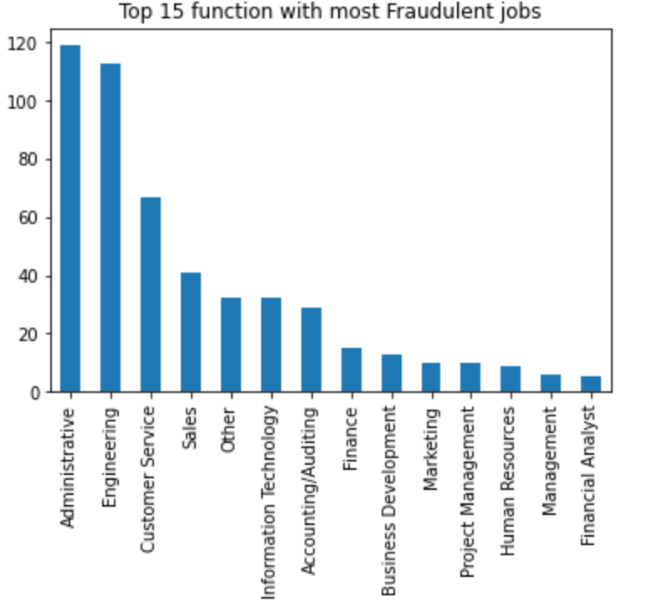


Fig 5.7 Job Postings For Top 10 Function Fig 5.8 Fraudulent Job Postings For Top 10 Function

Image Fig 5.7 represents the industry with most job posted. And fig.5.8 represents the industry with most fraudulent job posted. We can see that normally, most of the jobs are from IT division. But most of the fraudulent jobs are from Administrative function.

1. **ML Models & It’s Performance:**

I have used GridSearchCv to find out the best parameters for best performance in each of the below mentioned models. Accuracy is used as parameter to find the best parameters for classification models. But AUC will be used to evaluate and compare the performance of each model because though we have removed the bias from our dataset, still our dataset is unbalanced to some extent.

1. **Logistic Regression:** This classification model normally performs well for binary classification and it is faster as compared to other models. Following parameters and their values were used to hypertune the Logistic Regression model:

CV = 7, C : [0.1, 0.5, 1, 10, 15, 20], penalty:['l1','l2']. L1 and L2 penalty are used to avoid overfitting in Logistic Regresion. C is the inverse of regularization strength.

From GridSearchCv, best performing parameters based on accuracy are c = 2 and penalty = L2. Following is the performance of Logistic regression on testing dataset:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Predicted Class** | | |
| **Actual**  **Class** |  | Fraudulent **=** 0 | Fraudulent **=** 1 |
| Fraudulent **=** 0 | 302 | 22 |
| Fraudulent **=** 1 | 2 | 174 |

We can see that recall for this model is very high that is 0.99. Accuracy and AUC for this model are 0.95 and 0.98 respectively.

1. **Naïve Bayes:** Naïve Bayes considers that all the attributes are independent. We use Multinomial Naïve Bayes because its performance is better when it comes to textual attributes and classification. I have used GridSearchCv to hypertune Multinomial Naïve Bayes with CV = 7. Alpha = 0.0001 is the best parameter giving best accuracy of 0.92 on training data. Following confusion matrix will help to better analyse the performance of this model on testing dataset:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Predicted Class** | | |
| **Actual**  **Class** |  | Fraudulent **=** 0 | Fraudulent **=** 1 |
| Fraudulent **=** 0 | 294 | 38 |
| Fraudulent **=** 1 | 10 | 158 |

We can see that Recall and Precision both are very good for this model that is 0.94 and 0.81 respectively. Accuracy and AUC are bit less as compared to Logistic Regression that is 0.90 and 0.96.

1. **Decision Tree:** Using Gridsearchcv with CV = 7, following are the parameters of the best performing models:

criterion = entropy, max\_depth = 6, max\_leaf\_nodes = 9

Following confusion matrix will help to better analyse the performance of this model on testing dataset:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Predicted Class** | | |
| **Actual**  **Class** |  | Fraudulent **=** 0 | Fraudulent **=** 1 |
| Fraudulent **=** 0 | 247 | 49 |
| Fraudulent **=** 1 | 57 | 147 |

The precision and recall for this model is not very high but it is decent that is 0.75 and 0.72. The accuracy and AUC for this model is 0.79 and 0.84.

1. **Random Forest:** Random Forest is considered to have better performance because it usually avoids over-fitting by considering average of every trees. Generally, it performs very well for multi – class classification. I have used GridSearchCv to hypertune random forest model with cv = 7. Following are the best model based on accuracy:

'bootstrap': False, 'max\_depth': 20, 'min\_samples\_leaf': 5, 'min\_samples\_split': 10. Following confusion matrix will help to better analyse the performance of this model on testing dataset:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Predicted Class** | | |
| **Actual**  **Class** |  | Fraudulent **=** 0 | Fraudulent **=** 1 |
| Fraudulent **=** 0 | 297 | 50 |
| Fraudulent **=** 1 | 7 | 146 |

We can see that Recall is very high that is 0.95 and Precision is good but less than recall that is 0.74. This indicates that our model classifies very well but it also classifies some data of Non-fraudulent jobs as Fraudulent. Accuracy and AUC are almost same as Naïve Bayes that is 0.89 and 0.97. But time taken by Random Forest is more as compared to Naïve Bayes and Logistic Regression.

1. **SVM:** Generally, its considered that SVM will have a better performance. But time taken to fit the SVM model is very high. Following are the hypertuned parameters with best performance in Gridsearchcv with cv = 7:

C = 1, gamma = 0.1, kernel = linear, probability = True

Following confusion matrix will help to better analyse the performance of this model on testing dataset:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Predicted Class** | | |
| **Actual**  **Class** |  | Fraudulent **=** 0 | Fraudulent **=** 1 |
| Fraudulent **=** 0 | 297 | 22 |
| Fraudulent **=** 1 | 7 | 174 |

The recall and precision are very high that is 0.89 and 0.96. The accuracy and AUC are 0.94 and 0.98. But the main disadvantage of this model is time. It took almost 4-5 hours to train the model.

1. **Job Recommendation System:**

I created job recommender system using content based. I will take information about some key parameters from users and based on that features, I will try to recommend the best job postings to the user.

Inputs are taken from user for attributes like Location, Employment\_type, title and function. Out of which Location and employment\_type values are used to filter the dataset. Job Title and function value are merged together and then it is inserted into the job recommender dataset.

Job recommender dataset is created by merging all the textual columns. Whole dataset is vectorized using TF-IDF algorithm. Then cosine similarity is calculated between the user input data and all the rows present in the dataset and based on top 10 cosine similarity values, jobs are recommended to user.

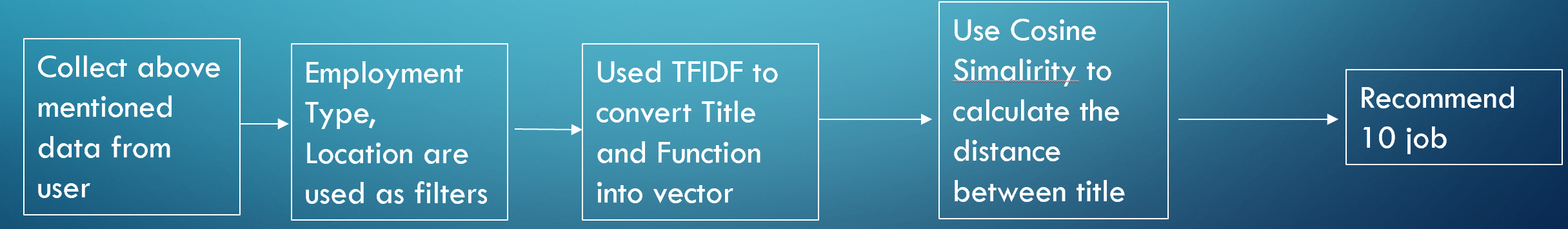


Fig 7.1 Block Diagram For Recommendation System

Fig 7.1 is the block diagram for the logic used to build the recommendation system. Before discussing the results of test data, lets understand basic of Cosine Similarity first.

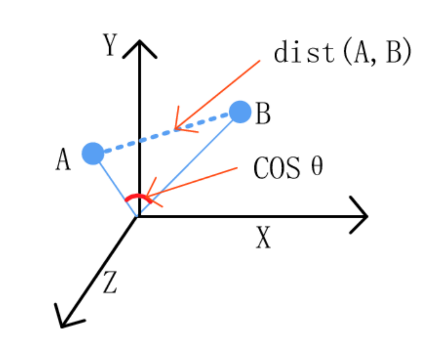
 **Cosine Similarity:** Input objects are expressed as two vectors. Then based on angle between the two vectors, similarity value is calculated. Similarity is nothing but COS( Angle between two vectors.) If the angle between the two vectors is 0 that is they are parallel in space that is they are similar then COS(0) = 1 which means they are similar. If two vectors are perpendicular to each other then COS(90) = 0 which means that they are not same. So the similarity values lie between 0 and 1 [4].

Fig 7.2 Cosine Similarity between two objects

Following data was used to check the recommendation system:

Country Name = United States, Enter Employment Type = Full-Time, Enter Title = Software, Enter Function = Engineering.

|  |  |
| --- | --- |
| Index | Similarity |
| 2400 | 0.61 |
| 1559 | 0.57 |
| 6023 | 0.52 |
| 5598 | 0.45 |
| 2843 | 0.43 |
| 4498 | 0.33 |
| 2996 | 0.33 |
| 809 | 0.31 |
| 352 | 0.30 |

Table on the left shows the cosine similarities value for the top 10 recommendations based on user input. We can see that the values of cosine similarities are not that great because of which the recommendation will also be not that good.

Table 7.1 Top 10 Cosine similarity values

Following is the screenshot of the recommendations provided by recommender system:

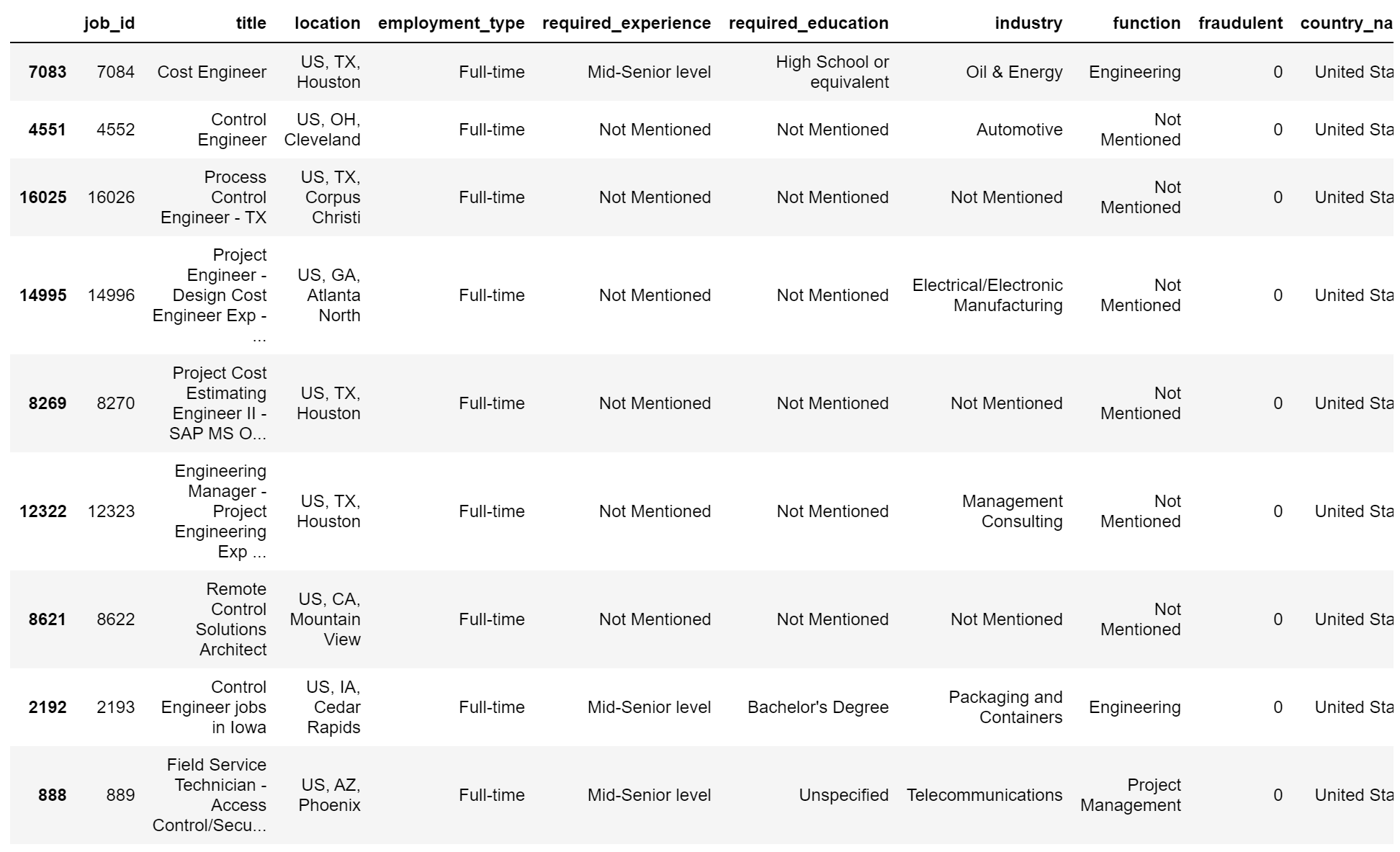


Fig 7.3 Tested Job Recommendations

1. **Conclusion:**

**Visualizations:** Following are the conclusions from the visualizations in section 5:

1. Fraudulent jobs normally tend to keep low education qualification as compared to real job postings. This is because they want to attract larger audiences and people who do not have high education but still who are looking for high paying jobs.
2. Fraudulent jobs scammers normally target entry level people. One of the reason for this is new graduates who are looking for new jobs are initially excited as well as desperate to start a new job. So they are more vulnerable to job scams.
3. Even though most jobs were posted for Information & Technology Services and IT department, most fraudulent job postings were from Oil industry and administrative function.

**ML Models:** Almost all the models had very good performance. The main reason is unbalance nature of the dataset was removed. But based on AUC from section 6, Logistic Regression, Naïve Bayes, Random Forest and SVM had the best AUC performances. But SVM and Naïve Bayes both had better Recall and Precision values which means that they were able to classify the job posting better. But SVM took a lot of time to train. So solely based on my dataset, I think that Naïve Bayes algorithm is enough to build a model to classify fake and real jobs.

**Recommendation System:** The example discussed in section 7 did not had good recommendations because the data that I have at my disposal is not enough. Also, I choose to filter the dataset based on 2 attributes, Location and employment type because my dataset is small. But dataset grows in real scenario then this might not be a good option. Different type of recommendation system like collaborative recommendation system, filtering recommendation system etc. can be used.

1. **References:**

[1] https://towardsdatascience.com/fake-job-predictor-a168a315d866

[2] https://www.kaggle.com/shivamb/real-or-fake-fake-jobposting-prediction

[3] <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html>

[4] https://www.analyticsvidhya.com/blog/2020/08/recommendation-system-k-nearest-neighbors/