TextIntelligence

Yiting Xiao
Jian Tian
Yongshuai Wang

Introduction

In this project, we proposed to create an intelligent career path recommender system. This system is intelligent in two main aspects. First, we build a hybrid machine learning model to give users the customized job recommendations. Second, we look deep into the job position data, and present key trends of the job market and the constantly changing need of talents in different industries. By simply uploading the resume of a user, we will show comprehensive and accurate career guidance.

It is without saying that, by recommending jobs that match the users' background, we are able to help them target the most relevant jobs and industries that they might be highly sought after. Also, our system can help employers fill the positions with most qualified candidates. With the help of big data and our machine learning system, we have faith in providing our users with highly relevant information based on their resume information. Additionally, advanced visual techniques can also help users better understand some statistical information related to a job position or even an industry.

In the following sections, we will introduce our machine learning algorithms, from bag of words feature extraction and dimensionality reduction technologies to state of art supervised classification model. We will also explain how we present the relevant records by data visualization, to show the information beyond numbers.

Problem Definition

Nowadays, job searching has become more and more convenient through the Internet. However, the massive job posts available online has make it a time-consuming task to target an ideal job. Most current job search engines only return a list of related jobs based on the keywords a user type in. Even though some websites do provide personalized recommendation to some extent, no websites specifically target data savvy professionals to assess whether or not resume holders is a good fit for data science, and recommend them to enter a specific industry.Moreover, the lack of data visualization techniques also makes it harder for users to have intuitive grasp of key information they want. Therefore, in this project, we will train machine learning models based on massive job posts data to provide career path recommendations: to assess resume holders whether or not it is a good choice for them to enter data science field, as well as what specific industry they should apply their talent. We will also visualize the statistical information in an artsy way, instead of presenting plain numbers.

Survey

Dimension Reduction: Truncated Singular Value Decomposition[1]

In text mining, one of the most widely deployed dimension reduction technique is latent semantic analysis. The mathematical techniques utilized in latent semantic analysis is singular value decomposition (SVD): Am * n =Um * z Σz * z (Vn * z)T.

The diagonal matrix Σ contains the singular values sorted in descending order. Each singular value represents the amount of variance captured by a particular dimension. The left-singular and right- singular vector linked to the highest singular value represent the most important dimension in the data. For example, if we want to retain only the first K dimensions of the data, we need to strip off the N-K dimensions of singular values and singular vectors. The reconstructed matrix will be the best possible least square approximation of the original matrix in the lower dimensional space, which is much smaller than the original matrix. The newly constructed matrix is also able to describe the data in terms of its principle components, but with much smaller dimensions. After applying SVD to our matrix, we got a new TF-IDF matrix which is ready for further analysis.

Multi-class Support Vector Machines[2]
Multiclass SVM aims to assign labels to instances by using support vector machines, where the labels are drawn from a finite set of features. The dominant approach for doing so is to reduce the single multiclass problem into multiple binary classification problems. There are mainly three ways of combining binary classifiers into multi-class classifier: “one against one”, “one against all”, directed acyclic graph SVM (DAGSVM). Empirical results show that “one against one” and DAGSVM approaches are superior to other approaches. In our project, we will use ‘one against one’ approach to combine binary SVM into multi-class SVM.

Random Forest[3]
Random forest simply build a set of trees and then use boosted aggregation strategy to combine many weak tree learners to form a strong learner. The mechanism for building trees in the forest is to randomly select features at each step of splitting the data and then randomly select two instances from the dataset and then use the average feature value as splitting value to build the trees. Although the tree seems to be built on a random basis, by query every tree in the forest and use the majority vote or mean value of the query result of all trees as classification result, the out of sample classification accuracy has been proved to be quite high.

Proposed Methods

Intuition
We design and implemented a multi-label version of cutler’s random forest algorithm, which add another layer of randomization in order to reflect the nature of multilabel instances as much as possible. Traditional random forest model only randomize the selection of training data and/or the selection of features, however, when the instance has a set of labels, we think traditional random forest model will lost information if our strategy is to assign each instance randomly selected label from the instance’s original label set and then use this data as training set to randomly select a portion of training data to build trees. Our model not only randomly select 80% of the total training data every time it build a tree, it also randomly select a label from the original label set of each training instance, in this way, because we build multiple trees, in our case 30 trees, the choice of label for each instance approximately reflect the multi-label nature

of the instance. So it should perform better than traditional random forest algorithm that does not reflect the multi-label nature of the instance.

Detailed description of approaches

TF-IDF: Text representation through vector space model

In order to effectively compare job posts, we need to transform each job post into a bag of words first. The careerbuilder API that we used to extract job information offers mainly three sections of information that we deem useful: job description, job requirement, job skills. Therefore, attributes like programming skills, industry specific knowledge and other soft skills such as personality and leadership capability are the information we want to retain as key words in our “bag of words”.

Therefore, data preprocessing with tokenizing, punctuation removal, long meaningless words removal and stemming are the techniques we performed on each job posts we analyzed. In the end, each job post is a python list of stemmers, and we store all the job post keyword lists into a huge python list.

With these python list of stemmers, we then formed a sparse matrix with each row representing a job post and each column representing a feature (or keyword) in all job posts. This is vector space model of representing text. In order to weight the importance of each word in this matrix, we utilized TF-IDF metric, which takes both the frequency of each word appearing in each job post as well as the times the word appears across different job posts into considerations. A word achieves high TF-IDF if it has high frequency of appearance in a job post and has low frequency of appearance across job post sets. One can imagine how big this sparse matrix could be, especially if we process large amount of job post.Therefore, we think it is necessary to apply dimension techniques to reduce the dimensions of the TF-IDF matrix.

As has been explained in the survey section, truncated singular value decomposition is a mathematical technique of conducting latent semantic analysis in text mining. Because our feature set is very large, it is well known the curse of dimensionality will make multiclass classifier less effective at predicting labels of unknown instance. Therefore, We will conduct further exploratory analysis to show how the number features will impact the accuracy of classification, then choose a dimension that balances the accuracy of classification and running speed. This is easily understandable because the higher the dimension, the larger the TF-IDF matrix, and the longer it takes to get the correct result.

Multi-label random forest algorithm

As has been mentioned from the intuition part, the major innovation of our project is to modify original version of random forest based on Cutler’s PERT algorithm. Each time a tree is built, a

random 80% of the total training data is used, if the instance has more than one labels, the training label of those 80% of training data is randomly selected from the label set of the original instance, otherwise it will simply use the only label of the instance. Tree was built based on randomly selected feature to split on and the split value of that feature is calculated by randomly selecting two instances and computing the mean of the feature values of those two instances. Tree stopped growing when the feature values of all instances are equal, or there is only one instance so that it cannot be further divided.

Because the label of our instance is categorical, testing label of testing instance is given by the majority vote of all query results of trees in the forest.

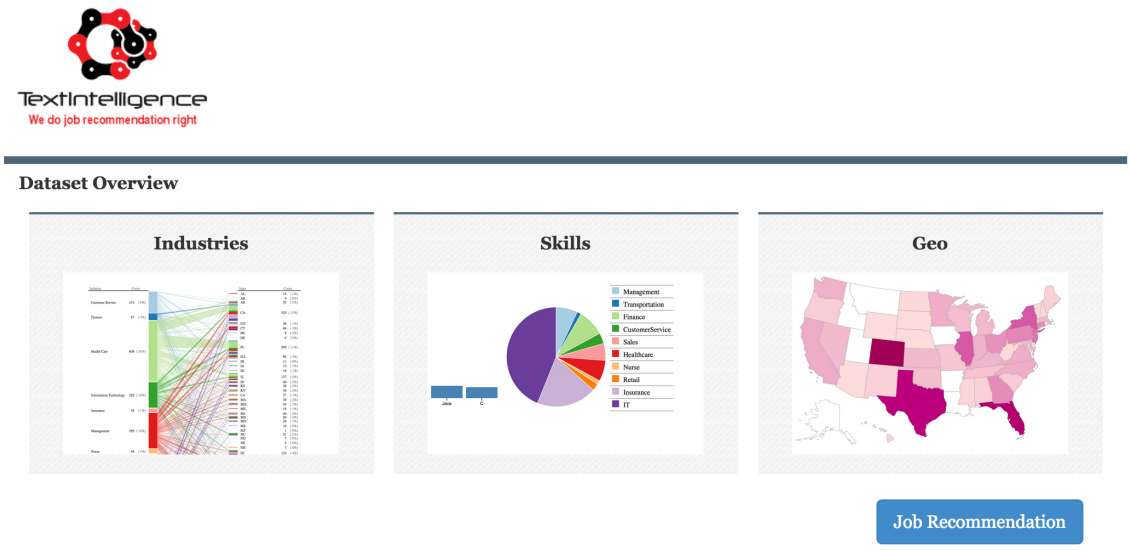
The original version of random forest we used to compare simply chooses a label for each instance from the instance's label set in the beginning, and does not randomize the label at each iteration of tree building process. All else is same with multilabel version of random forest.

We also compare our approach with multi-label linear support vector classifier using One vs Rest strategy. The strategy fit one classifier per class. For each classifier, the class is fitted against all the other classes. It has high computational efficiency and interpretability. Since each class is represented by one and one classifier only, it is possible to gain understanding about the class by inspecting its corresponding classifier. This is the most commonly used strategy and is a fair default choice.

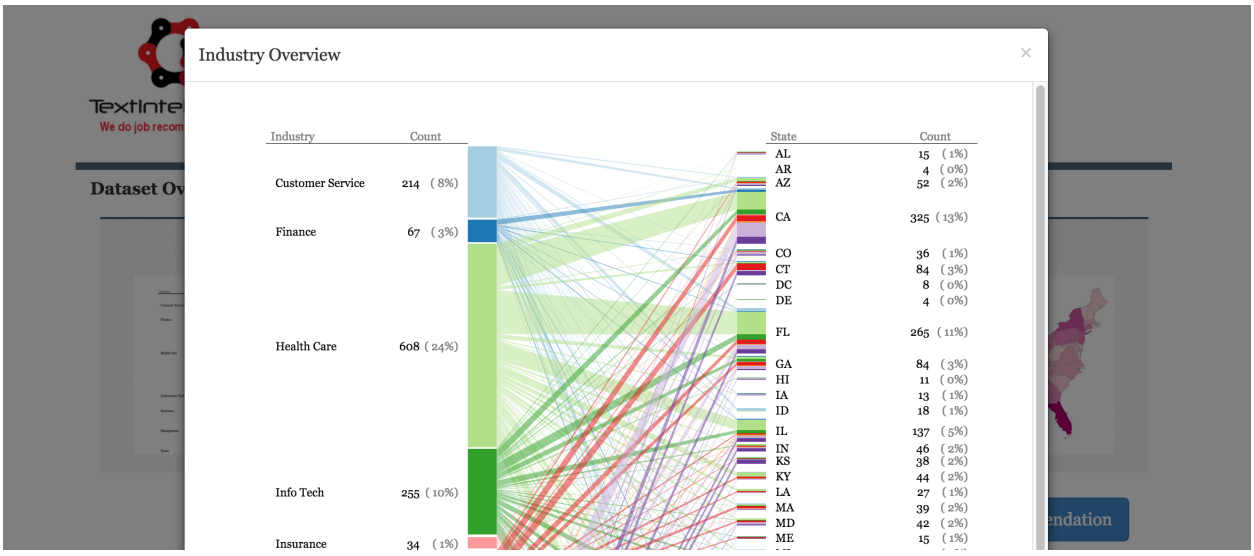
In the industry recommendation stage, because the ground truth of instance label is a set of labels, so our criterion of correctly classifying instance also modified somewhat. Our criterion now becomes as long as the predicted label is contained in the original label set, we will count this classification as accurate classification. Therefore, for the purpose of industry recommendation, our multi label random forest algorithm can be applied; for the purpose of assessing whether or not a resume holder is suitable to become a data scientist, because the original instance only has binary labels: has or not has the potential to be a data scientist, the multi label random forest is reduced to traditional random forest. Thus we will only compare the performance of multi-label random forest with traditional random forest based on industry labels (multilabel).

Detailed description of user interfaces

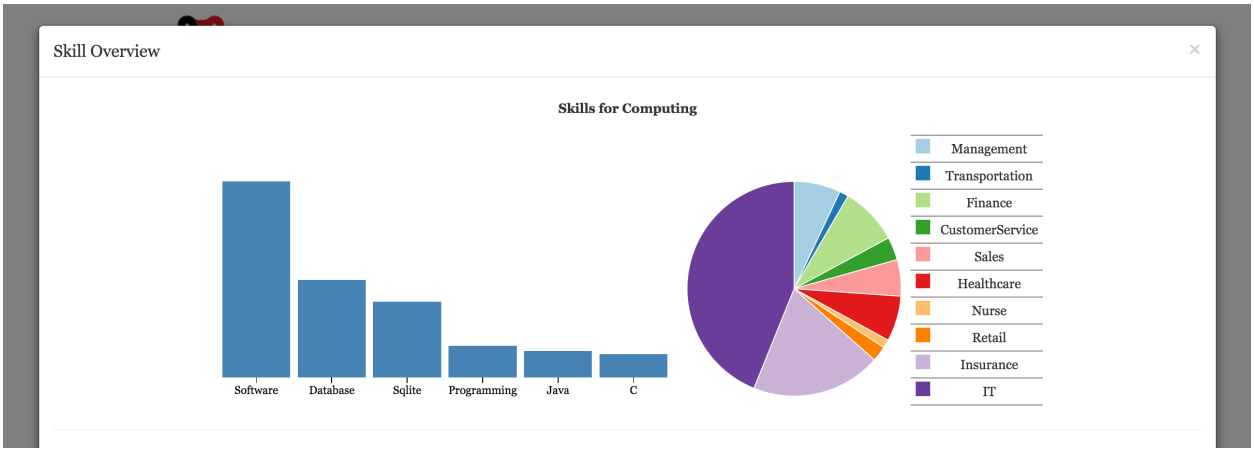
Our user interfaces contain three web pages. On the first page, we intend to give our users an overview of the job market from several different aspects.



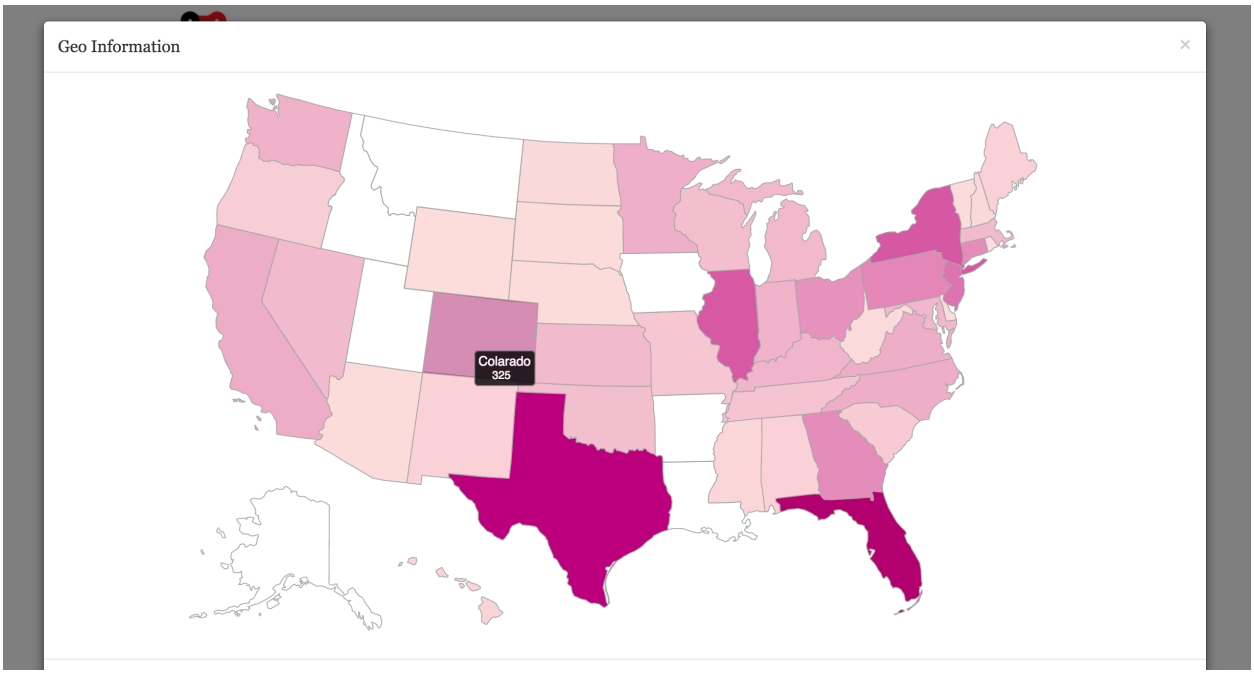
The first part we came up with is Industry Overview. Here the total number of jobs offered in each Industry and State is clearly visualized in an interactive bipartite graph. Users could also navigate through Industries to see the distribution of job opportunities across the States.



The second part we provide our users with is a dashboard for Skill Overview. The top 6 skills for Computing, Statistics and Visualization are visualized in the bar chart. To see what skill sets are most needed for a specific industry, simply point to that industry on the pie chart on the right. We believe our Skill Overview could help users to figure out what skills are most worthwhile to develop if he/she wants to enter a specific industry.

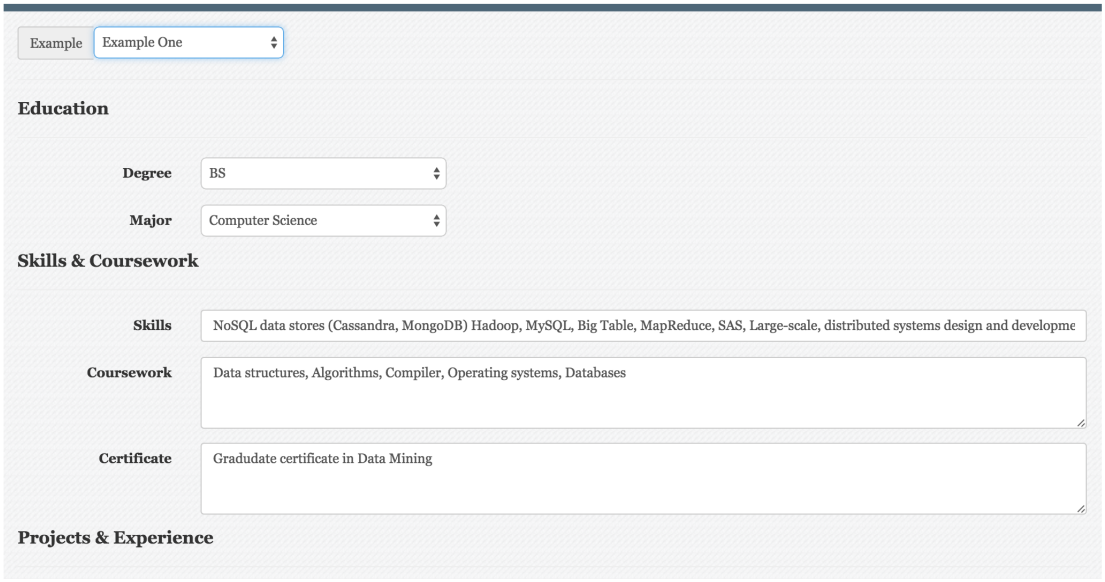


The third part is a map for Geo Information. Here the color gets darker as jobs opportunities grow. User could simply navigate to a specific State to get a more detailed information.



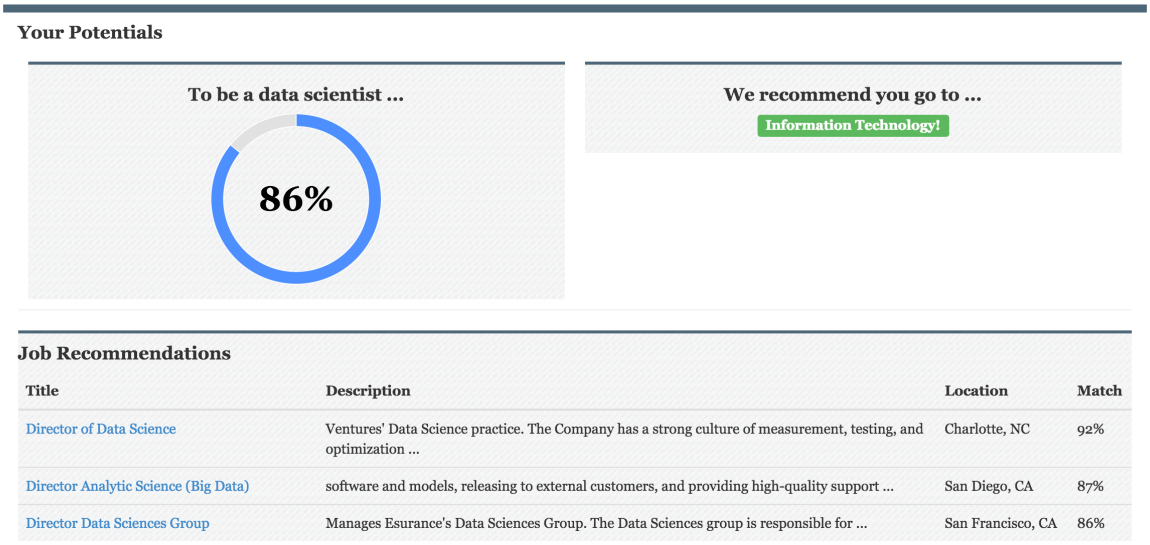
After giving our users an overview of the job market, we will move on to the job recommendation system. And our second web page is mainly designed for information input. Here we ask our users to provide detailed information including their Education, Skills, Coursework, Projects and Experience. The more detailed information we get, the more accurate our recommendation will be. When finished, simply click the Submit button to get access to the next page.

TextIntelligence
We do job recommendation right



The last page addresses the following three questions: 1) How likely is our user to succeed as a data scientist? 2) Which Industry should he/she go to best leverage his/her strength? 3) Which job posts best fit the user's past experience? (Direct link to each recommended job post is provided on our web page.)

TextIntelligence
We do job recommendation right



Experiments/Evaluation

Questions to be answered:

1. How accurate is our proposed multilabel random forest model compared with traditional random forest model, as well as multi-label support vector classifier, in conducting industry label classification using cross validation?

2. How will the number of dimensions of our training data (or number of features) influence classification accuracy of three classification algorithms we experimented with?

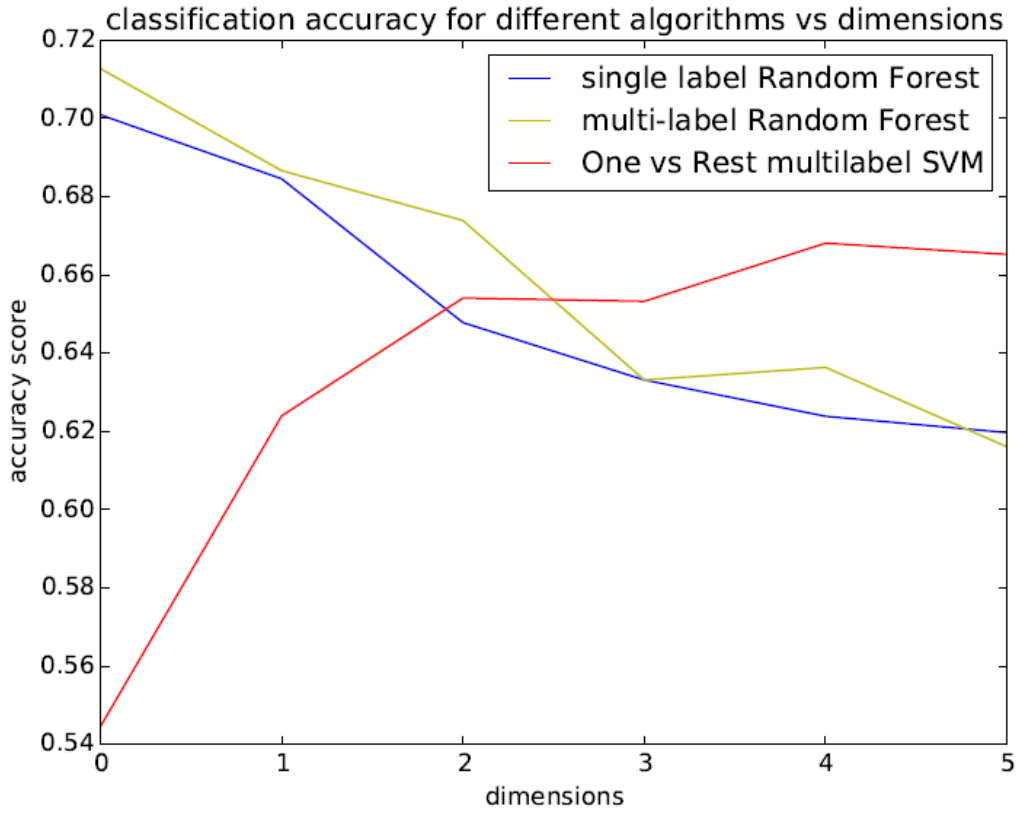
3. By inputting three real world resume, what industry label will our machine learning algorithm assign to each resume ? Whether our multi-label version of random forest algorithm could assess the resume holder of a real data scientist resume be recommended to become a data scientist? Whether those resume holders will be recommended industries that actually make sense?

4. Compared with logistic regression, is multilabel random forest/ traditional random forest algorithm more accurate ? Does their conclusion regarding whether a certain resume holder is a good fit to be a future data scientist make sense?

Detailed description of experiments conducted to answer above questions:

For question 1:
Due to the comparative low speed of running through all required analytic process, we conducted 5 folds cross validations for all three classification algorithms, both original random forest algorithm and multi-label algorithm constructed 30 trees using training data, and select first 10%-50%, 20%-60% , 30%-70%, 40%-80% and 50%-90% rows of TF-IDF matrix as training data, and use the rest rows of TF-IDF matrix as testing data, then calculate their average testing error.

For question2:
we experimented with TF-IDF matrix with dimensions ranging from 10 to 60 by 10, and the testing accuracy of three algorithms is shown below:



As shown above, as the dimension of the TF-IDF matrix increases, the out of sample classification accuracy for both traditional random forest algorithm and multi label random forest algorithm decreases, the testing accuracy for multi-label support vector machine increases, which is quite a surprise for us. Note, dimension 0-5 represent dimension 10 to 60 by 10.

For question3:
By inputting three resume text at the end of our job posting text and integrated into the TF-IDF matrix, our random forest algorithm successfully classify doctor to enter healthcare industry, data scientist to enter information technology industry, and chef to customer service industry when dimension is 50 when dimension of TF-IDF matrix is 50. When dimension of TF-IDF matrix varies, the classification result may vary. So the classification result actually make sense.

For question 4:
Although logistic regression has lower classification accuracy than multilabel/traditional random forest algorithm, their assessment of three real world sample resume makes more sense, as shown in our results.

Conclusion

The conclusion we reached in our project is that through our modified multi-label random forest algorithm, we achieve a high level of accuracy of recommending resume holders to choose the

right industry based on their resume information, and have a comparatively high accuracy of assessing whether or not their background is appropriate for them to be a data scientist. Both multi-label support vector machines and random forest algorithm have comparable level of out of sample classification accuracy through cross validation. We believe with our algorithm, our initial goal of recommending the right talent to pursue a right career path can be reached with some further improvement in our algorithms. We also believe with more interactive and colorful visualization, we are able to deliver our message in a more concise and coherent way to our audience.