Job Database

Author: Kavita Patidar
Email: patidar.ka@husky.neu.edu

ABSTRACT

The target of this project is acquiring information from a site through Scraping and utilizing it to direct different assignments, for example, expectation and essential exploratory investigation. For this investigation, the information is scratched from www.indeed.com an occupation posting site for various organizations. The pertinent factors that will be rejected because of the idea of the site include: Job, Location, Job portrayal, Salary. In the wake of gathering the information, information cleaning will be directed and after that to anticipate factors such as compensation specifically territory, an ability required in all different of occupations and future pattern of a vocation, straight relapse, content classifier and strategic relapse is utilized.

The project is subsequently partitioned into two sections, the information accumulation (Webscraping) and expectation. For Data scraping part I have Data Scraping: Firstly, made list of 450 companies and their websites. Secondly, job related data scraped from one company. Thirdly, data scraped from all companies. Fourthly, Data cleaning has been done. Lastly, a schema has been prepared to use it for use-cases. I scraped data from Indeed.com by using the python module BeautifulSoup. Since the site is relatively well structured and a simple text page hence easy to find the relevant data.

To cover different use cases or expectations, different machine learning models and algorithm has been used, that helped to predict outcome from scraped dataset. These models included linear regression model, linear support vector classification model, Text Classifier, Logistic regression model. Moreover, there are a number of python libraries used namely TfidfTransformer, MultinomialNB, Pipeline and many more which will be discussed in paper.

I. INTRODUCTION

Searching for employments or entry level positions appears its very own errand and the hunt is never again dependent on sole satisfaction of the required activity aptitudes; however, a great deal of systems administration and proposals is included around as well. The measure of work associated with securing the right position assembles a lot of nervousness among the activity searchers and the enrollment specialists who need the correct ability for their organization.

Therefore, I tried to make my project purposeful in solving these two major problems: 1) coordinating the job searchers with the correct jobs and

2) give direction to hopeful occupation searchers on the aptitudes that are sought after with the goal that they can construct them to remain pertinent in the activity advertise. The job suppliers and occupation searchers structure a lot of information which accommodates many intriguing patterns for investigation and translation to capitalize on information accessible.

With the information as of now accessible from the searchers and suppliers, these entanglements can be fixed. The nearness of data on employment aptitudes, compensations and client propensities in many

existing sites, for example, Indeed, LinkedIn, Glassdoor and so forth can be used to coordinate individuals to positions which may appear to be basically incomprehensible without utilizing AI to examine information.

Understanding the job requirements and reaching out to dream company is a tedious task, it is a time consuming and exhausting process if we are going to do it manually. Machine learning provides some easy solution for such problems. The basic idea behind using machine learning is to teach machine to give desired outputs. Experts teach machines using number of algorithms that uses statistics to calculate tasks.

II. METHODOLOGY

1. Data collection through web-scraping:

1.1 **Listing company names and websites-** As the task initiated with figuring out the job-related data from various companies. I took reference from job searching portal Indeed.com. So, I prepared a list of 450 companies and their websites so that I can use them to scrap useful information from their websites. Here is a small snapshot of list.

```
Companylist=[['Walmart', 'www.walmart.com'],
 ['Exxon Mobil', 'www.exxonmobil.com'],
 ['Apple', 'www.apple.com'],
 ['Berkshire Hathaway', 'www.berkshirehathaway.
 ['McKesson', 'www.mckesson.com'],
 ['UnitedHealth Group', 'www.unitedhealthgroup.
 ['CVS Health', 'www.cvshealth.com'],
 ['General Motors', 'www.gm.com'],
 ['Ford Motor', 'www.ford.com'],
 ['AT&T', 'www.att.com'],
 ['General Electric', 'www.ge.com'],
 ['AmerisourceBergen', 'www.amerisourcebergen.c
 ['Verizon', 'www.verizon.com'],
 ['Chevron', 'www.chevron.com'],
```

1.2. Data Scraping- I gathered information from Indeed.com by utilizing the python module BeautifulSoup. Since the site is moderately all around organized and a basic content page thus simple to locate the significant information. I created a robot that extracted data from a single company and then generalized the code to collect information from all companies. As I had use-cases mainly for salary prediction and job skill set, the features I included for data set were job title, salary, company name, location.

Looking at Indeed.com page: (http://www.indeed.com/jobs?q=data+scientist+%2420%2C000&l=New+York&start=10")

1.3. Data Cleaning- I did the data cleaning and guaranteed that there are no invalid qualities such as null values, irrelevant values, unwanted strings with salary data in the dataset. Invalid qualities regularly influence the model results by inciting commotion.

1.4. Creating Schema- to collect all the information at one place to use it further for the use-cases, I firstly append all the features (city, job, salary, company, expertise, recruiter) and then exported it into a csv file.

2. Use-Cases:

2.1. Prediction of what can be the salary for a particular region in USA for a specific job title (e.g. Data Scientist).

Linear regression is used to predict salary. Finding- for Data Scientist position highest salary is in San Francisco and second highest is in New York. We need to foresee a twofold factor - regardless of whether the salary was low or high. Process the middle compensation and make another paired variable that is genuine when the sa;ry is high (over the middle)

I could likewise perform Linear Regression (or any relapse) to anticipate the compensation esteem here. Rather, we are going to change over this into a double grouping issue, by foreseeing two classes, HIGH versus LOW compensation. While performing relapse might be better, performing characterization may help evacuate a portion of the commotion of the extraordinary pay rates. We don't need to decision the middle as the part point - we could likewise part on the 75th percentile or some other sensible limit.

city	job	salary	company	how high	is_hi
New York	Data Analyst - Cardiology	33650.0	Columbia University	super_low	
New York	Associate Research Scientist	120000.0	Columbia University	super_high	
New York	Research and Brand Strategy Apprentice	147500.0	Insight Strategy Group	super_high	
New York	City Research Scientist II	77975.0	City of New York	low	
New York	Business Intelligence Engineer	150000.0	Amazon Corporate LLC	super_high	

Algorithm- Linear Regression model created to predict the salary in different region. Create a few new variables in your dataframe to represent interesting features of a job title.

- created a feature that represents whether 'Senior' is in the title
- or whether 'Manager' is in the title.
- Then build a new Logistic Regression model with these features.

```
salarypredictions = model.predict(X)
datafr['salarypredictions']=salarypredictions
#Prediction for each given city and for a Datascientist
jobt=datafr[datafr['job'].str.contains("Data Scientist")==True]
predctsalary=jobt[["city","job","salarypredictions"]]
predctsalary
```

	city	job	salarypredictio
11	New York	Senior Data Scientist	123633.0600
12	New York	Lead Data Scientist	123633.0600

Finding- for Data Scientist Job, highest salary will be in San Francisco and second highest will be in New York. However, Lowest salary will be in Pittsburgh.

2.2. Prediction of what can be the skill sets requirements in a particular region in USA

Encoding- Firstly I encoded the features with the help of Label Encoder. I used “preprocessing.LabelEncoder()” which Encode labels with value between 0 and n_classes-1.

Train and Test data split- I split dataset into train and test data to perform model on train set and then validate it on test set. 'numpy.reshape'- - Gives another shape to a cluster without changing its information (use-It is normal to need to reshape a one-dimensional exhibit into a two-dimensional cluster with one section and various exhibits. NumPy gives the reshape() work on the NumPy exhibit object that can be utilized to reshape the information)

Linear SVC (Support Vector Classifier) is to fit to the information we give, restoring a "best fit" hyperplane that partitions, or arranges, our information. From that point, in the wake of getting the hyperplane, we would then be able to encourage a few highlights to our classifier to perceive what the "anticipated" class is.

I checked the accuracy of model Linear support vector as shown below:

```
#Checking accuracy of the model
print("Linear Support Vector Classification in % : ",accuracy_score(y_test, pred, normalize = True)*100)
```

```
Linear Support Vector Classification in % :  100.0
```

Finding: From the above tables the required skills in New York City include: Android development, Data Science, Data analyst, IOS developer, SAP PI/PO, Java, SQL, Full Stack Java & UI, DevOps, QA, and Net & UI.

```
: #Prediting which skills arerequired in New York
datafr["Skill requirement per city"] = pred
data=datafr[datafr['city'].str.contains("New York")==True]
predctskill=data[["city","Skill requirement per city"]]
predctskill
```

:

	city	Skill requirement per city
0	New York	2.0
1	New York	7.0
2	New York	0.0
3	New York	7.0
4	New York	28.0

2.3. Most relevant skill required for a job position.

After splitting that data into test and train set, I used logistic regression to predict most relevant skill required for any job position.

Finding- For one to get employment for the Front End SDE position in Amazon Corporate LLC, the job seeker should contain skills in IOS

2.4. Find out a skill required in how many kinds of jobs.

To predict these jobs, I used a number of libraries.

"CountVectorizer"- It converts text content to a count matrix.

```
#Extracting features from text files.
from sklearn.feature_extraction.text import CountVectorize
count_vect = CountVectorizer()
X_train_counts = count_vect.fit_transform(datafr)
X_train_counts.shape
```

"TfidfTransformer"-Transform a count matrix to a normalized tf or tf-idf representation. Tf stands for term-frequency while tf-idf stands for term-frequency times inverse document-frequency.

```
from sklearn.feature_extraction.text import TfidfTransformer
tfidf_transformer = TfidfTransformer()
X_train_tfidf = tfidf_transformer.fit_transform(X_train_counts)
X_train_tfidf.shape
```

The multinomial Naive Bayes classifier is appropriate for characterization with discrete highlights (e.g., word means content arrangement). The multinomial dissemination ordinarily requires number component tallies. In any case, by and by, partial considers such tf-idf may likewise work. Multinomial calculated relapse (frequently just called 'multinomial relapse') is utilized to anticipate an ostensible ward variable given at least one free factor. A clear-cut variable (here and there called an ostensible variable).

```
from sklearn.naive_bayes import MultinomialNB
clf = MultinomialNB().fit(X_train_tfidf, target)
```

"Pipeline"- - Sequentially apply a rundown of changes and a last estimator. Middle strides of the pipeline must be 'changes', that is, they should actualize fit and change techniques. The last estimator just needs to execute fit. The transformers in the pipeline can be stored utilizing memory contention.

```
from sklearn.pipeline import Pipeline
text_clf = Pipeline([('vect', CountVectorizer()),('tfidf', TfidfTransformer()),('clf', MultinomialNB()),])
text_clf = text_clf.fit(data ,target )
```

The reason for the pipeline is to collect a few stages that can be cross-approved together while setting diverse parameters. For this, it empowers setting parameters of the different advances utilizing their names and the parameter name isolated by a '__', as in the precedent beneath. A stage's estimator might be supplanted completely by setting the parameter with its name to another estimator, or a transformer expelled by setting to None

Finding- Skills required for data scientist used in a number of other jobs (few examples: Quantitative Analyst, Research Analyst, Perinatal Data Center, 'Account Executive - Razorfish, 'Marketing Research Analyst', 'Lead Data Scientist - Video/Images'

One more example-

```
classf=classified.tolist()
datafrf["classf"]=classf
detailstext=datafrf[datafrf["job"].str.contains("Research Analyst, Perinatal Data Center")==True]
detailstext
```

	Rec	city	job	salary	company	Expertise	Recruiter	salarypredictions	jobt	cityt	companyt	Expertiset	Recruitert	Skill requirement per city	classf
24	90	White Plains	Research Analyst, Perinatal Data Center	58000.0	March of Dimes	Salesforce	Dennis Sanchez	58000.0	80	35	55	30	6	30.0	Research Analyst, Perinatal Data Center

Naive Bayes text Classification Model accuracy in % : 92.85714285714286

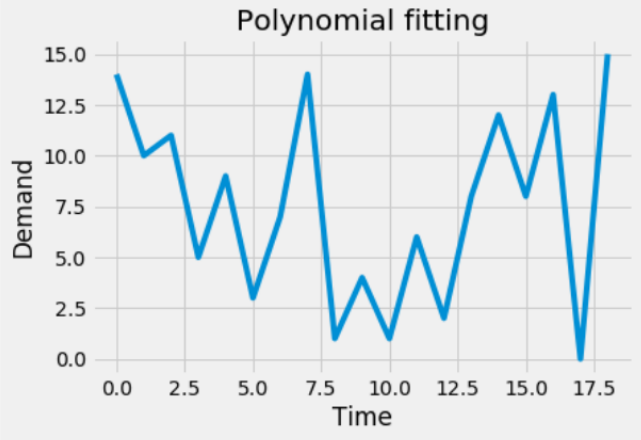
2.5. Predicting trend in Data Science Jobs in the next 10 years

Since there is not a timeseries dataset, I first converted it to a time series. Time series - is a collection of information focuses filed, recorded or charted in time request. Most normally, a period arrangement is a grouping taken at progressive similarly separated focuses in time. Thus, it is a succession of discrete-time information.

Then I created a time range of 19days in January 2019 and pattern of Data Scientist job demand in these days. For example:

Date	job
2019-01-01	Senior Data Scientist
2019-01-02	Lead Data Scientist
2019-01-03	Lead Data Scientist - Video/Images
2019-01-04	Data Scientist - Statistics
2019-01-05	Indoor Environmental Data Scientist
2019-01-06	Data Scientist - Forecasting
2019-01-07	Data Scientist needed for Tech Medical Company!
2019-01-08	Senior Data Scientist

Then I created polynomial graph that showed demand of data scientist job in increasing time.



Forecasting the Time Series- For future forecasting, I used ARIMA model.

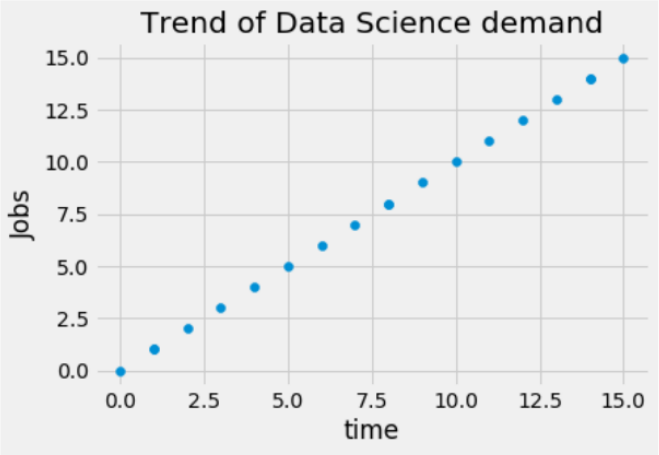
"ARIMA model"-- statistical method for analyzing and forecasting time series data. ARIMA is an acronym that stands for AutoRegressive Integrated Moving Average.

AR: Autoregression. A model that utilizes the needy connection between a perception and some number of slacked perceptions. I: Integrated. The utilization of differencing of crude perceptions (for example subtracting a perception from a perception at the past time venture) so as to make the time arrangement stationary. Mama: Moving Average. A model that utilizes the reliance between a perception and a remaining mistake from a moving normal model connected to slacked perceptions. Every one of these parts are expressly indicated in the model as a parameter. A standard documentation is utilized of ARIMA(p,d,q) where the parameters are substituted with whole number qualities to rapidly show the particular ARIMA model being utilized.

The parameters of the ARIMA model are characterized as pursues:

p: The quantity of slack perceptions incorporated into the model, likewise called the slack request. d: The occasions that the crude perceptions are differenced, additionally called the level of differencing. q: The measure of the moving normal window, likewise called the request of moving normal.

Polynomial trend curve of demand-



To evaluate the stability of model in predicting the demand for data scientist job, I used linear regression model. Where I found the p-value of the polynomial model is 0.000 which is less than 0.05 hence we fail to reject the null hypothesis of no effect and conclude that time has an effect of the demand of data scientists.

Finding- There is an increase in the demand of data scientist next to 10 years.

III. CONCLUSION

Job database project has been created to address some problems such as

1) matching the job seekers with the right employers and

2) provide guidance to aspiring job seekers on the skills that are in demand so that they can build them to stay relevant in the job market. The job providers and job seekers form a large amount of data which provides for many interesting trends for analysis and interpretation to make the most of data available.

To complete the project, process is divided into two part: Data Scraping and Use-cases.

Data Scraping: Firstly, made list of 450 companies and their websites. Secondly, job related data scraped from one company. Thirdly, data scraped from all companies. Fourthly, Data cleaning has been done. Lastly, a schema has been prepared to use it for use-cases.

Use-Cases: 5 use cases have been covered.

1. Prediction of what can be the salary for a particular region in USA for a specific job title (eg. Data Scientist). Linear regression is used to predict salary. Finding- for Data Scientist position highest salary is in San Francisco and second highest is in New York.

2. Prediction of what can be the skills sets requirements in a particular region in USA Linear Support Vector Classification is used to predict skill set. Finding- the required skills in New York City include: Android development, Data Science, Data analyst, IOS developer, SAP PI/PO, Java, SQL, Full Stack Java & UI, DevOps, QA, and Net & UI

3. Most relevant skill required for a job position. Logistic Regression used. Finding- For one to get employment for the Front End SDE position in Amazon Corporate LLC, the job seeker should contain skills in IOS

4. Find out a skill required in how many kinds of jobs Text Classifier is used. Finding- Skills required for data scientist used in a number of other jobs (few examples: Quantitative Analyst, Research Analyst, Perinatal Data Center, 'Account Executive - Razorfish, 'Marketing Research Analyst', 'Lead Data Scientist - Video/Images'

5. Predicting trend in Data Science Jobs in the next 10 years ARIMA and Linear model used. Finding- There is an increase in the demand of data scientist next to 10 years.

REFERENCES:

1. Safdari, N.(2018).Multi-Class Text Classification with SKlearn and NLTK in python| A Software Engineering Use Case.Retrieved from:https://towardsdatascience.com/multi-class-text-classification-with-sklearn-and-nltk-in-python-a-software-engineering-use-case-779d4a28ba5

2. Data Quest.(2019). Scikit-learn Tutorial: Machine Learning in Python. Retrieved from:https://www.dataquest.io/blog/sci-kit-learn-tutorial/

3. Arnaud, Z. (2018).Playing with time series data in python. Retrieved from:https://towardsdatascience.com/playing-with-time-series-data-in-python-959e2485bff8

4. Marsden, E.(2019). Monte Carlo sampling methods. Retrieved from:https://risk-engineering.org/notebook/monte-carlo-LHS.html

5. Aarshay,J. (2016).A comprehensive beginner's guide to create a Time Series Forecast (with Codes in Python). Retrieved from:https://www.analyticsvidhya.com/blog/2016/02/time-series-forecasting-codes-python/

6. Thomas,C. (2011).|Spectrum - Spectral Analysis in Python (0.5.2): Fourier Methods. Retrieved from:http://thomas-cokelaer.info/software/spectrum/html/user/ref_fourier.html

7. Bronshtein, A.(2017).Simple and Multiple Linear Regression in Python. Retrieved from:https://towardsdatascience.com/simple-and-multiple-linear-regression-in-python-c928425168f9

8. Sharma, M.(2018).Web Scraping with Python and BeautifulSoup .Retrieved from:https://medium.com/incedge/web-scraping-bf2d814cc572

9. https://stackoverflow.com/questions/41792471/installing-wordcloud-using-jupyter-notebook

10. https://www.kaggle.com/adiljadoon/word-cloud-with-python

11. https://github.com/jupyter/notebook/issues/1892

12. https://www.fernandomc.com/posts/using-requests-to-get-and-post/

13. https://stackoverflow.com/questions/46253288/scrape-with-correct-character-encoding-python-requests-beautifulsoup

14. https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.get_dummies.html

15. https://towardsdatascience.com/the-dummys-guide-to-creating-dummy-variables-f21faddb1d40

16. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html

17. https://scikitlearn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

18. https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/