

***A Mini-Project Report On***

**“Finding the trend of jobs in the IT market”**

***Submitted By***

**Mangirish Bhagat**

**Roll No: 6**

**F.Y. M.Sc. (Data Science and Big Data Analytics)**

**School of Computer Science**

**Faculty of Science**

**MIT – World Peace University**

**Pune - 411038**

**Academic Year 2019-2020**

**APRIL – 2020**

**MIT WORLD PEACE UNIVERSITY, PUNE**

**SCHOOL OF COMPUTER SCIENCE**

***Certificate***

This is to certify that

**Mangirish Bhagat, Roll No. 6,**

Of ***M.Sc. (Data Science and Big Data Analytics)*** successfully completed his Mini-Project in

**“Finding the trend of jobs in the IT market”**

to our satisfaction and submitted the same during the academic year 2019- 2020 towards the partial fulfillment of degree of **Master of Science in Data Science and Big Data Analytics** of MIT World Peace University under the School of Computer Science, MIT WPU, Kothrud, Pune.

|  |  |  |
| --- | --- | --- |
| **Prof. Dr. Shubhalaxmi Joshi** | **Prof. Sachin Bhoite** | **Prof. Supriya Aras** |
| **Associate Dean** | **Program Head** | **Assistant Professor** |
| **Faculty of Science** | **School of Computer Science** | **School of Computer Science** |
| **MITWPU** | **MIT WPU** | **MIT WPU** |

**ACKNOWLEDGEMENT**

I would like to express my special thanks of gratitude to my teacher **Prof. Prajakta Soman** as well as our dean **Dr. Shubhalaxmi Joshi** who gave us the golden opportunity to do this wonderful project on the topic “Finding the trend of jobs in the IT market”, which also helped me in doing a lot of Research and I came to know about so many new things I am really thankful to them. Secondly, I would also like to thank my team members who helped me a lot in finalizing this project within the limited time frame. Lastly, I would like to thank to all the lab assistants and professors who helped us to achieve us the end goal till the end of our project.

Mangirish Bhagat

Roll No. 6

Contents

[Introduction 6](#_Toc39134749)

[Domain Name 6](#_Toc39134750)

[Motivation 6](#_Toc39134751)

[Problem Statement 6](#_Toc39134752)

# Introduction

## Domain Name

**IT Job Market:**

Corporate IT Industry job analysis is the process of reviewing key aspect of job’s posted by various companies on employment websites, we have used this data from Indeed.com, Glassdoor.com & Naukri.com. This data helps not just only to the job seeking employee but also the employer to analyse the market, the need of the industry and the skills requirements of various companies and job designations. Employees are able to access these employment websites but there are not many applications which would show which jobs are trending and what skills can give you the desired job. having this data gives you opportunity to learn the skill to get into the corporate world.

## Motivation

Predicting the job market is not one of the most explored domain, in recent years there has been very few analysis done over this particular domain, This problem statement was given for 2020 Smart India Hackathon by the government of Uttarakhand, this also shows the need to explore this domain and put out the results to the young job seeking professional’s as this analysis would help each individual to analyse what the corporate world is asking for and then be proficient in those particular skills.

From research point of view, this domain was just brushed upon. We were motivated to do explore this domain in a different way as we decided to analyse the jobs based the skills demand. Our model helps such that one can go through their interested, strong skills and analyse the kind of job position and salary offered, this helps one to understand the job market and waste no time in researching, also would help in utilising same time to gain the skills.

## Problem Statement

Predicting the trends of quality-oriented jobs in IT sector.

Dataset Details

In our initial plan, when we started the project was to get the data, which was available on the job websites, in the form of APIs, which could let us download and use that data. As the APIs where not available freely and also we had applied for the Indeed API and there has not been any update from the Indeed team. We also searched for other sites if they are giving us their API(s) but most of these websites had paid API(s) and some weren't offering any API(s) on their platform. As students we thought that purchasing API(s) wont be a good option, so finally, we took the advantage of web-scrapping. Web-scrapping is a process where the data is extracted directly from the website using a programming language to directly use it for own purposes. As we moved forward, we realized that on most of the websites web-scrapping is not possible due to certain restriction on the web servers. Only Indeed was providing us to web-scrape on their website. Since web scrapping was successful on indeed.com, we were getting data of the regular updated job posts and also jobs posted over a month back and it had the attributes necessary for the further analysis. The code written in python would also convert the data to a comma separated value (CSV) file. Size of data would be 90000+ records (till date) as new data is coming every day.

**The dataset attributes are:**

|  |  |
| --- | --- |
| **Title** | Title of the job |
| **Location** | Location of the job |
| **Company** | Name of the company |
| **Salary** | Salary of the job |
| **Sponsored** | Show sponsored if the job is posted by agent and show organic if posted by company. |
| **Description** | An important attribute of the dataset would be Description. It explains the whole description of the job. We used NLP (Natural Language Processing) to analyze the data and obtained many new attributes from it. |
| **Time** | The date when the job was posted. |

These were all the attributes obtained from web scrapping.

**Attributes which were obtained from NLP:**

|  |  |
| --- | --- |
| **Salary\_Unit\_Month, Salary\_Unit\_Year, Salary\_Unit\_Week, Salary\_Unit\_Day, Salary\_Unit\_Hour** | From the job description column, the salary was not  given perfectly so we had to divide the parts into month,  year, week, day and hour, so if a company gives salary  month wise the **“Salary\_Unit\_Month”** attribute will  have value **1** |
| **Salary\_New** | This attribute is the preprocess part of Salary. The null  values were removed. |
| **Salary\_Average** | Taking the average of the above attribute as the value was given is range. |
| **Month\_Salary** | Here we have taken every type of unit into a single  unit that is month. Taking this unit was useful for some  analysis as data was getting restricted due to the units it had |
| **Job\_Type\_Part\_Time, Job\_Type\_Full\_Time** | Shows if the job has a full time / part time job |
| **XP\_Experience, XP\_Fresher** | If a company wants experience / fresher candidates. |
| **Gender, Gender\_Male, Gender\_Female** | If a company wants a specific gender. |
| **Education\_Tenth, Education\_Twelvth, Education\_Bachelors, Education\_Masters,**  **Education\_Doctorate** | Shows the qualification required for a particular job |
| **Skills** | Shows the basic skills required from the candidates. |
| **Position\_Junior, Position\_Senior, Internship** | The post which the company is offering. |
| **Title\_New** | This attribute contains only the skills of IT  company. |

Theory and Facts

Technology continues to rapidly evolve and grow, which means the IT growth outlook is stronger than ever.

Employment of computer and information technology occupations is projected to grow 12 percent from 2018 to 2028, much faster than the average for all occupations. These occupations are projected to add many new jobs. Demand for these workers will stem from greater emphasis on cloud computing, the collection and storage of big data, and information security.

The rising number of technology companies and the subsequent high demand for qualified professionals have changed the IT growth outlook. Salaries of information technology jobs are flourishing.

Some fields where the most Growth was seen:

**(1) Information security analyst**

**(2) Data scientist**

**(3) Business intelligence developer**

**(4) Development Operations engineer**

**(5) Mobile applications developer**

**(6) Web developer**

**(7) Database administrator**

**(8) User interface designer**

And more will be added in the future

**Big Data Tools:**

**Apache Spark** is open source, general-purpose distributed computing engine used for processing and analysing a large amount of data. Just like Hadoop MapReduce, it also works with the system to distribute data across the cluster and process the data in parallel.

**Python** is a general purpose and high-level programming language. You can use Python for developing desktop GUI applications, websites and web applications. Also, Python, as a high-level programming language, allows you to focus on core functionality of the application by taking care of common programming tasks.

**PySpark** is the Python API written in python to support Apache Spark. Apache Spark is a distributed framework that can handle Big Data analysis. Using PySpark, one can easily integrate and work with RDDs in Python programming language too. There are numerous features that make PySpark such an amazing framework when it comes to working with huge datasets.

**MongoDB** is an object-oriented, simple, dynamic, and scalable NoSQL database. It is based on the NoSQL document store model. The data objects are stored as separate documents inside a collection — instead of storing the data into the columns and rows of a traditional relational database.

**PyMongo** distribution contains tools for interacting with MongoDB database from Python. The bson package is an implementation of the BSON format for Python. The pymongo package is a native Python driver for MongoDB.

**Django** is a Python-based free and open-source web framework, which follows the model-template-view architectural pattern.

**Dash and Plotly** is a python based free open source web framework for designing a single page interactive AJAX based websites.

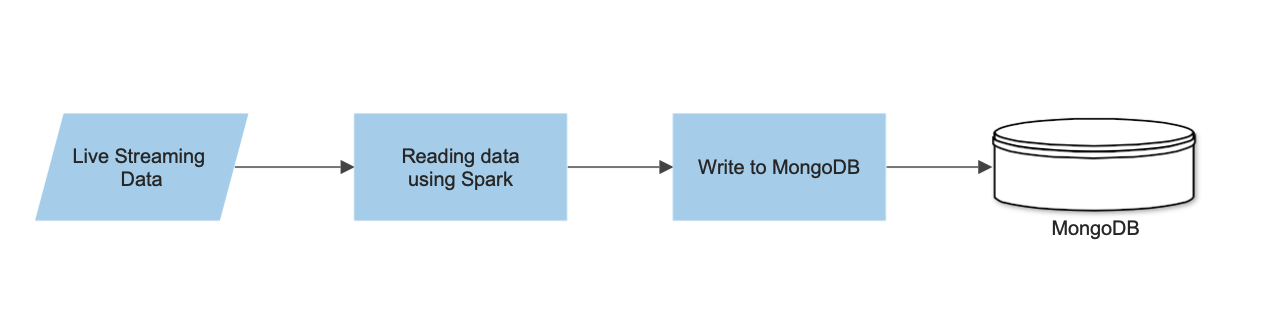
**Design of the System**

**Proposed Architecture (Initially Planned)**

We were using 6 PC(s) for our big data project.

1. **5 PC for CentOS**
   * Installed Hadoop in every PC
   * Out of 5 PC’s, 2 were used for Name Node and 3 were used for Data Node
   * So, for name nodes there were basically 1 primary name node and 1 secondary name node
   * If the primary name node fails due to some reason the secondary name node automatically takes place
   * Replication Factor
2. **1 PC for Kali Linux**
   * selenium would be used for web scrapping
   * Kali Linux also used for security testing
   * To find out any risk factors

**Architecture of implemented system:**



**Technology Stack**

**Programming Language**: Python.

**Database:** MongoDB.

**Framework:** PySpark , Django.

**Tools:** PyMongo.

**Editor:** Jupyter Notebook.

**Software Specifications:**

CentOS

Kali Linux

Spark – 2.0 or higher (we are using 2.4.5)

Python – 3.5 or higher (we are using 3.7.6)

MongoDB – 3.6 or higher (we are using 4.2.5)

**Hardware Specifications:**

8GB RAM or higher

Intel Core i5 3rd generation or higher

**Implementation Details**

Remaining:

1. Implementation Details (This section should be in sync with design/architecture mentioned in section 4)

* 1. Step-by-step processing done, including config and parameter settings, pre-processing the data etc
  2. Pseudocode, flowchart or algorithm details
  3. Any important observations in tabular format
  4. Final output details, results, visualizations used if any. Explain your findings and output
  5. Any other point relevant to the implementation
  6. Other details
     1. Actual time taken to complete the project in terms of no. of days \* 3 members.

Alvyn Abranches - 58 hours

Mangirish Bhagat - 3 hours

Utkarsh Patel - 20 hours

* + 1. LOC

.

├── apriori.ipynb - 25 lines, By Alvyn Abranches

├── backup

│   ├── fd57d77770172fa7de301ff18cb0087c.xlsx

│   └── indeed\_results.xlsx

├── bigdata

│   └── website

│   ├── app.py - 174 lines, By Alvyn Abranches

│   └── \_\_pycache\_\_

│   └── app.cpython-37.pyc

├── bigdata.html

├── chromedriver

│   ├── 80

│   │   ├── chromedriver

│   │   └── chromedriver.exe

│   └── 81

│   ├── chromedriver

│   └── chromedriver.exe

├── data

├── debug.log

├── documentation

│   ├── gru vs lstm.jpg

│   ├── lstm vs gru.png

│   └── neural network architectures.png

├── EDA\_I.ipynb - 55 lines

├── Education vs Jobtitle - ANN.ipynb - 109 lines, By Alvyn Abranches

├── Education vs Jobtitle.ipynb - 79 lines, By Alvyn Abranches

├── Glassdoor

│   ├── Android-India-job-results.csv

│   ├── data+scientist-Pune-job-results.csv

│   └── GLASSDOOR\_WEBSCRAPPING.py - 125 lines, By Mangirish Bhagat

├── indeed\_results\_pp\_2020-04-27.xlsx

├── indeed\_results.xlsx

├── india\_pincodes.txt

├── \_\_init\_\_.py - 42 lines, By Alvyn Abranches

├── jobs.py - 17 lines, By Alvyn Abranches

├── Linear Reg.ipynb - 23 lines, By Utkarsh Patel

├── locations.py - 1620 lines, By Alvyn Abranches

├── Logestic\_Regression on XP\_Experience, XP\_Fresher VS Salary\_Average.ipynb - 22 lines, By Alvyn Abranches

├── LSTM.ipynb - 96 lines, By Alvyn Abranches

├── ml

│   ├── algorithms

│   │   ├── base\_classifiers.py - 58 lines, By Alvyn Abranches

│   │   ├── base\_regressors.py - 74 lines, By Alvyn Abranches

│   │   └── \_\_pycache\_\_

│   │   ├── base\_classifiers.cpython-37.pyc

│   │   └── base\_regressors.cpython-37.pyc

│   ├── model

│   ├── preprocessing

│   │   ├── cleansing.py - 698 lines, By Alvyn Abranches

│   │   ├── preprocess\_data.py - 153 lines, By Alvyn Abranches

│   │   └── \_\_pycache\_\_

│   │   ├── cleansing.cpython-37.pyc

│   │   └── preprocess\_data.cpython-37.pyc

│   └── website

│   ├── app.py - 263 lines, By Alvyn Abranches

│   └── \_\_pycache\_\_

│   └── app.cpython-37.pyc

├── preprocessing.ipynb - 39 lines, By Alvyn Abranches

├── public

│   └── pincodes.csv

├── \_\_pycache\_\_

│   ├── \_\_init\_\_.cpython-37.pyc

│   ├── jobs.cpython-37.pyc

│   └── locations.cpython-37.pyc

├── Qualification, Experience vs Salary - ANN.ipynb - 96 lines, By Alvyn Abranches

├── Qualification, Experience vs Salary.ipynb - 93 lines, By Alvyn Abranches

├── README.md - 15 lines, By Alvyn Abranches

├── server.py - 40 lines, By Alvyn Abranches

├── try.ipynb - 84 lines, By Alvyn Abranches

├── webscrapping

│   ├── multithreadwebscraping.py - 63 lines, By Alvyn Abranches

│   ├── preprocessing.py - 43 lines, By Alvyn Abranches

│   ├── \_\_pycache\_\_

│   │   ├── multithreadwebscraping.cpython-37.pyc

│   │   ├── preprocessing.cpython-37.pyc

│   │   └── webscrapping.cpython-37.pyc

│   ├── Readme.md - 4 lines, By Alvyn Abranches

│   └── webscrapping.py - 188 lines, By Alvyn Abranches

└── website

├── bigdata

│   ├── admin.py - 4 lines, By Alvyn Abranches

│   ├── apps.py - 6 lines, By Alvyn Abranches

│   ├── \_\_init\_\_.py - 4 lines, By Alvyn Abranches

│   ├── migrations

│   │   ├── \_\_init\_\_.py - 1 line, By Alvyn Abranches

│   │   └── \_\_pycache\_\_

│   │   └── \_\_init\_\_.cpython-37.pyc

│   ├── models.py - 4 lines, By Alvyn Abranches

│   ├── \_\_pycache\_\_

│   │   ├── admin.cpython-37.pyc

│   │   ├── \_\_init\_\_.cpython-37.pyc

│   │   ├── models.cpython-37.pyc

│   │   └── views.cpython-37.pyc

│   ├── tests.py - 4 lines, By Alvyn Abranches

│   └── views.py - 8 lines, By Alvyn Abranches

├── db.sqlite3

├── manage.py - 22 lines, By Alvyn Abranches

├── ml

│   ├── admin.py - 4 lines, By Alvyn Abranches

│   ├── apps.py - 6 lines, By Alvyn Abranches

│   ├── \_\_init\_\_.py - 1 line, By Alvyn Abranches

│   ├── migrations

│   │   ├── \_\_init\_\_.py - 1 line, By Alvyn Abranches

│   │   └── \_\_pycache\_\_

│   │   └── \_\_init\_\_.cpython-37.pyc

│   ├── models.py - 4 lines, By Alvyn Abranches

│   ├── \_\_pycache\_\_

│   │   ├── admin.cpython-37.pyc

│   │   ├── \_\_init\_\_.cpython-37.pyc

│   │   ├── models.cpython-37.pyc

│   │   └── views.cpython-37.pyc

│   ├── tests.py - 4 lines, By Alvyn Abranches

│   └── views.py - 8 lines, By Alvyn Abranches

└── website

├── asgi.py - 17 lines, By Alvyn Abranches

├── \_\_init\_\_.py - 1 line, By Alvyn Abranches

├── \_\_pycache\_\_

│   ├── \_\_init\_\_.cpython-37.pyc

│   ├── settings.cpython-37.pyc

│   ├── urls.cpython-37.pyc

│   └── wsgi.cpython-37.pyc

├── settings.py - 123 lines, By Alvyn Abranches

├── urls.py - 29 lines, By Alvyn Abranches

└── wsgi.py - 17 lines, By Alvyn Abranches

33 directories, 91 files

* + 1. Monetary Cost involved if any (cloud space purchase, hardware purchase etc)

No monetary cost is involved in this project as we tried to keep it local to our system.

Testing details

1. In tabular format, mention few test cases and expected behavior.

In testing we having only tried to find the comparisons between PyMongo and PySpark, and we found out that PySpark was faster than just inserting or retrieving directly from PyMongo.

Limitations

1. Some of the drawbacks of **Apache Spark** are that is does not work correctly on Windows based system as it just simulates the enviroment over the Windows Operating System and doesnt work like production or even development server.
2. No Java or Python API bindings.
3. Can only read (no updates)

Conclusion and Future Work

**Conclusion**

After doing this project we found out that there are interesting insights we can get from the dataset and it will be useful for companies if they use a similar type of dataset. This work focuses on the challenge of showing the trends of the jobs offered by companies through job posts on the website. The app would show all the jobs available from the attributes selected by the user. App can be used by students so they can see the trends going on in the market and it could help them choose a better option. Employees could use it to see if there is a better scope for them in a different fields based on their skills, experience etc.

**Future Scope**

1. Instead of focusing just on IT Industry we can extend it to all other sectors.
2. Creating a better GUI (Graphical User Interface), by adding more effects and features and also decreasing the time it takes to render the webpage on the client side.
3. Getting more amount of data not just for India but worldwide and increase it country by country.
4. Getting data from APIs if they will be available for us freely in the future.

Bibliography

[1] Ms. C. Vijayalakshmi, “Execution of an Advanced Data Analytics by Integrating Spark with MongoDB,” Department of Computer Science and Engineering, Jubail University College (Female Branch).

[2] A MongoDB white paper, “Apache Spark andMongoDB–Turning Analytics into Real-Time Action”

[3] Poonam Ligade, “Processing JSON data using Spark SQL Engine: Data Frame API”,October 2015,Article in edupristine

[4] <http://www.slideshare.net/SparkSummit/adding-complex-data-to-spark-stackneeraja-rentachintala>

[5] <https://www.mongodb.com/press/mongodb-enables-advanced-real-time-analytics-on-fast-moving-data-with-new-connector-for-apache-spark>

[6] “Analysis of real-time data with spark streaming,”Nikitha Johnsirani Venkatesan, Choon Sung Nam, Earl Kim, Dong Ryeol Shin, School of Electrical and Computer Engineering, Sungkyunkwan University, Suwon, South Korea

[7] <http://spark.apache.org/research.html>

[8] <http://www.kdnuggets.com/2015/09/spark-sql-real-time-analytics.html>

[9] <http://www.simba.com/webinar/connect-tableau-big-data-source/>

[10] <https://docs.mongodb.com/spark-connector/master/python-api/>

[11] <https://www.mongodb.com/blog/post/tutorial-for-operationalizing-spark-with-mongodb>

[12] <https://docs.mongodb.com/spark-connector/master/python-api/>

Appendix

**Challenges faced during lockdown:**

1. We could not implement the proposed system, because of lack of resources.
2. Since we were web scrapping the data which requires continuous internet connection. Most of the times scrapping got terminated due to inconsistent bandwidth.
3. Working together when present physically brings a lot of productivity in the project, and it wasn’t the same after the lockdown.
4. Solving each other’s doubts and solving queries within the project was a challenge because of the distance and sometimes connectivity issues.