Basket Analysis By Association Rule in Tech Skills

*

1st Yousef sarkar
Computer Science -Big Data.)
Umm Al Qura University)
Makkah , Saudi Arabia
yousef0sarkar@gmail.com

***Abstract*—In this project, the latest data mining algorithms were used to find the related elements to process the related skills at the big data level, and the tools and techniques were integrated to produce this solution.**

I. Introduction and Motivation

there are a lot of skills in tech jobs such as programming languges or frameworks in technology sectors some of them is related together and other skills are not , so we aim to provide helping tool that can help job seeker to find a set of skills related togather as basket or group of skills ,based in main skill that user defined . we will using big data technology to solve this problem in this project.

II. Related Work

[1]Unsupervise learned word embeddings have seen tremendous success in numerous Natural Language Processing (NLP) tasks in recent years in Skill2Vec . [2]Skill2vec: Machine Learning Approach for Determining the Relevant Skills from Job Description

III. Project Idea Description and Challenges

The main idea of the project is the possibility of searching for skills that related to each other, as job advertisements include some requirements, including skills Our goal is to learn the skills associated with each other through A key can be entered by the user for a field, and then Associated skills are shown as output .

we aim to Mining frequent skills in dataset, the size of the dataset is 50k rows, and every row has a set of skills, we need to get the frequent item based in threshold confidence so we will use algorithms. such as FP-growth to find the set when new data is inserted into the dataset

in our project we have use a list of tools :

1) databricks to run code in cluster .
2) Pyspark as framworks of big data analysis
3) python
4) data set of job skills 50k as CSV file

The project workflow in these steps :

1) getting the Dataset We searched the dataset for job skills and we chose the data set from the paperswithcode website the data was a sample from a data set 5GB The format of data :

9978	127681	interfaces	logos	who	knowledge	html	adobe phot
9979	3094	cisco	wireless	wireless_network	airmagnet	cwna	ieee_8
9980	69440	IT Project Management	Business Process Re - engineering	Software Development Life Cycle	Project Review	Consulting	Business An
9981	65380	Bi	MS SQL	Data Warehousing	SSAS	Application Development	Data Manag
9982	114395	Hosting	HTML	Wordpress	Javascript	JQuery	
9983	97435	C#	WCF	SQL Server	Winforms	JQuery	
9984	68454	BE Studies	Clinical Project Management	Program manager	sponsor coordination	sponsor	study exe
9985	8728	medical_writing	epidemiology	publications	medical_writer	medical_write	onc
9986	66890	MBBS	Steam Sterilizers	NaN	NaN	NaN	
9987	11926	Security Business Operations	Business Operations Specialist	MIS	business analysis	Strategy planning	
9988	37047	ASP.Net	WCF	MVC	C#	SQL Server	J

2) create free account in databricks
3) create and run new cluster in databricks
4) upload dataset file to databricks and creaet notepad to do handle this data

	_c0	_c1	_c2	_c3	_c4
1	125720	HR Executive	screening	selection	Intervie
2	112708	Special Teacher	Teaching	Education	null
3	115226	consulting	fresher	IT helpdesk	Techinc
4	19805	diploma	machining	cnc m	mould
5	80208	Compensation	Benefits	HR Functions	Alm
6	64086	Storage Administrator	null	null	null
7	48468	HR Operations	Exit Formalities	Shortlisting	Screeni
8	122729	Simulink	stateflow	Matlab developer	tar

5) preprossing the data

The data extracted from the file must be processed in a way that pyspark can handle The framework cannot treat it as a normal dataframe, so the closest equivalent to this formula must be used The columns to be worked on must also be specified as a basket

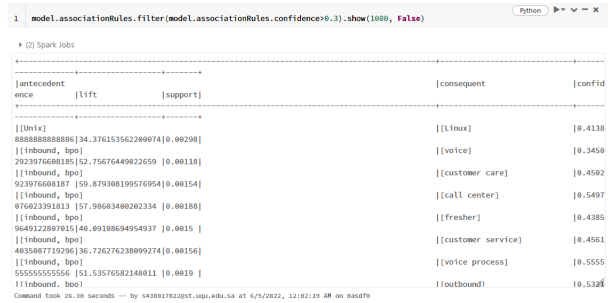
```
Python
fp = FPGrowth(minSupport=0.001, minConfidence=0.001, itemsCol='basket', predictionCol='prediction')
model = fp.fit(df_aggregated)

(3) Spark Jobs
  Job 48  View (Stages: 1/1)
  Job 49  View (Stages: 1/1)
  Job 50  View (Stages: 2/2)

Command took 7.25 seconds -- by s438017822@st.uqu.edu.sa at 6/4/2022, 3:46:54 AM on Mega10002
```

6) implement FP-Growth to find frequent itemset

here we use FP-Growth we will discuess in details about fp-growth
The FP-growth algorithm is described in the paper Han et al., Mining frequent patterns without candidate generation, where “FP” stands for frequent pattern. Given a dataset of transactions, the first step of FP-growth is to calculate item frequencies and identify frequent items. Different from Apriori-like algorithms designed for the same purpose, the second step of FP-growth uses a suffix tree (FP-tree) structure to encode transactions without generating candidate sets explicitly, which are usually expensive to generate. After the second step, the frequent itemsets can be extracted from the FP-tree. In spark.mllib, we implemented a parallel version of FP-growth called PFP, as described in Li et al., PFP: Parallel FP-growth for query recommendation. PFP distributes the work of growing FP-trees based on the suffixes of transactions, and hence is more scalable than a single-machine implementation. We refer users to the papers for more details.

spark.ml’s FP-growth implementation takes the following (hyper-)parameters:

- minSupport: the minimum support for an itemset to be identified as frequent. For example, if an item appears 3 out of 5 transactions, it has a support of 3/5=0.6.
- minConfidence: minimum confidence for generating Association Rule. Confidence is an indication of how often an association rule has been found to be true. For example, if in the transactions itemset X appears 4 times, X and Y co-occur only 2 times, the confidence for the rule X =¿ Y is then 2/4 = 0.5. The parameter will not affect the mining for frequent itemsets, but specify the minimum confidence for generating association rules from frequent itemsets.
- numPartitions: the number of partitions used to distribute the work. By default the param is not set, and number of partitions of the input dataset is used.
  The FPGrowthModel provides:
  - freqItemsets: frequent itemsets in the format of DataFrame(“items”[Array], “freq”[Long])
  - associationRules: association rules generated with confidence above minConfidence, in the format of DataFrame(“antecedent”[Array], “consequent”[Array], “confidence”[Double]).
  - transform: For each transaction in itemsCol, the transform method will compare its items against the antecedents of each association rule. If the record contains all the antecedents of a specific association rule, the rule will be considered as applicable and its consequents will be added to the prediction result



```
1  model.associationRules.filter(model.associationRules.confidence>0.3).show(1000, False)
```

(2) Spark Jobs

```
|antecedent                       |consequent          |confid
ence         |lift              |support|
|[Unix]                           |[Linux]             |0.4138
8888888888886|34.376153562200074|0.00298|
|[inbound, bpo]                   |[voice]             |0.3450
2923976608185|52.75676449022659 |0.00118|
|[inbound, bpo]                   |[customer care]     |0.4502
923976608187 |59.879308199576954|0.00154|
|[inbound, bpo]                   |[call center]       |0.5497
076023391813 |57.98603400202334 |0.00188|
|[inbound, bpo]                   |[fresher]           |0.4385
9649122807015|40.09108694954937 |0.0015 |
|[inbound, bpo]                   |[customer service]  |0.4561
4035087719296|36.726276238099274|0.00156|
|[inbound, bpo]                   |[voice process]     |0.5555
555555555556 |51.53576582148011 |0.0019 |
|[inbound, bpo]                   |[outbound]          |0.5324
```

Command took 26.30 seconds -- by s438017822@st.uqu.edu.sa at 6/5/2022, 12:02:19 AM on 0asdf0

7) add new data to model

To test the model, new data must be added and configured to be used in the model

```
1  columns = ['basket']
2  new_data = [(['html'],), ([ 'Software Engineering'],)]
3  rdd = spark.sparkContext.parallelize(new_data)
4  new_df = rdd.toDF(columns)
5  new_df.printSchema()
6  new_df.show(2,False)
```

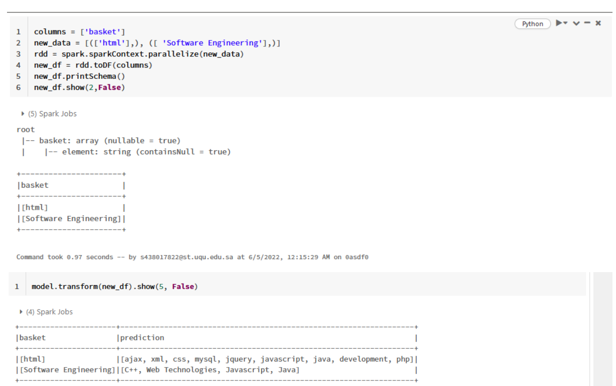
(5) Spark Jobs

```
root
 |-- basket: array (nullable = true)
 |    |-- element: string (containsNull = true)

+----------------------+
|basket                |
+----------------------+
|[html]                |
|[Software Engineering]|
+----------------------+
```

Command took 0.97 seconds -- by s438017822@st.uqu.edu.sa at 6/5/2022, 12:15:29 AM on 0asdf0

8) predict the related skills related to skills added

here we transfer new skills to see related skills in dataset , As shown in the pictures, the skills associated with the inputs appeared, and the outputs appeared based on confidence and support



```
1  columns = ['basket']
2  new_data = [(['html'],), ([ 'Software Engineering'],)]
3  rdd = spark.sparkContext.parallelize(new_data)
4  new_df = rdd.toDF(columns)
5  new_df.printSchema()
6  new_df.show(2,False)
```

(5) Spark Jobs

```
root
 |-- basket: array (nullable = true)
 |    |-- element: string (containsNull = true)

+----------------------+
|basket                |
+----------------------+
|[html]                |
|[Software Engineering]|
+----------------------+
```

Command took 0.97 seconds -- by s438017822@st.uqu.edu.sa at 6/5/2022, 12:15:29 AM on 0asdf0

```
1  model.transform(new_df).show(5, False)
```

(4) Spark Jobs

```
+----------------------+-------------------------------------------------------------------+
|basket                |prediction                                                         |
+----------------------+-------------------------------------------------------------------+
|[html]                |[ajax, xml, css, mysql, jquery, javascript, java, development, php]|
|[Software Engineering]|[C++, Web Technologies, Javascript, Java]                          |
+----------------------+-------------------------------------------------------------------+
```

This is what was done, but due to time constraints and the lack of data for the courses taught in universities, no analyzes were made showing the link of study subjects to job opportunities and the required skills related

IV. Reflection

In this project, through this term, several tools and techniques were made that increased our software and technical knowledge, and we can summarize them in the following: We used a platform that simulates distributed systems, clusterer and we used Data Brix We worked on several languages and frameworks, primarily on Python and the Pyspark framework as a task distribution system. Through the databricks, we used the built in in algorithms such as FP-Growth algorithm and the Prefix Span algorithm., which helped us to find the frequent item set and Association rule , I also learned the dealing with various files and different types of datatype and preprocessing them before analysis , so in the future I can use these tools perfectly to solve this type of problem .

.

V. Conclusion

Through big data techniques, data was mined through association rule algorithms in our case FP-GROUTH, and this thing was done through the databricks platform and in the future it will be transferred to a web application that deplayed in internet .

References

1) https://smartbridge.com/market-basket-analysis-101/
2) https://spark.apache.org/docs/latest/ml-frequent-pattern-mining.html
3) https://databricks.com/
4) https://www.oracle.com/big-data/what-is-big-data/
5) https://spark.apache.org/