Techniques To Understanding Hidden Networks In Data

Using Polyglot Persistence, Graphs, Machine Learning and Big Data on CMS Open Payments Data

Advantages of Including Machine Learning to BI

It just boils down to one word!!! "**Answers**". Yes!! Machine Learning techniques provides a different set of tools which will provide **Answers** which the traditional BI using simple statistical measures and techniques cannot. Let us focus on one of the important area of BI, Identifying patterns with in data. Patterns translates to **Answers** for questions known or Unknown. Patterns translates to **Business Opportunities**.

Traditional BI would depend on simple summary statistic like mean, median and standard deviation and try to find the initial groups of data by aggregating data based on the Business Questions. Next steps would be trying to find a reference distribution for various variables and also figure out associations between various continuous and categorical variables. This isa a rigid approach and these techniques fall short when the data is huge or number of features are huge. Also, the grouping they achieve are very broad and often fail to identify interesting patterns. Some times they just fail to work because of the shear size.

In this paper, we will explore identifying patterns using Head Tail Break and Machine Learning clustering techniques like K-Means in the CMS open payment data. when data is huge and runs into millions of records we need to group to find useful patterns hidden in the data. Further network analysis is performed on each of the group.

Initial Look At The CMS Open Payments Data

Aim of this analysis to demo pattern recognition techniques using Machine Learning and Graph analytics. !0% of the 2015 General payment data is taken to perform the analysis. Cleaned up sample data would look as follows

	id	name	payee_type	primary_type	speciality	company	amount	payment_type	drug_or_device	dd_type
0	293808	GARY DEVOSS	P	Medical Doctor	Allopathic & Osteopathic Physicians|Internal M...	GlaxoSmithKline, LLC.	20.06	Food and Beverage	ANORO	drug
1	181505	WEYMIN HAGO	P	Medical Doctor	Allopathic & Osteopathic Physicians|Internal M...	Par Pharmaceutical, Inc.	3.22	Food and Beverage	NASCOBAL CYANOCOBALAMIN, USP	drug
2	205977	MUHAMMAD SHEIKH	P	Medical Doctor	Allopathic & Osteopathic Physicians|Internal M...	Bristol-Myers Squibb Company	21.00	Food and Beverage	DAKLINZA	drug
3	237391	PETER RUMORE	P	Medical Doctor	Allopathic & Osteopathic Physicians|Internal M...	Janssen Pharmaceuticals, Inc	14.67	Food and Beverage	SIMPONI ARIA	drug
4	231874	PETER KWOFIE	P	Medical Doctor	Allopathic & Osteopathic Physicians|Internal M...	Kaleo, Inc.	11.42	Food and Beverage	Evzio	drug

Column description is given below

Column Name	Type	Description
id	Identifier Variable	Physician id and hospital id are combined in this column. This column does not participate in the data analysis.
name	Identifier Variable	This column would not participate in the data analysis. Would be used to identify the physician or hospital by name.
payee_type	Categorical Variable	This column distinguishes the type of payee. It can have two values 'H' for hospital and 'P' for Physician.
primary_type	Categorical Variable	It explains the type of physician.
speciality	Categorical Variable	It explains the speciality of the physician.
company	Identifier Variable	This variable identifies the company. It does not contribute any signal to the analysis.
amount	Continous Variable	The amount paid for a group of transactions. Grouping would be explained below.
payment_type	Categorical Variable	It explains the nature of the payment paid.
drug_or_device	Identifier Variable	This variable identifies the drug or device. It does not contribute any signal to the analysis.
dd_type	Categorical Variable	This column distinguishes if the transaction is for a device or a drug. Possible values are 'drug', 'device' or 'no_drug_device'

Business Question and EDA Strategy

In this analysis instead of focusing on each individual transaction we wanted to understand **How much money each physician or hospital was paid for each drug or device and how many times?** To understand this, instead of taking individual transactions we consider the following grouping

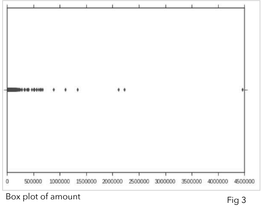
```
aggregations = {
    'amount': {
        'amount': 'sum',
        'n': 'count'
    }
}
dfg1 = df.groupby(['name','drug_or_device','company','primary_type','payment_type','speciality','dd_type','payee_type']).agg(aggregations).reset_index()
print(dfg1.count())
dfg1.head()
```

All the transactions are grouped by the name of the physician, the drug or device he has received money for, company which made the payment, primary type of the doctor, the type of payment made by the company. Total amount and the frequency of the number of transactions is computed for each grouped transaction.

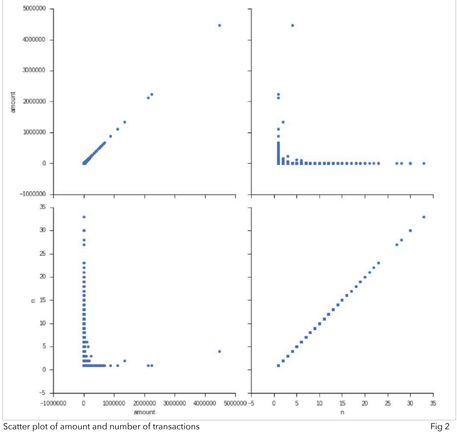
Initial look at the data reveals some interesting facts, there are 4.5k records in this sample. Range is from few cents to 4millions. 75% of the data has an amount of $35. ~4k records are only paid once. High frequency transactions are paid smaller amounts. It is an **indication that companies made sure that they don't pay high value transaction multiple times.**

	id	amount	n
count	448971.000	448971.000	448971.000
mean	316022.879	243.641	1.149
std	319699.407	9345.732	0.553
min	6.000	0.010	1.000
25%	111721.000	11.800	1.000
50%	225230.000	16.340	1.000
75%	338320.000	35.000	1.000
max	1400579.000	4462319.000	33.000

Standard Summary details Table3



Box plot of amount Fig 3

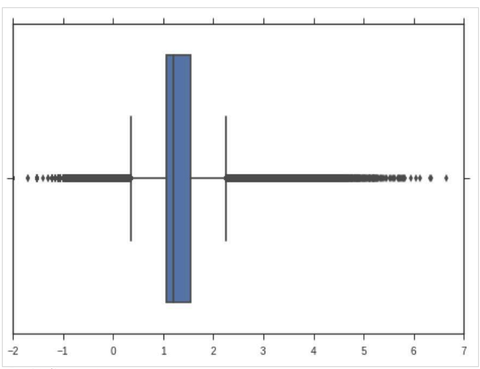


Scatter plot of amount and number of transactions Fig 2

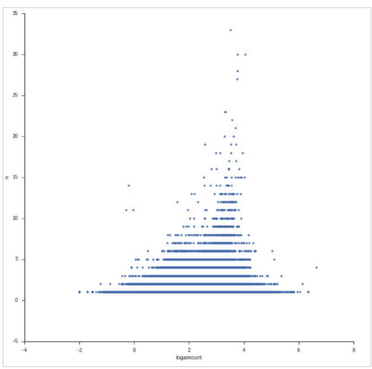
1	400470
2	37935
3	7054
4	1898
5	734
6	359
7	174
8	107
9	72
10	43
11	32
12	27
13	20
14	11
15	9
16	6
18	4
19	3
30	2
17	2
20	2
23	2
21	1
22	1
27	1
28	1
33	1

Frequency and Count of Transactions Fig 6

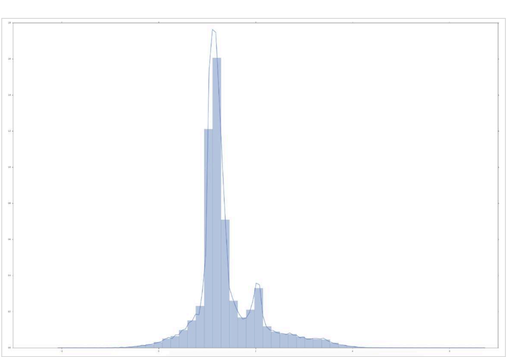
Looking at the box plot of the amount, it was decided to try out the **log amount instead of the actual amount**.



Box plot of log amount Fig 4



Scatter plot of log amount and number of transactions Fig 4.3



Histogram of log amount Fig 1

Pharmaceutical Spend data shows a huge concentration of data points close to zero and it gets sparse at higher amounts. In the histogram we can look at multiple peaks. We can infer that there is an overlap of more than one distribution in the data. It has all indications of an existence of a long tail. To explore it further Exploratory Data Analysis would **focusing on catching the long tail.**

There are many advantages of focusing on long tails when you see one. They are very good at identifying **business opportunities which has lower competition or ignored by the competition and has greater opportunity**. It can **answer** the following questions.

- Why are few doctors are paid large amounts of money very few times and only few hospitals are involved?
- Find all the drugs and devices which has least competition and look for hidden demand or opportunity.
- May be traditional methods or normal supply chain with its inherent limitation is not able to capitalize on the demand existing on the tail. It is time to think out of the box. Use new business models.
- Budgets are limited and often get allocated to the drugs and devices which are safe and have a high demand. Because of the nature of high demand there are many competitors and the profit margins are very less. Long tail gives you a chance to look at the drugs and devices where the competition is less and the demand is latent. There is risk and also sufficient reward.

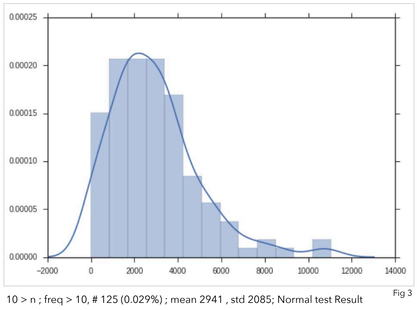
Next step in our EDA would be to stratify the data first using the frequency. We can see that there are less transactions where the frequency is high. To start with let us consider all the transactions which has a frequency greater than 10 as high frequency and rest of it as low frequency.

Techniques To Understanding Hidden Networks In Data

Using Polyglot Persistence, Graphs, Machine Learning and Big Data on CMS Open Payments Data

Catching the Long Tail

Divide the data into two parts based on the high frequency n > 10 and low frequency n <= 10. We clearly see that low frequency shows all the characteristics of a long tail.



10 > n ; freq > 10, # 125 (0.029%) ; mean 2941 , std 2085; Normal test Result (statistic=34.474261382349553, pvalue=3.2659479345919546e-08)

Fig 3

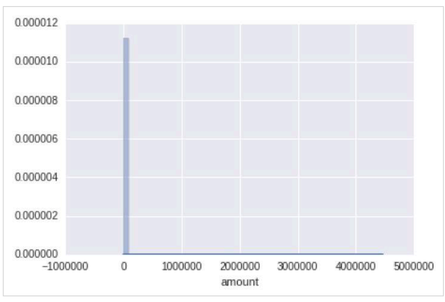
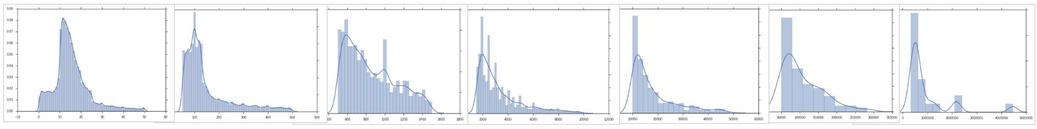


Fig 8

Further we stratify based on trail and error using the percentile values obtained from the box plots and summary statistics. On another interesting fact that all the transactions below 1000 accounted for 20.97% of the amount and the transactions greater than 1000 account for 79% of the amount. These characteristics confirms a long tail for the low frequency data.

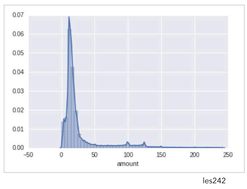


amount < 50	50 > amount < 500	500 > amount < 1500	1500 > amount < 10000	10000 > amount < 50000	50000 > amount < 300000	amount > 300000
freq <10	freq <10	freq <10	freq <10	freq <10	freq <10	freq <10
# 351555 (78.3%)	# 71400 (15.9%)	# 12335 (2.75%)	# 10928 (2.43%)	# 664 (0.15%)	# 112 (0.025%)	# 21 (0.005%)
mean 15.6	mean 149.209	mean 856.488	mean 3322	mean 18029	mean 107319	mean 938884
std 8.512	std 99.500	std 267.120	std 1768	std 8817	std 54165	std 965792

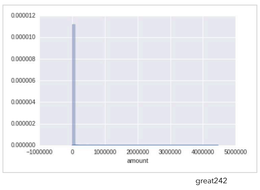
Fig 5

Using Head Tail Breaks to analyze the long tail

Head Tail Breaks method iteratively divides head and tail part by cutting the data at the mean. If a part shows long tail characteristics it will be further cut it at the mean. This method captures the hierarchy. This method is shown below.

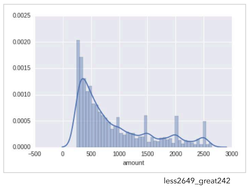


les242

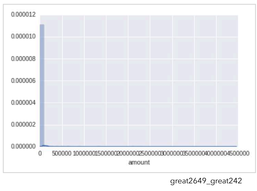


great242

Level 1
mean @ 242.89012534366037 of low frequency dataframe;
groups : "less242" and "great242";
"great242" exhibits long tail behavior. This will be further divided.

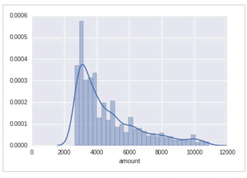


less2649_great242

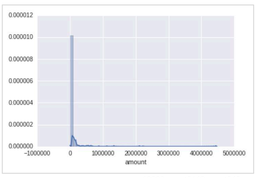


great2649_great242

Level 2
mean @ 2649.814199961778 for "great242";
groups : "less2649_great242" and "great2649_great242";
"great2649_great242" exhibits long tail behavior. This will be further divided.



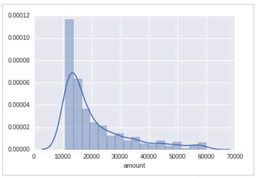
less10900_great2649_great242



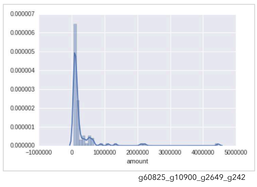
g10900_great2649_great242

Level 3
mean @ 10900.115461790912
for "great2649_great242";
groups : "less10900_great2649_great242" and "g10900_great2649_great242";
"g10900_great2649_great242" exhibits long tail behavior. This will be further divided.

Level 4
mean @ 60825.42906647805
for "g10900_great2649_great242";
groups : "l60825_g10900_g2649_g242" and "g60825_g10900_g2649_g242";
"g60825_g10900_g2649_g242" exhibits long tail behavior. This will be further divided.

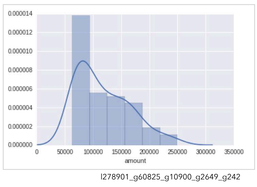


l60825_g10900_g2649_g242

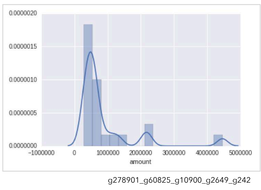


g60825_g10900_g2649_g242

Level 5
mean @ 278901.3335779816
for "g60825_g10900_g2649_g242";
groups : "l278901_g60825_g10900_g2649_g242" and "g278901_g60825_g10900_g2649_g242";
Both groups stopped exhibiting long tail behavior and the cut would stop here.



l278901_g60825_g10900_g2649_g242



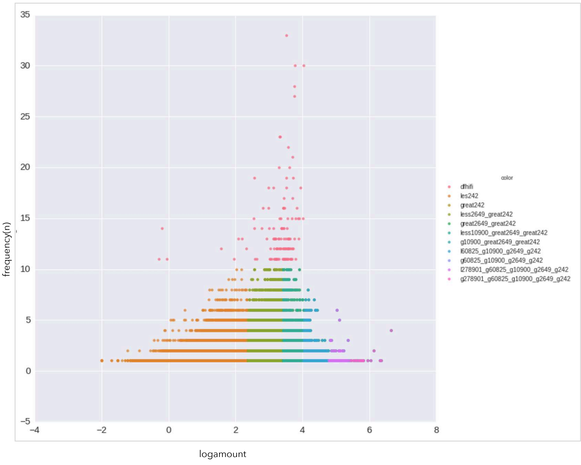
g278901_g60825_g10900_g2649_g242

Visualizing the results

les242	412329
great242	36517
less2649_great242	30118
great2649_great242	6399
less10900_great2649_great242	5692
g10900_great2649_great242	707
l60825_g10900_g2649_g242	598
dfhifi	125
g60825_g10900_g2649_g242	109
l278901_g60825_g10900_g2649_g242	86
g278901_g60825_g10900_g2649_g242	23

We combined all the dataframes from various groups and took the value counts for each group. There is an interesting pattern. The tail part of the previous group and the head part of the next group is very close and also only the tail part of each group exhibited long tail behavior.

Took the log of the amount and frequency and tried few visualizations. If formed like a frequency pyramid. only high frequency at the top. At lower frequency you have transactions of all amounts but as the frequency started to increase values started to tapper off. This show how frequency and amount are distributed.



Above we have seen how head tail break method has broken the data into various interesting groups. Next, we will use Machine Learning to find various clusters. All these clusters would be further analyzed to find interesting patterns.

Techniques To Understanding Hidden Networks In Data

Using Polyglot Persistence, Graphs, Machine Learning and Big Data on CMS Open Payments Data

Using Machine Learning To Identify the Clusters

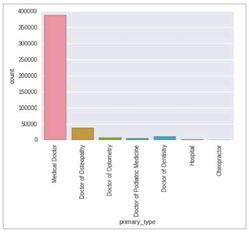
First step in using ML to identify the clusters is to encode all the categorical variables. ML based algorithms uses euclidean based distance and it cannot handle categorical variables. we will use one hot encoding and convert the categorical variable into numerical variables. After encoding the following using one-hot encoding

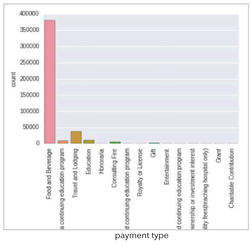
- primary type
- payment type
- device or drug type
- payee type, hospital or doctor

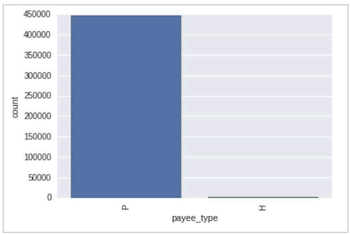
We will have the following variables

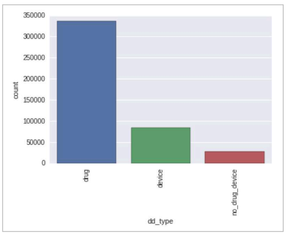
```
Data columns (total 39 columns):
id                          448971 non-null int64
drug_or_device              448971 non-null object
company                     448971 non-null object
primary_type                448971 non-null object
payment_type                448971 non-null object
speciality                  448971 non-null object
dd_type                     448971 non-null object
payee_type                  448971 non-null object
amount                      448971 non-null float64
n                           448971 non-null int64
name                        448971 non-null object
logamount                   448971 non-null float64
chiropractor                448971 non-null float64
dentistry                   448971 non-null float64
optometry                   448971 non-null float64
osteopathy                  448971 non-null float64
podiatric                   448971 non-null float64
hospital                    448971 non-null float64
medical_doctor              448971 non-null float64
charity                     448971 non-null float64
other_consulting            448971 non-null float64
comp_non_accredited         448971 non-null float64
comp_accredited             448971 non-null float64
consulting                  448971 non-null float64
ownership                   448971 non-null float64
education                   448971 non-null float64
entertainment               448971 non-null float64
food                        448971 non-null float64
gift                        448971 non-null float64
grant                       448971 non-null float64
honoraria                   448971 non-null float64
royalty                     448971 non-null float64
rental                      448971 non-null float64
travel                      448971 non-null float64
device                      448971 non-null float64
drug                        448971 non-null float64
no_drug_device              448971 non-null float64
paid_hospital               448971 non-null float64
paid_physician              448971 non-null float64
```

A look at how the values are distributed for all these variables.



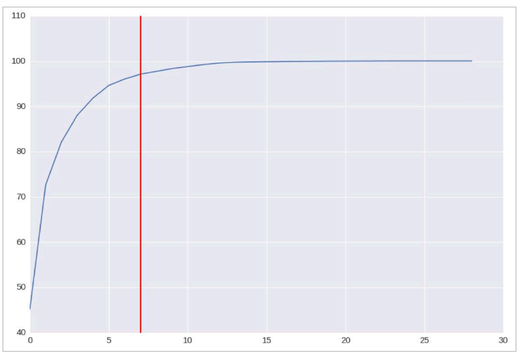




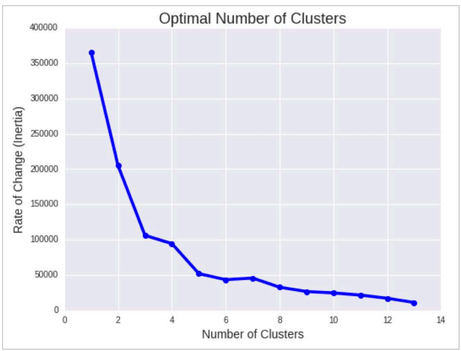


Upon visual inspection of the categorical variable we find that the most common transaction is a payment for the primary type medical doctor, payment type of food and beverage for the payee type physician and for a drug. Because there is very little variable in the data after one-hot encoding further the variance is distributed among various column we expect the influence of the categorical variables to be minimum. Log amount and the frequency is expected to be the two main features.

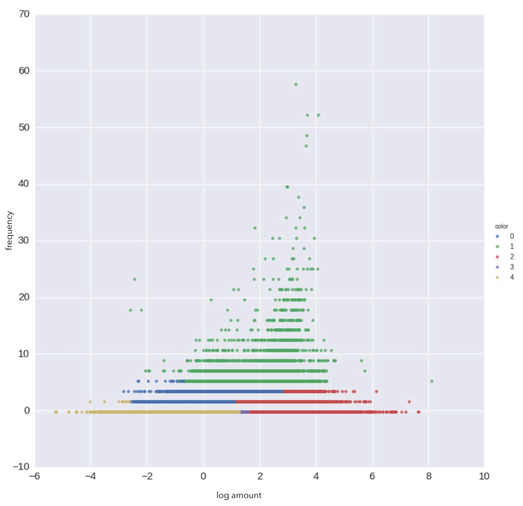
We have a total of 29 dimension of numerical variables. We will use PCA to identify the principal components and reduce the dimensionality. Before we use PCA we have to scale the attributes. We plan to scale log amount and frequency and then run PCA. Scree plot for PCA is given below



7 Principle components explain 96.03% of the variance. We will consider 7 features. Next step is to perform the clustering using k-means algorithm. We have to identify the optimum number of clusters.



The scree plot indicates that the optimum number of clusters would be 5. We will run the k-means algorithm with 5 clusters and plot the results



k-means clustering has grouped the data into 5 groups. We have 11 groups identified using Head Tail Break. Why are these groupings are important for identifying the patterns? With out these grouping there is huge data and summary statistic like mean, median, standard deviation does not give out much useful information. Few numbers cannot describe the huge data. K-means uses Euclidian distance is used as a similarity measure to group the data.

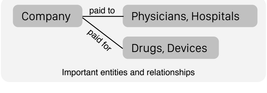
Once the groups are identified, each group should be studied independently to identify interesting patterns. One such mechanism is to use network analysis. Inherently this data has the following relationships

- One company can pay multiple physicians and multiple hospitals.
- One physician can be accepting money from multiple companies.
- One company might be investing in one or more device or drug.
- One or more physician might be working on one or more device or drug.

Company, Physician, Teaching hospitals, Drugs and Devices are the various nodes of a graph. These nodes are connected with each other and form networks. These networks can be studied to reveal important **answers**. If we do not group the data and try to create a huge network component, it would look like a huge hair ball and would not provide any insight. Clustering allows grouping of data and provides graphs with fewer meaningful nodes.

Role of Networks and Centrality Measures To Understand The Data

Basic graph we plan to address is as follows



Company pays Physicians and Hospitals for Drugs and Devices. We will be interested to know the following questions related to a connected component in the network.

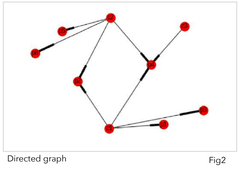
- ✓ which companies, doctors or drugs are central to the component?
- ✓ which companies, doctors or drugs are most influential?
- ✓ which companies, doctors or drugs are most powerful?

There are two important measures

a. How much money paid by a company to a physician or drug?
b. How many times a physician or drug is paid by the company?

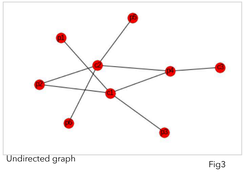
- ✓ Amount and frequency should be considered as a edge weight in determining the centrality measure.
- ✓ A company is paying a **lot of money** to a physician means the company is seeking that physician and it is an indication that the physician is very important for that company.
- ✓ If a company is paying a physician **many times**, it is an indication that the company is trying to contact the physician many times. In this case, the amount is not important it is the frequency. Company is making a serious effort to get the attention of the physician to promote the drug.

Let us understand the network measures like degree, betweenness and pagerank using following graphs.



Directed graph

Fig2



Undirected graph

Fig3

High degree in an undirected graph means the node is highly connected (Fig3). Node P4 is a physician, paid by companies C3, C2, C1. Node P4 when compared to P6 which is paid only by C1, P4 is more influential than P6. P4 has a degree of 3 and P6 has a degree of 1. **Degree** is centrality measure which is an indication of **power**.
Betweenness is an indication that this node lies in between other pair nodes. Companies generally has high betweenness score. They connect many physicians. In a connected component, companies which have a high betweenness score indicates **high influence**.

Techniques To Understanding Hidden Networks In Data

Using Polyglot Persistence, Graphs, Machine Learning and Big Data on CMS Open Payments Data

Physician who has high **betweenness** score indicates they are connected with many companies. It is an indication that they are highly sought out because they can **influence** the drug. More connections to a node means more votes to the node and higher the value of PageRank. Each vote is weighted by the PageRank of each linking node. This weight is governed by the amount of links that referring node has.
In Fig.3 C2 and C1 has highest PageRank because it is voted by nodes P2 and P4. In turn the PageRank of C2 and C1 influence the score of P2 and P4. In the case of undirected graph all the companies would have highest PageRank scores, followed by those physician who are connected to many companies.

Why are we considering an undirected graph in reality a directed graph makes more sense? Let us explore.
Companies pay Physicians and it is not the other way round. So it is directed, when it is a directed graph all these measures have a quite a different values and interpretations. They can no longer depend on the degree but they have in-degree and out-degree. PageRank depends on the In-degree.

	name	degree	pagerank	betweenness
0	p2	2	0.106901	0.160714
1	p4	3	0.159475	0.410714
2	p3	1	0.061784	0.000000
3	c1	4	0.212318	0.500000
4	p6	1	0.061784	0.000000
5	p1	1	0.061784	0.000000
6	c2	4	0.212318	0.500000
7	c3	1	0.061852	0.000000
8	p5	1	0.061784	0.000000

Undirected graph measures Table1

	name	degree	pagerank	betweenness
0	p2	2	0.123377	0.0
1	p4	3	0.196969	0.0
2	p3	1	0.104978	0.0
3	c1	4	0.086580	0.0
4	p6	1	0.104978	0.0
5	p1	1	0.104978	0.0
6	c2	4	0.086580	0.0
7	c3	1	0.086580	0.0
8	p5	1	0.104978	0.0

Directed graph measures Table2

Looking at Fig.2 and Table 2, it is clear that Companies would be having very less PageRank score. Reason is their in-degree is zero. Physicians would have High PageRank score because they in-degree is very high and out-degree is zero. They act like a **Rank Sink**.

PageRank follows the **Random Surfer** model. Random Surfer makes hops along a path. At each hop, the probability of choosing the next edge is its total weight divided equally among all edges.
From Fig.2, you can infer all the physician nodes are Rank Sinks, their out-degree is zero. When the Random Surfer encounters a sink it uses the concept of teleportation to find the next hop and weight of the edge does not have any effect.
Weight has effect when the node is company node. But, the problem with the company node is, its in-degree is zero and its PageRank is very small. It does not provide correct inference. Hence we chose to use the undirected graph for this analysis

For every highly connected cluster following analysis would be performed
1. Degree, it does not consider the weight. It identifies the central actors by their degree. Higher degree indicates higher influence.
2. Betweenness, it does not consider the weight. It identifies the key actors based on the connections they make. Companies would have a high betweenness if they pay more physicians. On the other hand physicians would have a big value if they are paid by many companies.
3. Finally, PageRank would be computed. We will computer three values
   a. PageRank without any weights
   b. PageRank with amount paid as edge weight
   c. PageRank with Frequency of payments as edge Weight

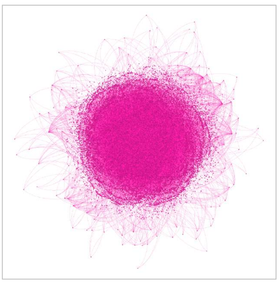
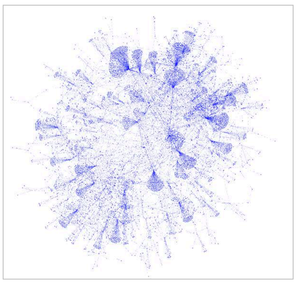
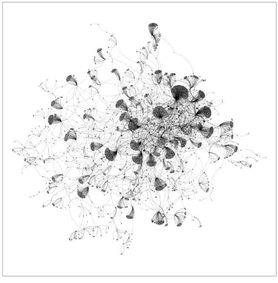
In the case of highly connected nodes we were more interested to know with in a cluster or a group

- who is more popular when amount is considered as an important driver?
- who is more popular when frequency is considered as an important driver?
- which nodes are more influential?
- which nodes are more powerful?

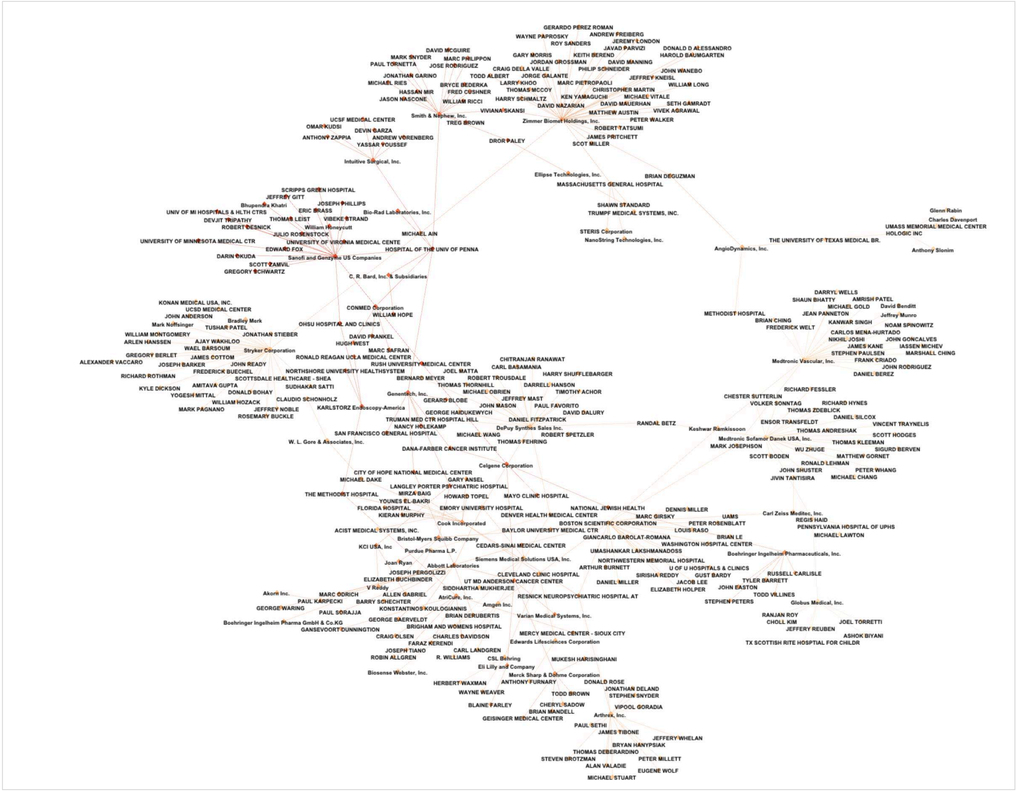
When the network is not highly connected, we can visually examine and understand the nature of various nodes. In addition to these measures we can also try to find answers to the question, why only few companies and few physicians are involved with a drug? Why such a drug is worked in isolation? Is there a hidden business opportunity?

Let us look at few example graphs

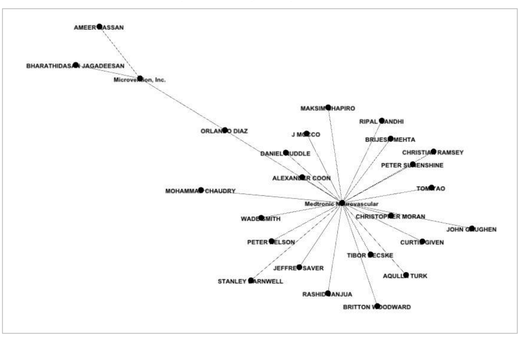
Hair Balls, looks good but reveal less useful information.

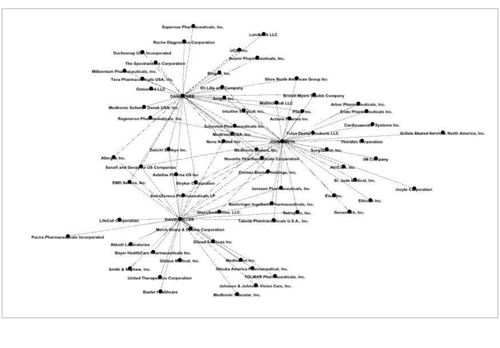




Applying filters to a giant component to get answers



Small Isolated connected components easy to explore





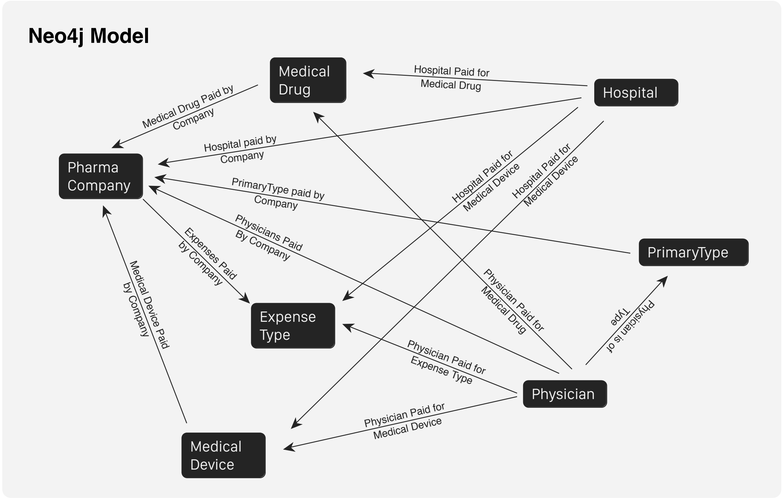
Techniques To Understanding Hidden Networks In Data

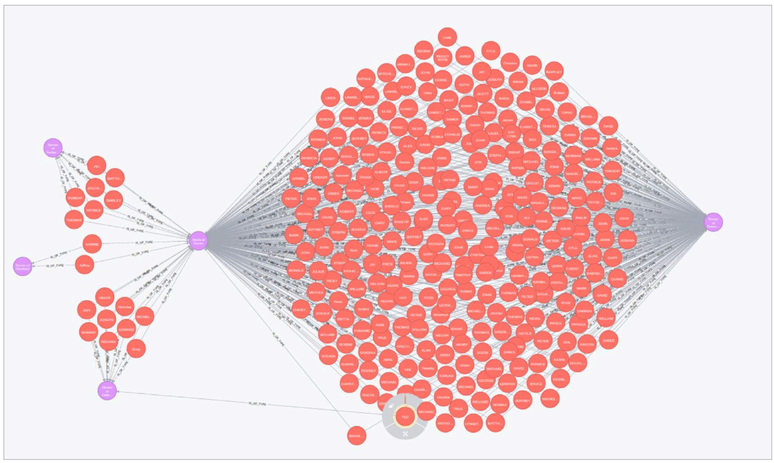
Using Polyglot Persistence, Graphs, Machine Learning and Big Data on CMS Open Payments Data

Polyglot Persistence - Advantages of Storing And Enriching The Data Based On The Specific Use case

Companies have lots of data stored in a traditional RDBMS or a Dataware House. Lot of money is spent on collecting, cleaning, transforming and storing it. Often it is used to generate summary statistics. These numbers are important but does not allow to discover the answers needed. RDBMS has its rigid structure which is helpful with certain use cases. In this section, we will present a simple architecture which allows to surf through the networks for answers. Also present a strong case for using polyglot persistence to achieve a better architecture.

In the previous section, we have presented network connect graphs. These graphs do not use all the nodes. It primarily focuses on a simple graph between companies, teaching hospitals, physicians, drugs and devices. It is very efficient to calculate the centrality measures. Once we have the centrality measures we would like to have the complete context. In order to achieve this we propose to store all the network information in a graph database like Neo4j. Presenting below is a graph model.





Storing the network data along with the centrality measures is more efficient than storing it in RDBMS. Cypher queries can help you to quickly explore the network. Clustering would help you to focus on a specific group and the centrality measures like degree, betweeness and pagerank would allow you to zero in on specific nodes quickly. These allows to explore answers to questions like

- ✓ Compare and Explore the competitors network quickly.
- ✓ Friend of a Friend relationship. Find physicians who are working for you and also your competitor.
- ✓ Shortest path relationships
- ✓ All other questions which involve navigation from one node to another.

Neo4j is a property graph. So we can capture various centrality measures and other attributes as properties for nodes and edges. This will help in developing advance queries.

System would be far efficient if it has an effective search. There might be a case where you are reaching a specific node or a cluster. Efficient search would make a huge difference. All the nodes and the attributes should be indexed using Solr Search.

Once the user reaches a node and want to explore accessing all the data related to the node using a document database like Mongo DB is more efficient than from a RDBMS. Storing data in multiple specialized databases and providing access to data using an app would make exploring the data to get answers is far efficient. It is easy to update and easy to maintain. This architecture would yield greater benefits. Apps can scale easily, they would be nimble to change with less rigid schema restrictions and easy to maintain.

Use polyglot persistence to develop apps to explore the data in addition to the static reports based on summary statistics and limited drill-down.

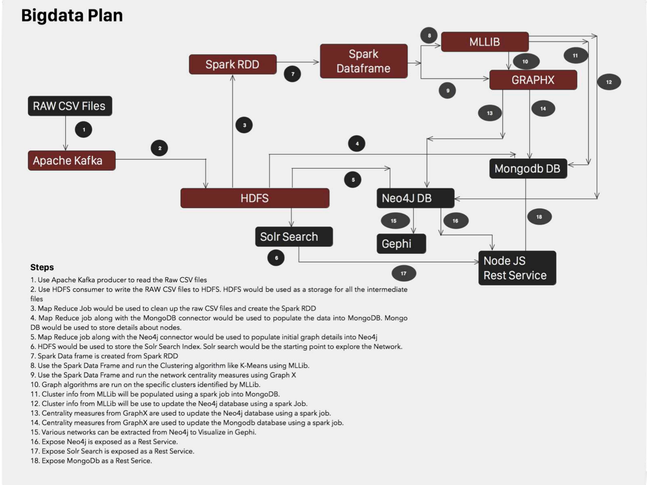
More the Data for More Answers

In this analysis we just used only CMS spend data which is publicly available. If this is combined with the following, analysis would yield more meaningful answers

- Physician Data
- Drug Data
- Company Profile Data
- Competitor Analysis Data
- Other related public data like medicare and medicaid data
- Internal transactional data

Right Tool For Right Size Of The Data - Reason for Big Data Architecture

Only 10% of the general payment data was used to show various techniques to find answers hidden within the data. CMS open payment data in addition to all other related data is huge to fit into memory of even expensive machines. We need bigger size tools to handle big data. Following is an architecture to address it.



This architecture is deliberately has little bit of redundancy to handle various kinds of datasources. This architecture offers a better ETL workflow and also an efficient way to store and present the data.