Project 2

Project Members

Sahil Puri (3114-1301)

Sethuraman Sundaraman (1132-1142)

Dataset

We used both the data set provided in the project statement:

- Netflix Dataset
- Enron Dataset

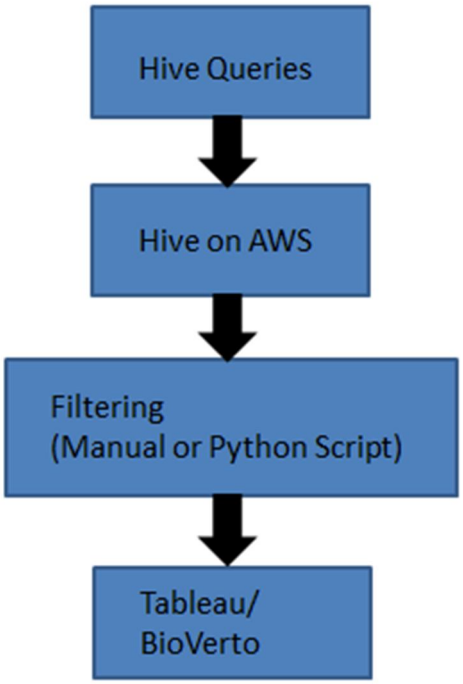
Group Member Contribution

- Sahil Puri: Basic queries on the Netflix data set, visualizing the results. An attempt to create Association Rule Mining on the Netflix dataset.
- Sethuraman Sundaraman: Queries on the Netflix data set and visualizing the results.

Interesting Resources

http://shop.oreilly.com/product/0636920023555.do

Data Processing Pipeline



- Our Sql queries were directly executed on Hive.
- A machine with Hive was provisioned using AWS.
- The data was the filtered by either manual screening it. For Hypothesis 4 of Enron Dataset python script was used.
- All Hypothesis were then generated in Tableau except Hypothesis 4 of Enron Dataset, where Bioverto was used.

Problems Faced, workarounds and Lessons Learnt

- Absence of sub-queries
  Use of sub queries was restricted to 'from' clause. Though it is even supported in the 'where' clause however the operators that can be used is only EXISTS or IN. Thus frequent queries like
  SELECT *
  FROM <table1>
  WHERE <colomn><=(SELECT * from <table2>)

  The workaround involves creating a new table using cross join of <table1> and <table2>, followed by a projection operator. Note we can even have an inner join of the two tables, if the operator for '=' instead of '<='. Inner join does not support greater than, less than or not equal to operators.
- Absence of finding row numbers
  Row numbers were needed to serve as unique ID in some queries in Association Rule Minning. We used concatenation of strings to generate new ID's. Initially all rows have a ID, all new rows were a combination of 2 existing rows, thus are ID was *id1+"0"+id2*
- Absence of set operations like INTERSECTION, UNION and EXCEPT
  Use of inner and outer joins can serve as alternatives for these operations. Please refer to the next point for, why this option was not adequately tested.
- Problems with JOINS on the version in hive installed on AWS
  The Version of Hive available on AWS had problems with use of joins, specially self joins. Some joins never worked (The Were Stuck on the reduce task while executing the query) while others worked on re-provisioning the cluster. The problem is listed in the following links:

  https://issues.apache.org/jira/browse/HIVE-4222
  https://issues.apache.org/jira/browse/HIVE-3739

The above problems and drawbacks allowed us to make only very little progress on Association Rule Mining. We have attached the source code we could complete. It generates all the frequent Items in the basket.

Netflix Dataset

Loading Data

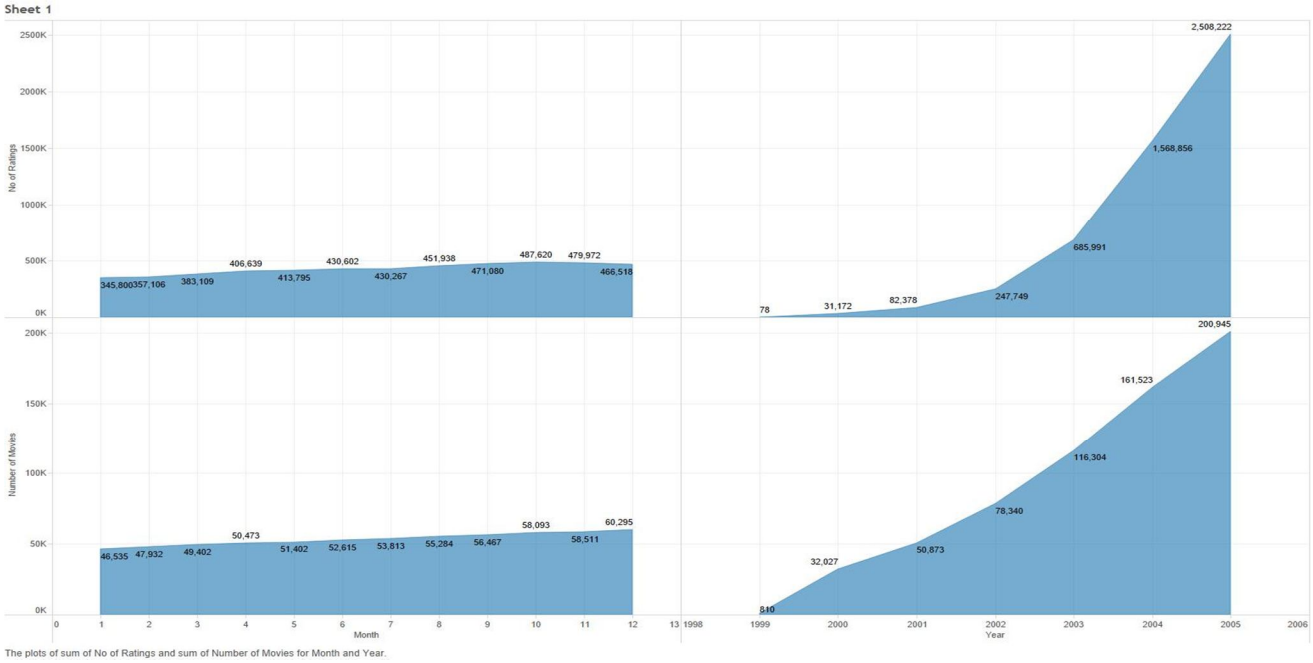
```
CREATE TABLE movie_ratings_raw(mid INT,cid INT,rating INT,date ARRAY<INT>)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
COLLECTION ITEMS TERMINATED BY '-'
LOCATION 's3n://spring-2014-ds/movie_dataset/movie_ratings/'

CREATE TABLE movie_title(mid INT,year INT,name String)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LOCATION 's3n://cis6930dic/data2/'
```

Hypothesis 1

The number of reviews will increase as the internet gets popular, also holiday seasons will have more active users.

```
INSERT OVERWRITE DIRECTORY 's3n://cis6930dic/out4/'
select date[0] as year,date[1] as month,count(distinct mid) movies,count(distinct cid) ratings
from movie_ratings_raw
group by date[0],date[1]
order by year,month
```



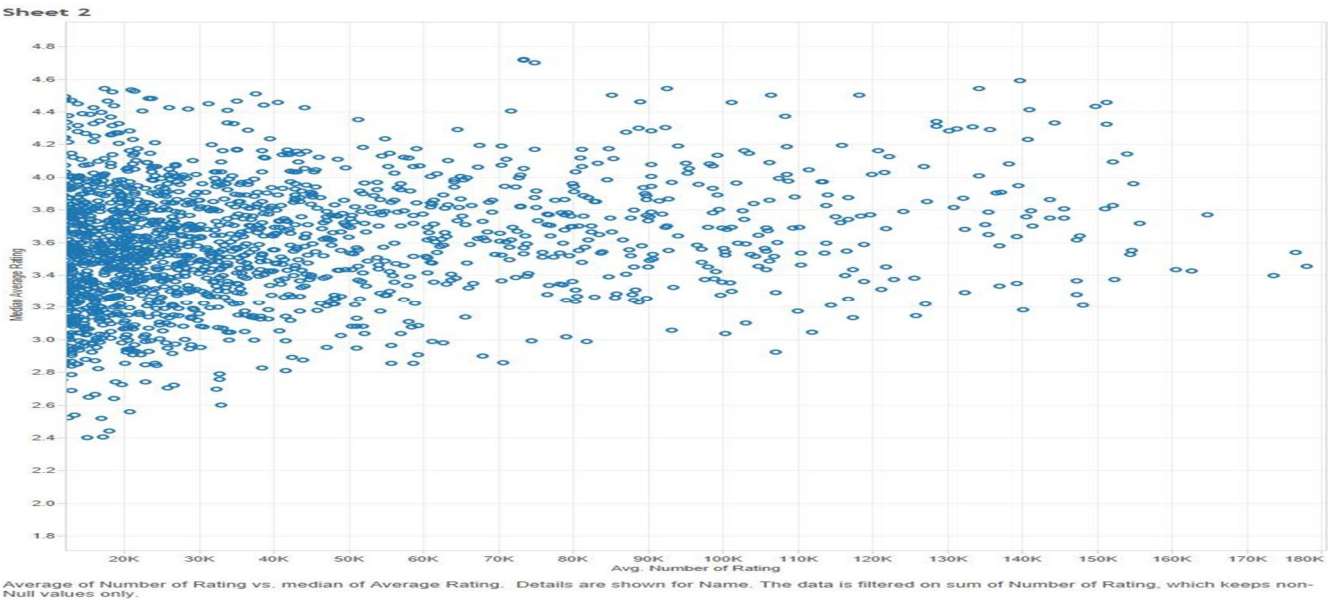
As seen the number of ratings and movies given increases as year proceeds and internet gets more popular. During Christmas and things giving there is an upsurge

Hypothesis 2

We consider the best movie to have the highest average rating, while the popular movie being a movie that receives the most reviews. Number of users who giving ratings will mostly be comparatively less as it is difficult to receive opinions from customers.Popular good movies will be on the upper right corner

```
create table ratingVsPopular as
Select mid,count(distinct cid) popular,
avg(rating) rating
from movie_ratings_raw
group by mid
order by popular
```

```
INSERT OVERWRITE DIRECTORY
's3n://cis6930dic/out1/'
Select t1.mid,t2.name,t1.popular,t1.rating
from ratingVsPopular t1 inner join movie_title t2
on t1.mid=t2.mid
```



We see a bigger cluster of number of ratings towards the left of the graph, thus fewer movies are evaluated by more people. The shawshank redemption well known popular and hit movie is in the upper right corner

Hypothesis 3

If the highest rated film of every month, of every year is listed, a potential movie which will be highest ranked movie of Netflix of all time and be identified.

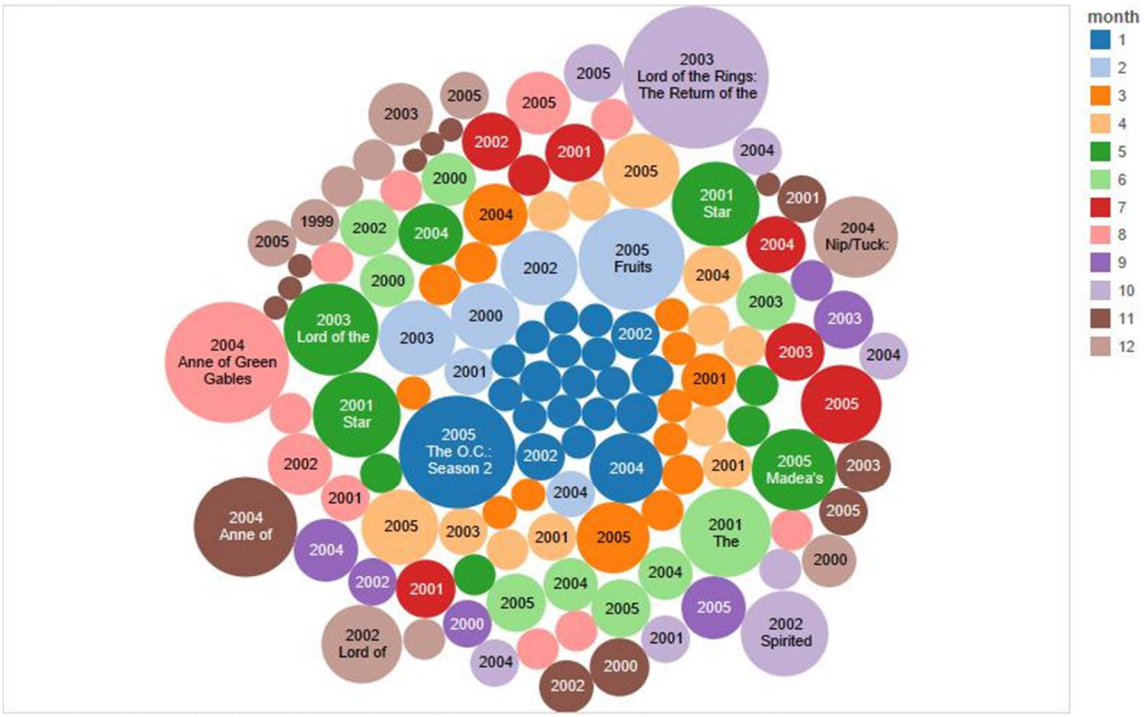
```
create table monthly_movie as
select date[0] as year,date[1] as month,mid,avg(rating) as
rating,count(cid) popular
from movie_ratings_raw
group by date[0],date[1],mid
order by popular
create table best_monthly as
select m.year,m.month,m.rating,m.mid,m.popular
from monthly_movie m inner join high_ranks h
on m.year=h.year and m.month=h.month and
m.rating=h.rating
order by m.year,m.month;
create table monthly_best as
select m.year,m.month,m.rating,m.mid,m.popular
from best_monthly m inner join high_popular h
on m.year=h.year and m.month=h.month and
m.popular=h.popular
order by m.year,m.month;
```

```
create table high_ranks as
select year,month,max(rating) rating
from monthly_movie
group by year,month

create table high_popular as
select year,month,max(popular) popular
from best_monthly
group by year,month

INSERT OVERWRITE DIRECTORY
's3n://cis6930dic/out3/'
Select
m.year,m.month,m.rating,t.name,m.popular
from monthly_best m inner join movie_title t
on m.mid=t.mid
```

Sheet 1



Year and name. Color shows details about month. Size shows sum of popularity. The marks are labeled by year and name.

We find that The Lord of the Rings: The Return of the King is seen in multiple months being rated as the best rated movie (ties are broken based on popularity). This movie is #9 according to the website http://www.100thingsilearned.com/netflix.php. Similarly we see Annie Hall which ranks #142

Note we have data only till 2005.

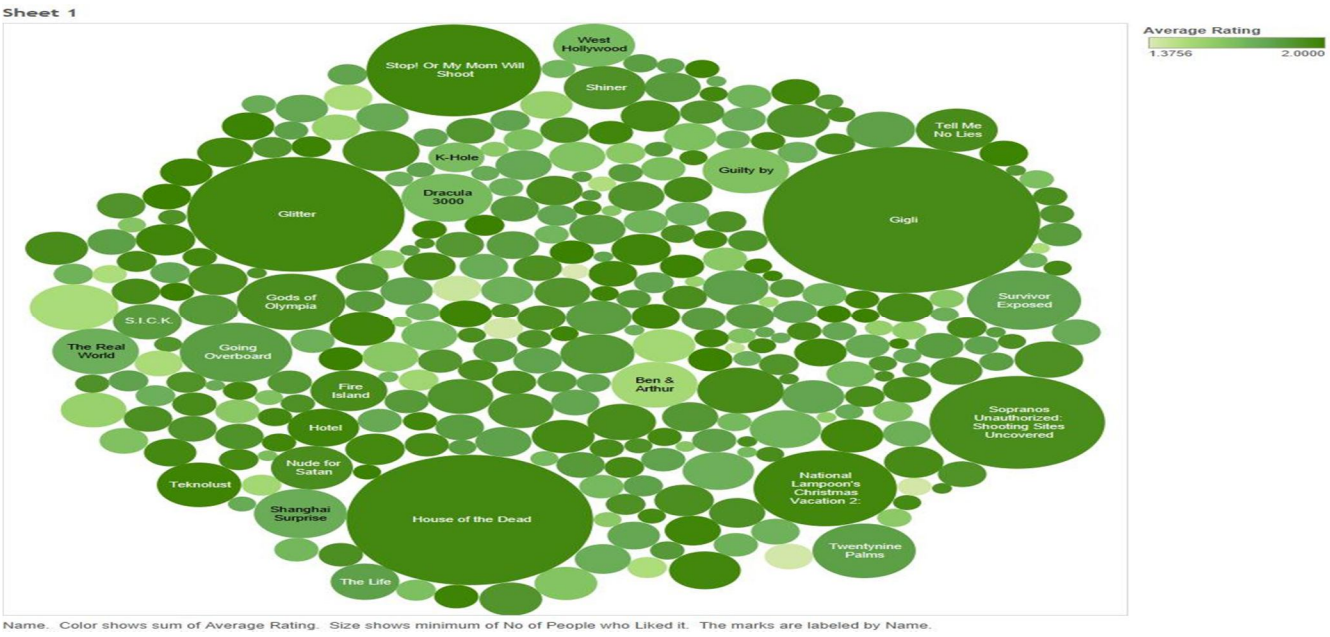
Hypothesis 4

Most of the low rated movies (movies having a rated below 2) will not have many coustomes who loved the movie(Rates who gave the movie 5/5)

```
create table high_rating as
Select mid,cid,rating
from movie_ratings_raw
where rating=5
```

```
INSERT OVERWRITE DIRECTORY
's3n://cis6930dic/out6/'
Select t2.mid,t2.name,t1.no,t1.rating
from hallOfSHame t1 inner join movie_title t2
on t1.mid=t2.mid
```

```
Create table hallOfSHame as
select h.mid,r.rating,count (distinct h.cid) no
from high_rating h inner join (select * from
ratingVsPopular where rating<=2) r on
h.mid=r.mid
group by h.mid,low_rating
```



The brightness of the colors represents relative average scores, i.e. the darkest colors score closer to 2 and lighter closer to 0. The size of the bubble is the people who love the film.

We see all larger bubbles are dark green. Lighter color movies are loved by fewer people.

Enron Data Set

Loading Data Set

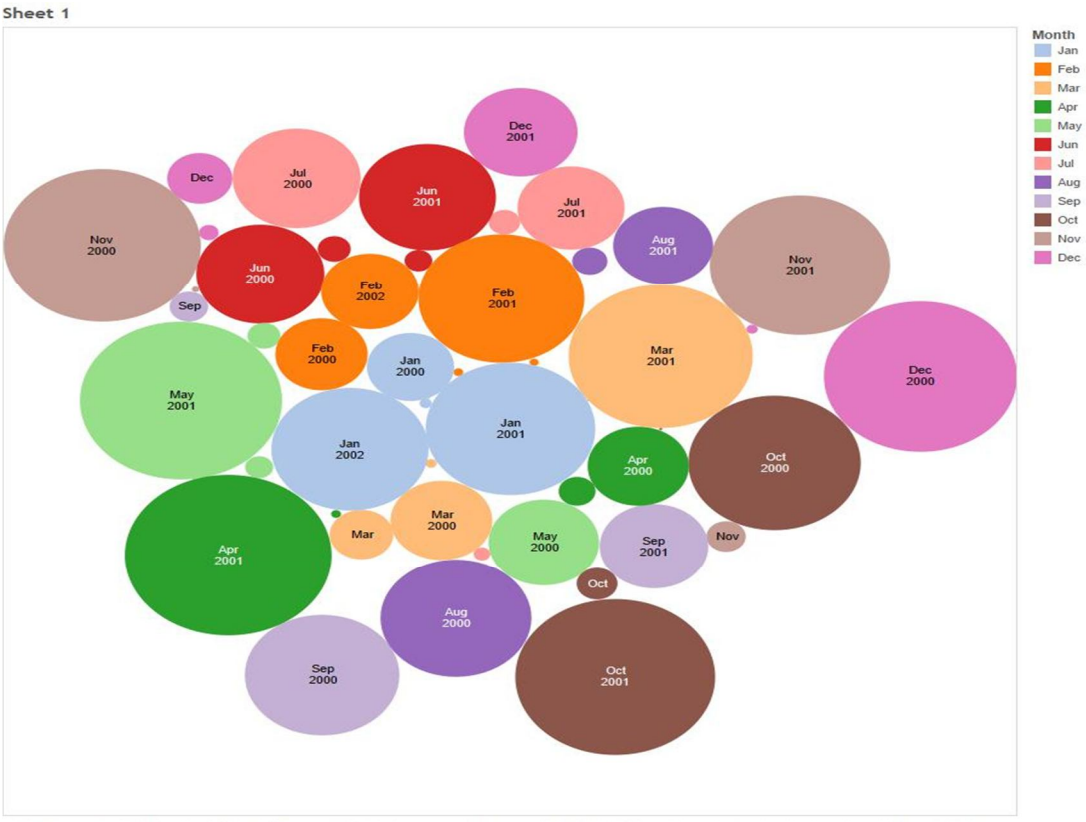
```
CREATE TABLE enron(id string, times string,fromheader String, toheader Array<String>,cc
Array<String>,Subject String, context String)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
COLLECTION ITEMS TERMINATED BY ','
LOCATION 's3n://spring-2014-ds/enron_dataset/';
```

Hypothesis 1

The number of communications surges when the time approaches the scandal.

```
Create Table enron_to as
select id, times, from, enron_to
From enron LATERAL VIEW explode(toheader) etab as enron_to;

insert overwrite directory 's3n://cis6930dic/enronq7/'
select regexp_extract(times,'([a-zA-Z]{3}, )([1-3]?[0-9] )([a-zA-Z]{3} )([0-9]{4} )(.*?)',4),
regexp_extract(times,'([a-zA-Z]{3}, )([1-3]?[0-9] )([a-zA-Z]{3} )([0-9]{4} )(.*?)',3),
count(distinct(id)) as total_mail  from enron_to
group by regexp_extract(times,'([a-zA-Z]{3}, )([1-3]?[0-9] )([a-zA-Z]{3} )([0-9]{4} )(.*?)',4),
regexp_extract(times,'([a-zA-Z]{3}, )([1-3]?[0-9] )([a-zA-Z]{3} )([0-9]{4} )(.*?)',3);
```



Month and Year. Color shows details about Month. Size shows sum of Total Number of Communications. The marks are labeled by Month and Year. The view is filtered on Month, which excludes Null.

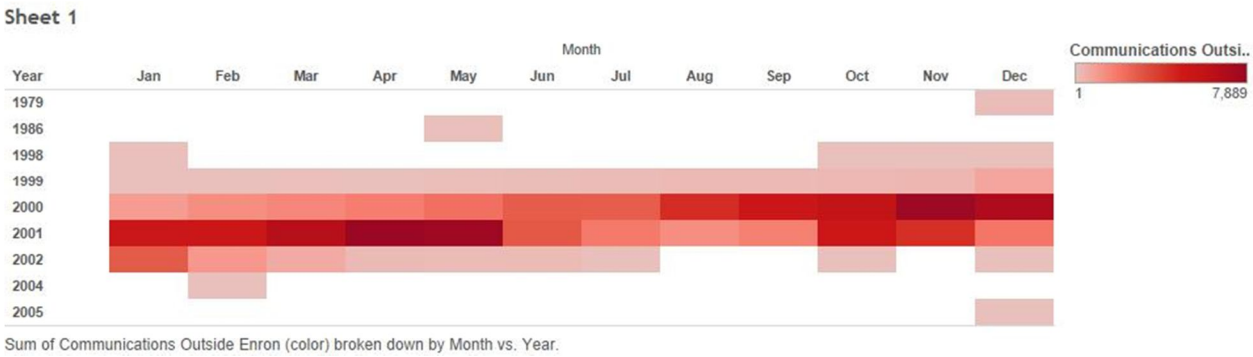
The color of the bubble represents the month while the year are listed as text. The size of the bubble is the number of emails exchanged in that month. From the image we infer that there is high activity is April of 2001.

Hypothesis 2

The number of communications specifically outside enron will increase as the date of scandal approaches

```
Create table outside_mail as
Select id,times,enron_to from enron_to where enron_to not like  '%enron%';

insert overwrite directory 's3n://cis6930dic/enronq7/'
select regexp_extract(times,'([a-zA-Z]{3}, )([1-3]?[0-9] )([a-zA-Z]{3} )([0-9]{4} )(.*?)',4),
regexp_extract(times,'([a-zA-Z]{3}, )([1-3]?[0-9] )([a-zA-Z]{3} )([0-9]{4} )(.*?)',3),
count(distinct(id)) as total_mail from outside_email group by regexp_extract(times,'([a-zA-Z]{3},
)([1-3]?[0-9] )([a-zA-Z]{3} )([0-9]{4} )(.*?)',4), regexp_extract(times,'([a-zA-Z]{3}, )([1-3]?[0-9]
)([a-zA-Z]{3} )([0-9]{4} )(.*?)',3);
```



The heat graph shows most emails were sent out of email in April 2001. From the above two hypothesis the scandal occurred in 2001.

Hypothesis 3

Degree Centrality: The person will send most number of emails and will receive most mails addressed to them will be an important person in the enron Company.

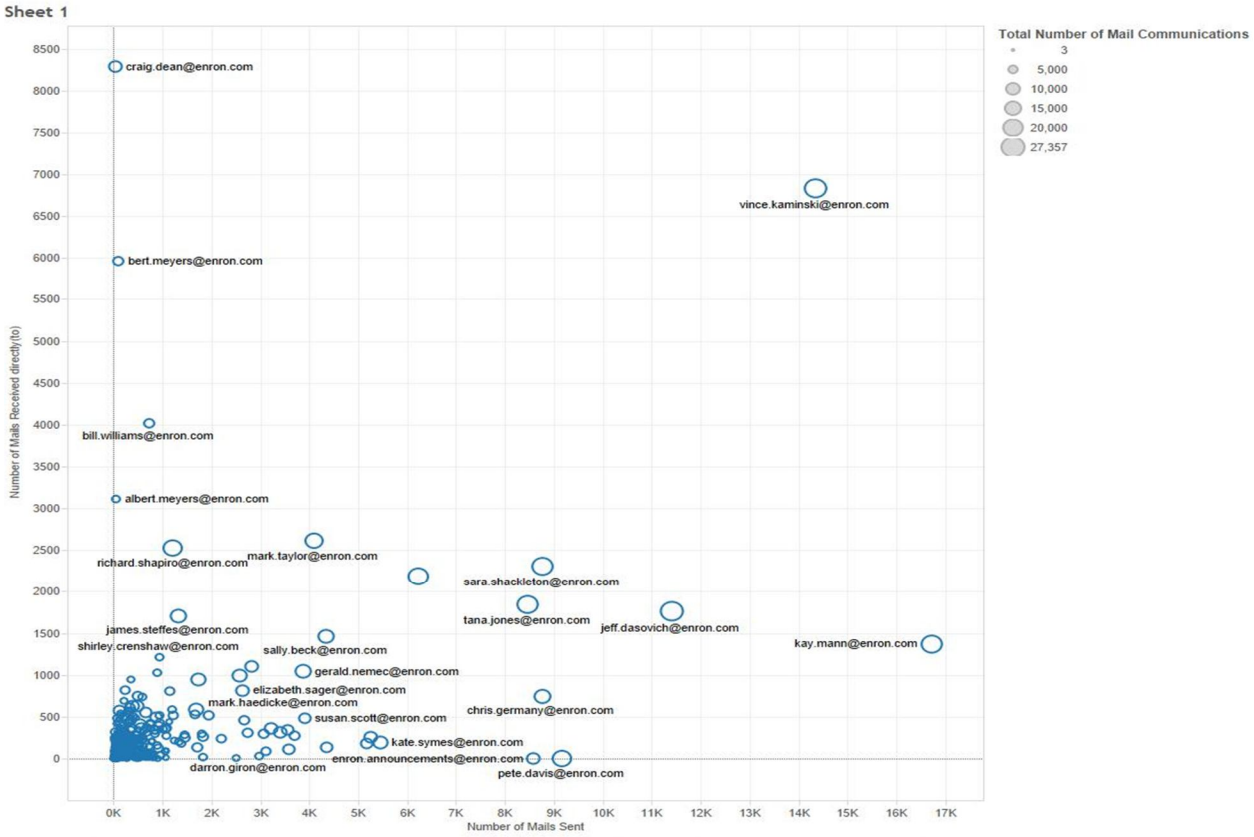
```
insert overwrite table from_track
select fromheader, count(*) as Num_sent
from enron
group by fromheader
order by  Num_sent desc ;

insert overwrite table to_track
select enron_to, count(*) as Num_cc
from enron_to
group by enron_to
order by  Num_cc desc ;
```

```
insert overwrite table cc_track
select enron_cc, count(*) as Num_cc
from enron_cc
group by enron_cc
order by  Num_cc desc ;

insert overwrite directory 's3n://hiveql/enron'
select  from_track.from_id,  from_track.num_sent,
cc_track.num_recv,
to_track.num_recv  from  cc_track join
from_track  on ( from_track.from_id=cc_track.cc)
join to_track on (to_track.to=cc_track.cc);
```

In the visualization ideally an important person will be in the upper right corner.



Sum of Number of Mails Sent vs. sum of Number of Mails Received directly(to). Size shows sum of Total Number of Mail Communications. The marks are labeled by Email Id.

In a graph of number of emails sent versus received people in the upper right quadrant are important. Two such important people are Vince Kriminski and Sara Shackeleton. Vince Kriminski was the Managing Director of research who flagged the financial Malpractices of the company and Sara Shackeleton was the vice president of the company.

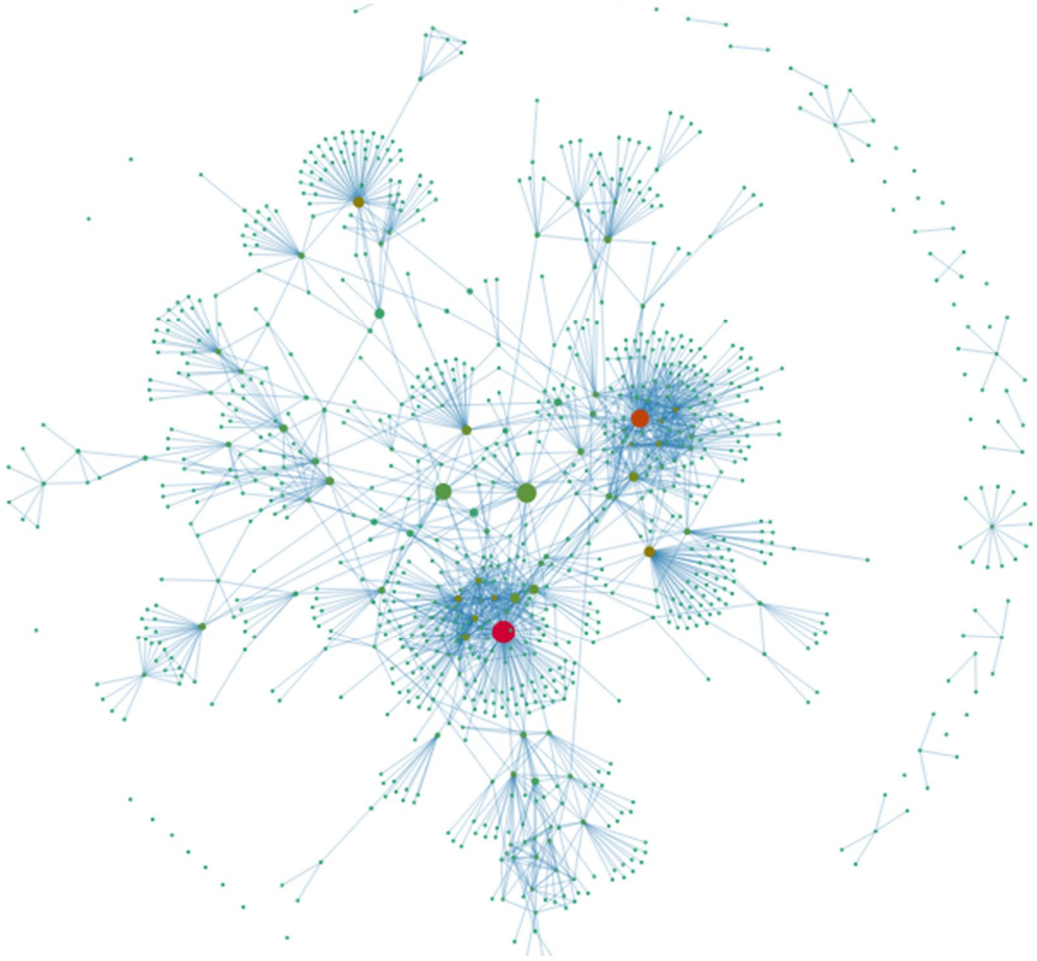
Hypothesis 4

In 2001, if we visualize a graph of every communication between employees, an important person will be centrally connected.

```
create Table enr as
select id,fromheader, times, enron_to
From enron LATERAL VIEW explode(toheader) etab as enron_to;

create table comm as
select trim(rtrim(fromheader)) as initiator, trim(rtrim(enron_to)) as dest, count(id) as count from
enr
group by  trim(rtrim(fromheader)),trim(rtrim(enron_to)) ;
```

```
insert overwrite directory 's3n://sethuids/graphFinal'
select et.initiator, et.dest, e.count+et. count as count from comm et  join comm e on et.initiator=e.dest
and et.dest=e.initiator;
```



The graph was generated by BioVerto.org(Developed by Karthik Chivukula, Dr. Dobra,). The edge represents communication between two people. Color signifies degree and size is betweeness (representing the closeness to other nodes).

Thus Jeff Dasovich represents an important personality in 2001, indecently he was the Government Reprehensive of Enron.