CONSTRAINT DETECTION AND OUTLIER EXPLANATION

Terrible Trio

Deepti Chavan(deeptisu)| Shruti Parab(shrutide) |Sushmita Sinha(ssinha7)

TABLE OF CONTENTS

Problem Definition **2**

Approach **2**

I. Potentially Correlated Candidate Columns **2**

A. Numeric Attributes **2**

Correlation coefficient 2

Chi-squared test 2

K-means clustering 2

B. Categorical **2**

II. Regression Techniques **2**

III. Data Cube **2**

Our Solution **2**

I. Potentially Correlated Candidate Columns **2**

A. Numeric **2**

Correlation coefficient 3

Chi-squared test 3

K-means clustering 3

B. Categorical **3**

Consistent Low Standard deviation method 3

II. Regression Techniques to find Patterns in the Dataset **4**

Linear Regression **4**

Robust Regression **5**

RANSAC 5

Observations **5**

III. Optimization- Data Cube **5**

1. Brute Force 5
2. Combining Multiple Aggregates 6
3. Building a data cube and querying over the cube 6

Building the data cube efficiently: **7**

Challenges faced **7**

PROBLEM DEFINITION

The high level aim is to find the correlations in the dataset and to optimize the process of finding the same. Using these correlations as constraints, find possible reasons justifying or identifying the presence of an outlier.
We intent to determine the type in which value attributes are associated with dimension attributes. The patterns that we are looking for are:

1. Increasing 2. Decreasing 3. Constant

APPROACH

Following are the different methods that we are working on to solve the problem of finding correlations.

I. POTENTIALLY CORRELATED CANDIDATE COLUMNS

A. NUMERIC ATTRIBUTES

CORRELATION COEFFICIENT

We intend to start with a brute force approach that will try to find the correlation coefficient between all the numeric attributes. The columns that have values greater than the threshold are considered for analysis.

CHI-SQUARED TEST

Chi- squared test gives us the measure of statistical independence of the columns. Finding the candidate columns that depend on each other forms the base of the problem and hence it is crucial to verify the columns given by the correlation coefficient approach, done by the Chi- squared test.

K-MEANS CLUSTERING

Clustering the data based on the mean gives us k-clusters in which every point belongs to a cluster with nearest mean which is the representative for that cluster. We aim to find the correlation of the different attributed based on the representative mean of every cluster.

B. CATEGORICAL

It is important to identify which columns have categorical data to make the analysis efficient. Potentially correlated categorical columns can be given by using the chi-squared test and columns having consistent close standard deviation values for all the categories for the value columns.

II. REGRESSION TECHNIQUES

Experimenting with different regression techniques over numerical attributes to find an increasing, decreasing, or a uniform pattern in the given dataset.

III. DATA CUBE

An optimization to improve the overall process of calculating aggregates.

OUR SOLUTION

Following is our proposed solution.

I. POTENTIALLY CORRELATED CANDIDATE COLUMNS

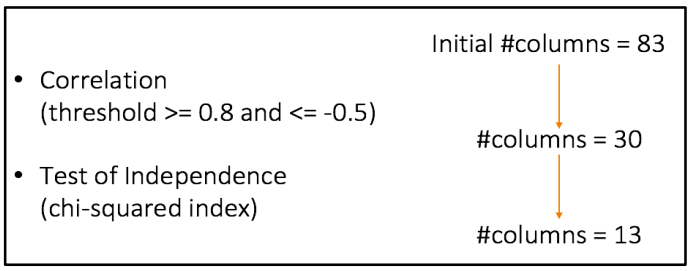
A. NUMERIC

CORRELATION COEFFICIENT

Correlation coefficient between all the numeric attributes is found in a brute force way. The columns that have values greater than the threshold (0.8) are considered as positively correlated and columns that have values lower than the threshold (-0.5) are considered as negatively correlated.

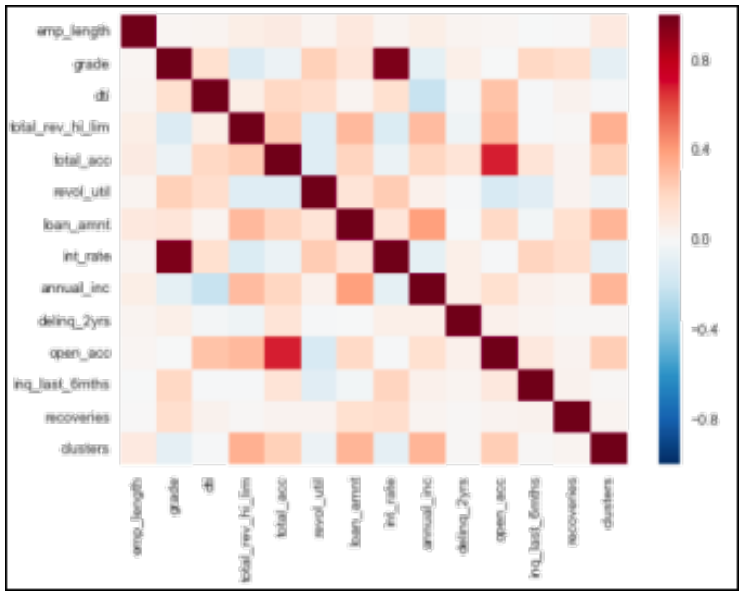
CHI-SQUARED TEST

Verification of the columns given by the correlation coefficient approach is done using the chi- squared test.



K-MEANS CLUSTERING

Using the columns that have been given by the chi-squared test, k-means clustering is done. We aim to find the correlation of the different attributes based on the representative mean of every cluster and identifying the relationship based on the heat map-based projection of the result.

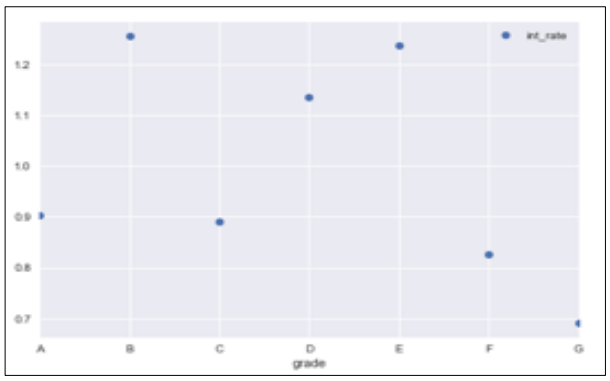


B. CATEGORICAL

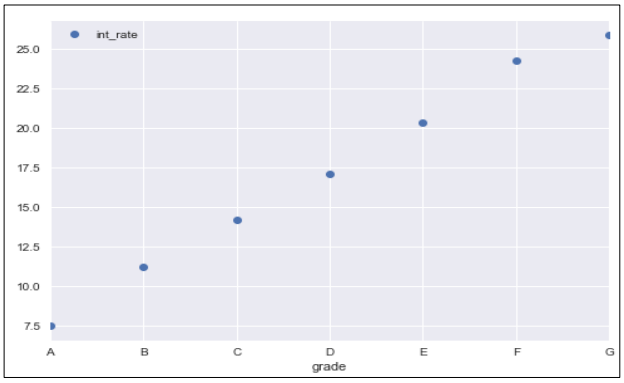
Chi-squared test is used to give the measure of independence of the categorical data and mean of the value column grouped by the category.

CONSISTENT LOW STANDARD DEVIATION METHOD

If the standard deviation of the mean around every category value remains consistent and close, points to the columns being closely related. A 5% deviation can be permissible to analyze the columns. Following graph shows the relationship between categorical column grade of the customer and int_rate.



Standard Deviation



Mean

II. REGRESSION TECHNIQUES TO FIND PATTERNS IN THE DATASET

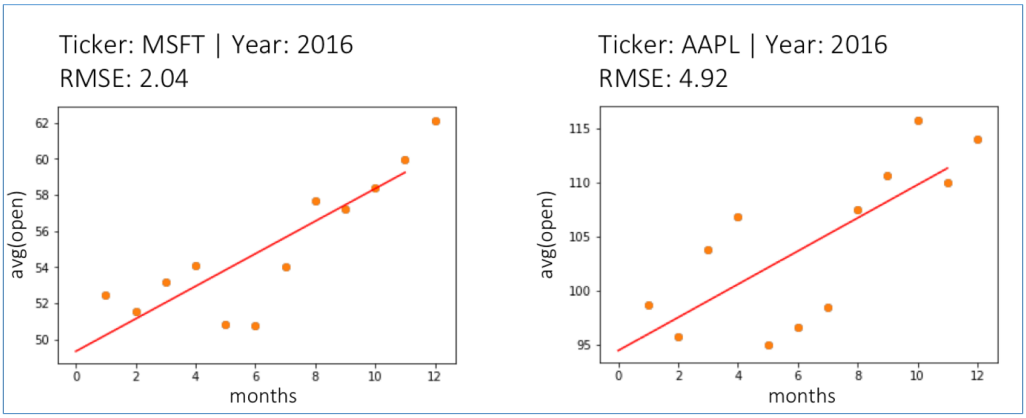
To find the patterns amongst the attributes, we have decided to apply regression techniques. Depending on the slope of the straight line that the regression model tries to fit, we infer whether an attribute follows an increasing trend, a decreasing one or is uniform.
Following are the three regression techniques that we are working on:

1. Linear Regression 2. Robust Regression 3. RANSAC Regression

Before the regression method, we had implemented the pattern finding approach with the Postgres's inbuilt regr_slope(Y, X). This function gives us the slope of the least-squares-fit linear equation determined by the (X, Y) pairs. But this is just a prediction. It does have any error value or an estimate associated with it. Thus, we decided to work with regression analysis.

LINEAR REGRESSION

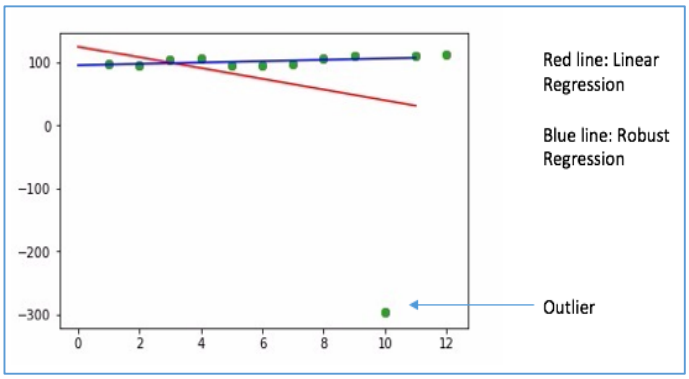
To find the relationship between the dependent variable (value attributes) *y* and an independent variable (dimension attributes) *x*, first we apply linear regression.
Here, y: aggregate of the value columns of the dataset. *E.g. avg(open_price), avg(high_price)*
And, x: dimension columns. *E.g. months, years*



Above two graphs are for the query with ticker value as AAPL and MSFT:
select avg(open) as avgOpen, EXTRACT(MONTH FROM StockData5.date) as month
from StockData5 where ticker = 'AAPL' and EXTRACT(YEAR FROM StockData5.date) = '2016'
group by month order by month
The slope of the regression line is positive in both the cases. And the Room Mean Square Error (RMSE) value will help us decide the pattern over the attribute with an accuracy.

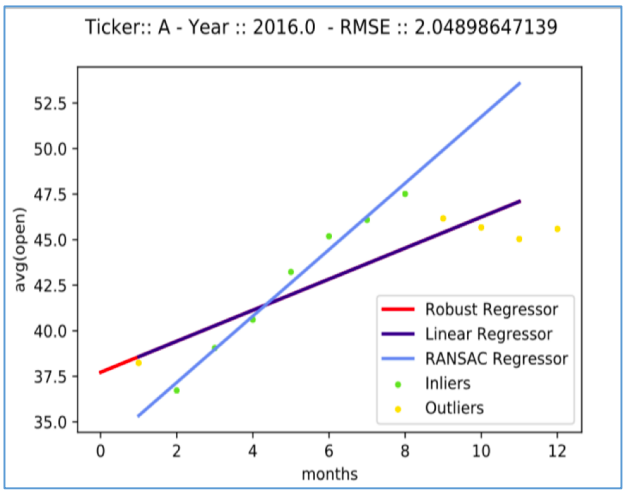
ROBUST REGRESSION

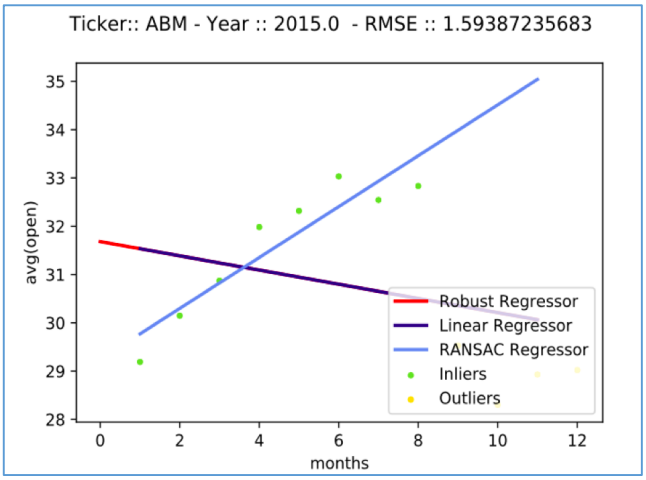
When fitting a least squares regression, we might find some outliers or high leverage data points. These are not necessarily an error in the entry. Hence, we do not want to exclude them. At the same time, we do not want the regression line to be affected by them. So, as an alternative to linear regression, we are also implementing Robust regression to find the patterns.



RANSAC

Addition to Robust regression, we are also experimenting with Random sample consensus (RANSAC). It helps to estimate parameters of a model by random sampling of observed data. Given a dataset whose data elements contain both inliers and outliers, RANSAC uses the voting scheme to find the optimal fitting result.





OBSERVATIONS

We have implemented these regression models using Python libraries, for a Stock Dataset that has around 1.5 million rows. We could see that all three types give us results near to each other. But there are few cases where the slopes of the lines are conflicting.
One way to interpret these results are:

- If all three regression lines have a positive slope, then the attribute follows an increasing pattern. If the slopes are negative, then the pattern is a decreasing one.
- In case of contradicting slope results, we could say that there isn't any pattern for the given attribute.

III. OPTIMIZATION- DATA CUBE

The process of finding patterns on the dataset involves calculating a lot of aggregates. This process can be optimized by pre-computing the aggregates and avoid overlapping aggregate operations on the data. Just to have an idea of what will be the effect of building the data cube, a time analysis was done to compare the cost for different approaches. Aggregate queries with all the possible group by combinations of dimension columns were fired on the dataset of over 1 million rows. There were three approaches:

1. BRUTE FORCE

An aggregate query for each column:
foreach (column) : SELECT agg(column) FROM table

2. COMBINING MULTIPLE AGGREGATES

For a given combination, aggregate for all the value columns is calculated in a single query.
SELECT agg(column1), agg(column2), ...FROM table

3. BUILDING A DATA CUBE AND QUERYING OVER THE CUBE

A naive approach for building the data cube. The following query builds all the possible combinations of the aggregates and unions them together. This cube table is then used as the base table for further analysis:

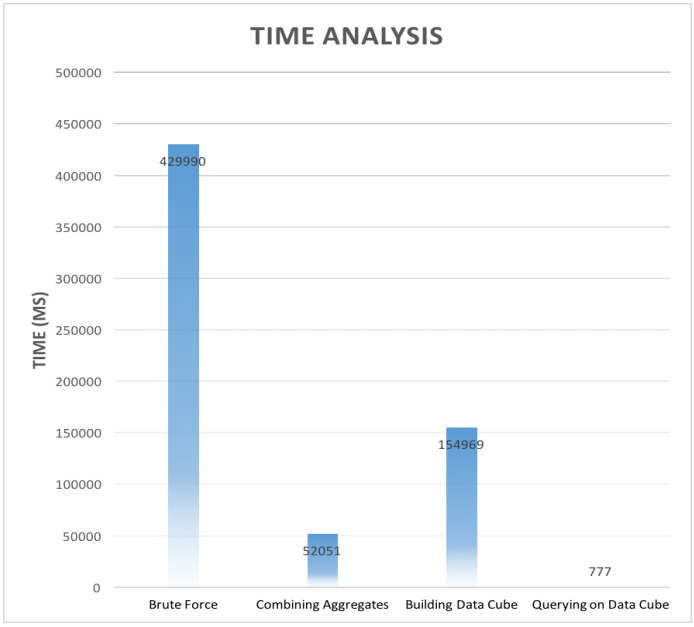
```
SELECT ticker, 11111 as year, 11111 as month, avg(stock.volume) as avg  FROM stock GROUP BY ticker
UNION
SELECT ticker, EXTRACT(YEAR FROM stock.date) as year, 11111 as month, avg(stock.volume) as avg FROM stock
GROUP BY ticker, year
UNION
SELECT ticker, EXTRACT(YEAR FROM stock.date) as year, EXTRACT(MONTH FROM stock.date) as month,
avg(stock.volume) as avg FROM stock GROUP BY ticker, year, month ORDER BY ticker,year,month
```

The figure gives a snapshot of the cube table which is built using the above query:

ticker	year	month	avg
FB	2015	1	27302883.050000000000
FB	2015	2	25007821.526315789474
FB	2015	3	26152272.590909090909
FB	2015	4	25815451.333333333333
FB	2015	5	21093535.450000000000
FB	2015	6	24452582.727272727273
FB	2015	7	35944606.090909090909
FB	2015	8	33798598.047619047619
FB	2015	9	30195472.190476190476
FB	2015	10	25999618.181818181818
FB	2015	11	27359991.100000000000
FB	2015	12	20021809.272727272727
FB	2015	11111	26955699.638888888889
FB	11111	11111	26955699.638888888889

The following graph shows the time analysis of all the three approaches:

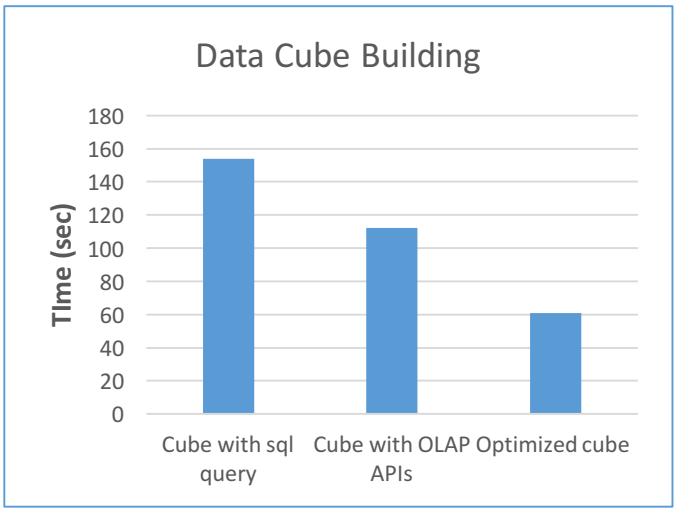
1. Time required to query all possible combinations with and without data cube.
2. The time required to build the data cube is quite high due to multiple selects and union operators. However, a good observation here is that once the data cube is built, the query time on this cube is significantly reduced.
3. Thus this analysis proved the promising effect of having a pre-computed data cube.



BUILDING THE DATA CUBE EFFICIENTLY:

The next step is finding out an efficient way to build a data cube.

- Cubes framework in Python models the data into an n-dimensional datacube. A datacube is basically a model which stores the aggregates while building the model itself. However, all the possible combinations are not pre-computed.
- For finding the patterns, we need to explore all the possible combinations. Hence, we use some slice operations on the datacube to compute the aggregates.
- One possible approach is to find the aggregate of a combination which is most granular from the datacube. Once this is done, we can use the lowest level combination to calculate the higher level aggregate combinations. This will prevent overlapping operations on the cube.
- The following time analysis shows that this approach would give a better performance in building the data cube. All the three approaches build all possible combinations of aggregates from a dataset of over 1.5 million rows.



CHALLENGES FACED

- Understanding the dimensionality columns, value columns and categorical columns
- Finding correlations for categorical data
- Finding the elbow point in the k-means clustering approach to get the most appropriate correlation values
- Understanding the concept of cubes in N-dimensional data
- Working with different OLAP frameworks
- Setting up Apache Lens OLAP framework
- Switched to Cubes framework in Python