ANOMALY DETECTION AND ATTRIBUTION USING AUTO FORECAST AND DIRECTED GRAPHS

Vivek Sankar and Somendra Tripathi

Latentview Analytics, Chennai, India

ABSTRACT

In the business world, decision makers rely heavily on data to back their decisions. With the quantum of data increasing rapidly, traditional methods used to generate insights from reports and dashboards will soon become intractable. This creates a need for efficient systems which can substitute human intelligence and reduce time latency in decision making. This paper describes an approach to process time series data with multiple dimensions such as geographies, verticals, products, efficiently, and to detect anomalies in the data and further, to explain potential reasons for the occurrence of the anomalies. The algorithm implements auto selection of forecast models to make reliable forecasts and detect such anomalies. Depth First Search (DFS) is applied to analyse each of these anomalies and find its root causes. The algorithm filters the redundant causes and reports the insights to the stakeholders. Apart from being a hair-trigger KPI tracking mechanism, this algorithm can also be customized for problems lke A/B testing, campaign tracking and product evaluations.

KEYWORDS

DFS, A/B Testing, Reporting, Forecasting, Anomaly Spotting.

1. INTRODUCTION

With the growing use of Internet and Mobile Apps, the world is seeing a steep increase in the availability and accessibility to different kinds of data – demographical, transactional, social media and so on. Modern businesses are keen to leverage on these data to arrive at smarter and timely decisions. But the existing data setup in a lot of organizations primarily provides post mortem reports of performance. The shift to a more pro-active or an instantaneous reactive approach to data based decision making requires investments in data infrastructure, skilled resources and enabling quicker and efficient dissemination of information to the right stakeholders. With more and more firms investing on their infrastructure to capture necessary data, there is a growing need for automated systems that can step in to process the raw data and provide actionable readouts to the required stakeholders.

This paper proposes one such completely automated frequentist framework which when provided with any casted data (a dataset which is a cross product of dimensions involved and their corresponding metric values for each time frame) can run in the background to provide actionable readouts. It helps business decision makers stay updated with the development in their portfolio

by providing timely updates by specifying a list of anomalies (spikes & dips) with respect to the KPIs of their interest and further providing reasons for the same.

The proposed method makes use of forecasting techniques to arrive at ballpark figures for every segment which acts as a logical substitute to user's sense. The estimates are pitted against the actual performance immediately after the availability of actual data to effectively spot anomalies. The framework consists of intermediate trigger systems that can alert corresponding stakeholders without much delay.

The system further digs down to analyze sub segments to attribute the reasons for every spotted anomaly. This approach which requires least human intervention is an effective aid in scenarios where:

- the number of segments involved could not be handled manually
- there is a lack of statistical expertise on the user front
- there is a high time lag in decision making due to existing reporting structures

This approach finds its application in a wide range of industries such as retail, e-commerce, airlines, insurance, manufacturing , logistic and supply chain, etc. to benefit portfolio managers, analyst, marketers, product and sales managers to name a few.

The task of anomaly detection and reporting starts with the processing of melted data coming from the database to cast data by producing all possible interactions between the various dimensions in the dataset. Auto-forecasting iterates through the entire cast data converts it into time series and generates prediction to be consumed in the later module. Further the framework establishes networks to understand the interdependencies between the various segments in the data. Depth First Search is then applied to spot anomalies, which are then checked for redundancy and reported.

The paper is structured as follows. The next section presents the methodology and tools used in the framework. This section has been broken into three sub-sections to highlight the pivotal modules running the entire framework. Section 3 discusses its implementation. Concluding remarks and future works are mentioned in Section 4.

2. METHODOLOGY

The proposed framework is a novel technique to spot anomalies in data with the minimum human intervention.

The three prime components that are required for its functioning are:

1. The actual value of the KPI
2. A ballpark value for the KPI
3. A scientific flagging approach

With the availability of these three components the method could be applied across a wide range of real life scenarios and across multiple verticals and KPIs. The following sections would explain the various steps involved in the framework for creating the above mentioned components and utilizing them to provide actionable insights to business users.

The entire framework is broadly broken down into the following sections:

- Data Processing Module (DPM)
- Auto Forecasting Module (AFM)
- Pattern Analysis and Reporting Module (PARM)

The flow chart in figure 1 is a concise representation of the internal structure of the modules involved and the flow of data between the modules.

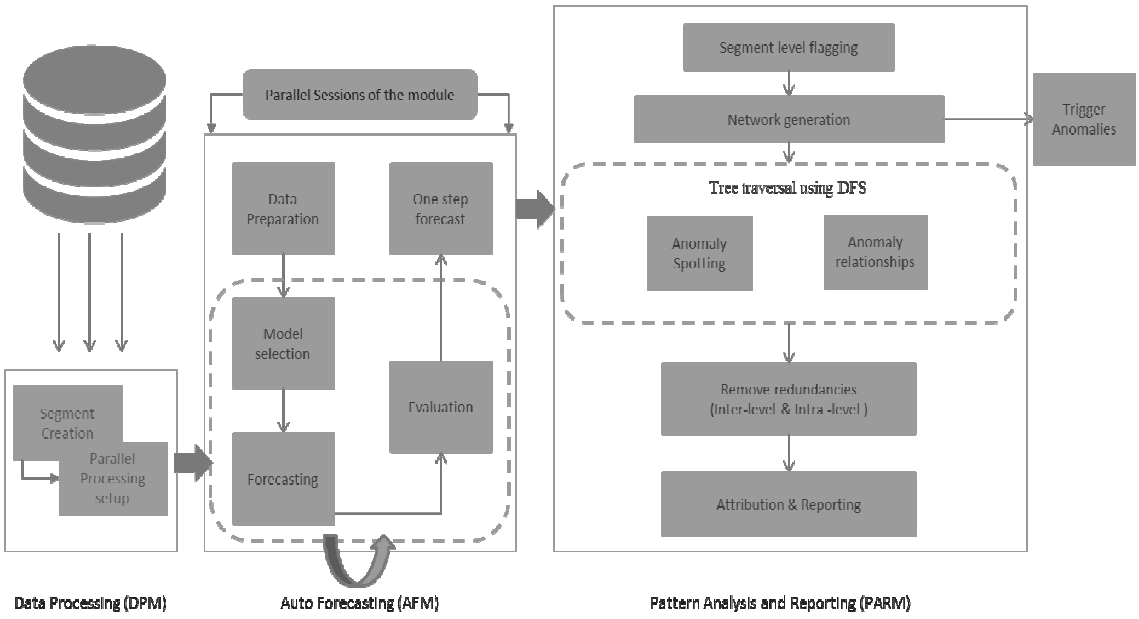

Figure 1: Framework - Block Diagram

2.1 Module 1: Data Processing Module

2.1.1 Data Input:

This module accepts raw data which could be any form of casted data. The dataset would include all the necessary dimensions that would best describe the KPI along with the KPI itself broken down at the least granularity of the time unit used for reporting. Some of the easily relatable datasets are Sales data of retail stores, web traffic of an e-commerce website, call volume data from call centers, risk decline volumes for a payment gateway and so on. Any data that encompasses trends and seasonal patterns could be a perfect fit for this module or the framework as a whole. Table 1 is an illustration of a sample data source.

Table 1: Sample Casted Dataset

Customer Segment	Product Category	Region	Date	Sales
Consumer	Furniture	Central	1/1/2010	690.77
Consumer	Furniture	Central	2/1/2010	5303.17
Consumer	Office Supplies	East	1/1/2010	1112.11
Consumer	Office Supplies	East	2/1/2010	84.01
Home Office	Furniture	West	11/1/2013	10696.84
Home Office	Furniture	West	12/1/2013	4383.98

Here the first three columns are the dimensions and date column is the indicator of frequency of reporting. The date column here is at a month level but in general the framework can be applied for Daily, Weekly, Monthly, Quarterly or Yearly reports. The final column (Sales) is the actual KPI that the business wants to track using this methodology.

2.1.2 Segments Creation:

Each of the dimensions in the dataset could hold 2 or more values and each of these could be of interest to different stakeholders. Referring back to the above sample dataset the Region dimension holds values of the geographies where there was Sales reported and each of the individual Regional Heads would want to keep an eye on the Sales of their region. Hence each of the values in every dimension potentially is a segment. The system further goes to generate segments by combining values of two dimensions. For example combining region and product category, segments like (Central _ Furniture) and (West _ Furniture) could be generated. The segment formation extends from treating every segment individually to combining all the available n dimensions. After the system generates all the possible combinations using the available dimensions in the data, each unique combination is given a Segment ID and the initial casted data is melted.

Table 2: Conversion of casted to melted data

Segment ID	Date	Sales
Seg 1	1/1/2010	690.77
Seg 1	2/1/2010	5303.17
Seg 2	1/1/2010	1112.11
Seg 2	2/1/2010	84.01
Seg 3	11/1/2013	10696.84
Seg 3	12/1/2013	4383.98

In short if there are n dimensions (D_1,D_2,D_3,….D_n) and the number of values in each dimension is (X_1,X_2,X_3,….X_n) respectively the total number of combinations generated would be (X_1+1) * (X_2+1) * …….. *(X_n+1).

2.1.2.1 Business Preferences/Inputs:

This is an optional step in the overall structure where the intention is to bucket values in each dimension to club smaller segments. Assuming there are 100 different products in the Product Category and of these 100 products 9 products lead to 95% of the overall Sales, then the rest of the 91 products could be clubbed as ‘Other Products’.

This removes certain segments from the picture based on business preferences. It could even be inputs in the form of flat files containing segments which businesses are not too concerned about. Both these measures result in the reduction of the number of segments and thereby improving the operating efficiency and memory consumption of this proposed approach.

The final melted output from this module is fed into the Auto Forecast Module after the removal of all data points corresponding to the latest unit of time and passing through the Multiprocessing stage to enable parallel processing in the Auto Forecast Module.

2.1.3 Parallel Processing Module:

The forecasting module can be duplicated as multiple processes as each of the segment is treated independent of the other. The dataset is broken down into multiple parts of equal number of segments and fed to the multiple sessions of the AFM.

2.2 Module 2: Auto Forecast Module

The auto forecast module is designed to generate forecasts on multiple time series data which gets fed to the PARM discussed in the next section.

The following steps explain how one step forecasts are generated:

2.2.1 Data Preparation:

Aggregated data is converted to time series data of the specified frequency. Each time series is then iteratively checked for missing values and outliers. The framework provides the flexibility of using linear and cubic spline interpolation for treating missing values and outliers [1], the time series is decomposed using STL and the trend component is smoothened. This helps in approximating missing values and minimizing the effect of outliers. Each time series is checked for 0-padding (both leading and trailing). Powerful transformations such as Box-Cox can be applied before moving on to the next phase.

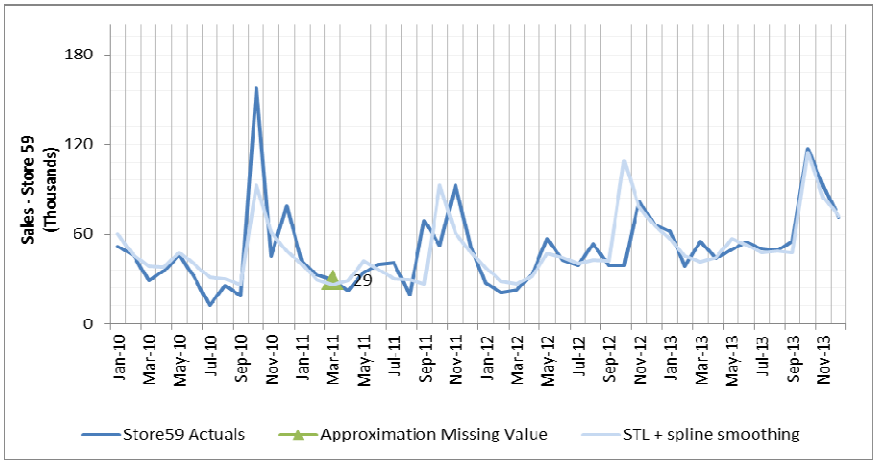

Figure 1: Missing value treatment using STL + spline smoothing

2.2.2 Model selection:

This phase involves choosing the right set of models for a given time series. The current pool includes Exponential Smoothing – ETS implementation for automatic selection and auto.Arima[2], Nnet - Feed forward neural networks with a single hidden layer, TSLM for fitting linear models to time series including trend and seasonality, TBATS (Exponential smoothing state space model with Box-Cox transformation, ARMA errors, Trend and Seasonal Components)[3,4] and STL.

It is important that the seasonality component is correctly identified before forecasting. Over-parameterization or force fitting seasonality when there is no real seasonal component might produce unreliable forecasts. Also, in cases where there are too few data points only ARIMA and Regression models are used. Croston's method is used in cases where time series have intermittent data. TBATS is used with data with weekly and annual seasonality. All relevant models are picked for forecasting in this phase.

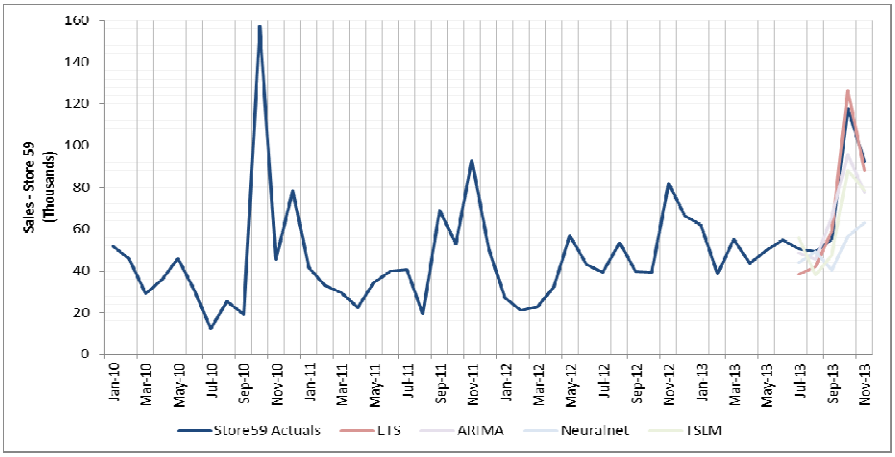

Figure 2: Model selection based on Holdout accuracy

2.2.3 Forecasting:

The level of the time series must be specified before this step starts. A small portion of the data is held out to be used as validation [5]. This helps in preventing over fitting and gives a truer estimation of the generalization error. The module then runs all the models picked for the given time series and provides the residuals, fit statistics and confidence intervals.

2.2.4 Evaluation:

The metric to be used for evaluating the fit can be specified at the start of execution of this module. The framework provides MAPE (Mean absolute percentage error), MSE (Mean squared error), MAE (Mean absolute error) and MASE as possible measures of forecast accuracy. A summary of all fits statistics from selected models applied on the data is generated and the winner model is chosen based on the model getting the best accuracy. Poor predictions are flagged off at this step for manual intervention.

Table 1: Selection of winner model based on Performance metric

Store #	ETS	TBATS	Arima	Nnet	Tslm	Croston's*
Store 53	7.2%	4.3%	4.7%	**2.0%**	3.3%	NA
Store 54	72.2%	44.8%	41.4%	104.3%	**15.9%**	NA
Store 55	2.6%	5.4%	**2.6%**	5.4%	6.6%	NA
Store 56	5.6%	2.5%	2.8%	12.6%	5.9%	NA
Store 57	4.5%	**1.8%**	4.4%	4.1%	2.3%	NA
Store 58	3.9%	**2.9%**	5.4%	4.8%	**2.5%**	NA
Store 59	**11.5%**	NA	14.9%	31.5%	20.8%	NA
Store 60	NA	NA	10.1%	NA	**8.9%**	NA
Store 61	1.8%	**1.3%**	3.5%	33.2%	1.8%	NA
Store 62	33.8%	52.1%	71.3%	27.0%	**14.0%**	NA
Store 63	12.7%	4.3%	7.8%	**4.0%**	8.7%	NA
Store 64	4.1%	**1.4%**	3.0%	8.0%	2.5%	NA

2.2.5 One step forecast:

The final forecast are generated by running the winner model on the entire data, as missing recent data points could cause loss of valuable information. The residuals, confidence interval and one step forecasts for this model are then passed to the next module.

2.3 Module 3: Pattern Analysis and Reporting Module (PARM)

PARM receives data feed separately from the Data Processing Module and Auto Forecast Module.

- DPM provides the KPI values of the latest unit of time for every segment. These values were the ones which were kept aside from the melted dataset before being fed into AFM.

- AFM provides reliable forecast values along with the residuals and prediction intervals based on the required confidence interval for every segment.

2.3.1 Segment Level Flagging

Every available segment in the data has an actual value, a ballpark value (forecasted output) and a prediction interval based on the point forecast.

2.3.1.1 Prediction Interval

As per Hyndman [7], a prediction interval is an interval associated with a random variable yet to be observed, with a specified probability of the random variable lying within the interval. For example, I might give an 80% interval for the forecast of GDP in 2014. The actual GDP in 2014 should lie within the interval with probability 0.8. Prediction intervals can arise in Bayesian or frequentist statistics.

A confidence interval is an interval associated with a parameter and is a frequentist concept. The parameter is assumed to be non-random but unknown, and the confidence interval is computed

from data. Because the data are random, the interval is random. A 95% confidence interval will contain the true parameter with probability 0.95. That is, with a large number of repeated samples, 95% of the intervals would contain the true parameter.

The range of the prediction interval is dependent on two factors:

i. The accuracy of the forecast. Higher the accuracy narrow is the band and a poor accuracy pushes the limit to $-\infty$ and $+\infty$.

ii. The desired confidence percentage. Higher the required confidence broader is the band.

2.3.1.2 Reasons for independent forecast

Every segment generated by the DPM could be unique in its properties and thereby could exhibit its own trend and seasonal attributes. Also obtaining forecast values from sub granular level would result in aggregating errors of the sub granular levels and hence affect the accuracy of the overall forecast.

2.3.1.2.1 Anomalies – Dips and spikes

The actual value for the KPI corresponding to the latest unit of time is pitted against the prediction interval for that segment.

If , $A_T > EUL_T$ => Flag for spike,

$A_T < ELL_T$ => Flag for dip

where,

A_T - actual KPI value for the time period T

EUL_T & ELL_T - upper and lower limits of prediction intervals for time period T

With every segment flagged independently for anomalies instantaneous triggers could be sent out to accountable stake holders alerting them to react without much time latency. In such scenarios businesses start to drill down the KPIs using dimensions based on their judgments to arrive at reasons for the anomaly.

2.3.2 Network Generation

The network here is a scientific substitute to replace the operations of extensive drill downs by the business consumers. The approach to node formation and node connections are explained in the next section.

2.3.3 Node Formation

Every segment created by DPM would be a node by itself. The number of dimensions involved in creation of the node indicates the level of the node i.e. if dimensions are taken one at a time the level is 2 while a combination of two dimensions indicates level 3 and extends up to (n+1) levels.

2.3.4 Node Connections

Every node at level k would be connected from one or more nodes from level k-1 where any one or more values involved in the formation of node at level k has a commonality of value with any node at level (k-1). For example, any node at level 3 formed by [region-central and product category-furniture] would be connected from nodes [region-central] and [product category-furniture] at level 2. This network structures is enabled using Directed Acyclic Graphs (DAG).

2.3.4.1 Directed Acyclic Graphs (DAG)

In mathematics and computer science, a directed acyclic graph is a directed graph with no directed cycles. That is, it is formed by a collection of vertices and directed edges, each edge connecting one vertex to another, such that there is no way to start at some vertex v and follow a sequence of edges that eventually loops back to v again.

DAGs may be used to model many different kinds of information. The reachability relation in a DAG forms a partial order, and any finite partial order may be represented by a DAG using reachability. A collection of tasks that must be ordered into a sequence, subject to constraints that certain tasks must be performed earlier than others, may be represented as a DAG with a vertex for each task and an edge for each constraint; algorithms for topological ordering may be used to generate a valid sequence. Additionally, DAGs may be used as a space-efficient representation of a collection of sequences with overlapping subsequences. DAGs are also used to represent systems of events or potential events and the causal relationships between them. DAGs may also be used to model processes in which data flows in a consistent direction through a network of processors, or states of a repository in a version-control system [6].

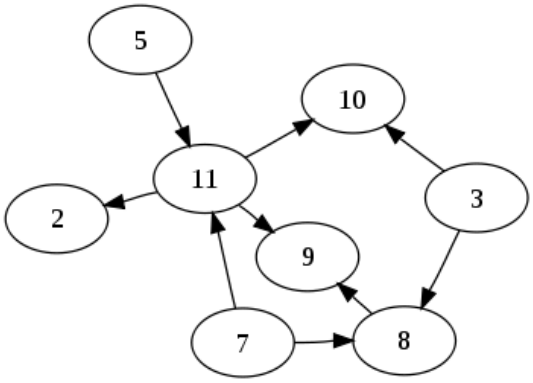

Figure 3: Directed Acyclic Graph (DAG)

2.3.5 Node Information

Every node is identified by the segment id and includes the respective flags for anomalies obtained for the segments at the previous step.

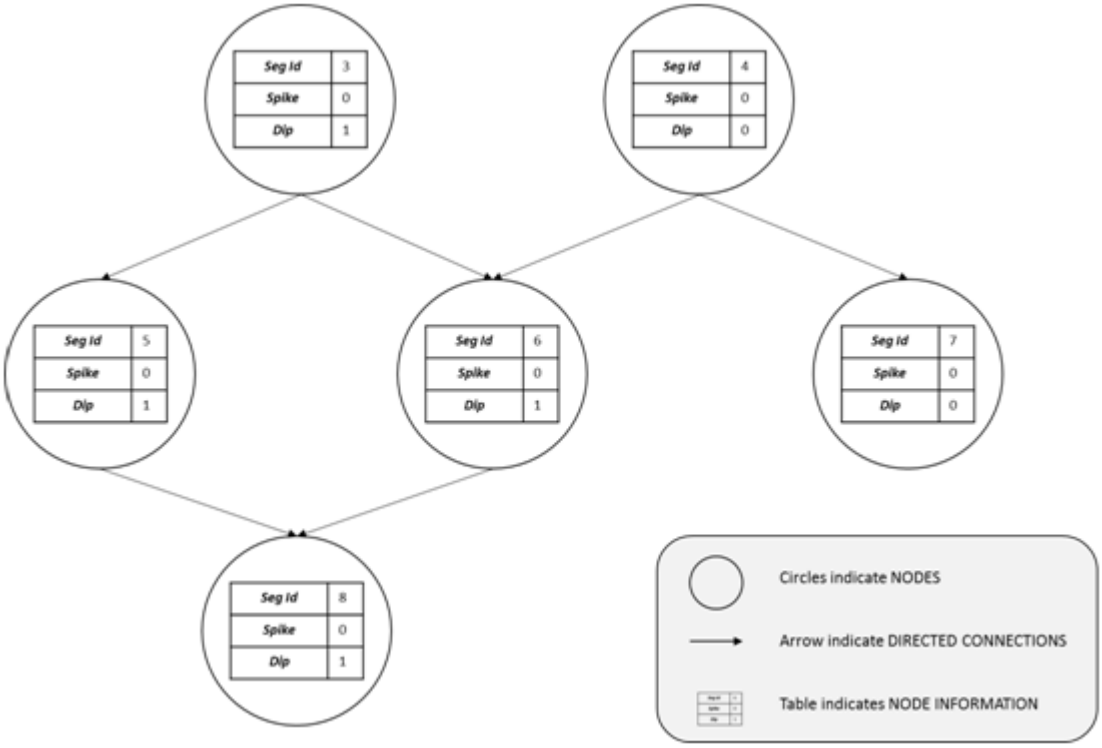

Figure 4: Network formation - An illustration

2.3.6 DAG as a substitute for Drill Down

The DAG network is used a substitute for manual drill down process as every possible dimensions is fitted above and below the other dimensions and combination of dimensions thereby enabling an exhaustive drill down setup. But in order to understand the cause for every anomaly, it is necessary to understand the relation between anomalies spotted and set up the right order of fitting the dimensions. This is enabled by traversing the network generated from every spotted anomaly which is explained in detail in section 2.3.7.

2.3.7 Network traversal for spotting dependent anomalies

Network is primarily used to understand the relationship between the nodes flagged for anomalies. A regular Depth First Search (DFS) traversal technique is used to traverse through the network.

2.3.7.1 Steps for primary consolidation

1. The traversal starts from the top most level (level 1)
2. Every node at level 1 is visited to see if it is flagged for an anomaly (spike or dip)
    - If it is flagged, the segment is marked as a Main Segment and is fixed as one of the starting points of DFS

- If the node is not flagged, the focus moves to the next node in the same level

3. Step 2 is repeated until level n and all the main segments are marked. The Main Segment nodes marked are of two categories – one flagged for spikes and the other for dips
4. From the list of nodes marked as main segment, the algorithm picks the first node. With this node as the starting point, the network is traversed till the last level using DFS. If a visited node on the traversal route is flagged for a similar anomaly, that node is marked as impacted node corresponding to the main segment.
5. Step 4 is repeated for all the Main Segments marked in step 3

The generated list of Main Segment and Impacted Segment combination concludes the primary consolidation thereby providing the relationship between one segment anomaly and its directly related segment anomalies.

2.3.7.2 Root Causing through redundancy removal

The output from the above level has several redundant factors which are wiped out in this step to provide a read out of a mutually exclusive anomalies for the business users' perusal. The redundancies present are of two types

a. Inter Level Redundancies

b. Intra Level Redundancies

2.3.7.2.1 Inter Level Redundancies

Since the Main Segment list had all the available flagged nodes an anomaly spotted at a top level could have its effect permeating down until the bottom most level. These sub level Main Segments are part of the list of Impacted Segments for the top level Main Segment. The framework identifies that having all these in the final read out would be reiterating the same anomaly repeatedly in multiple forms.

2.3.7.2.2 Intra Level Redundancies

Intra Level Redundancies are more due to inherent data aspects. Two or more nodes at the same level could be flagged for a similar anomaly. But more often than not it is the effect of one on the other as each of the nodes have an inherent volume of the other nodes. This is a more complex case to eliminate from the primary consolidation unlike the Inter Level Redundancies.

2.3.7.3 Steps to remove redundancies

1. The Main Segments at level n are picked.
2. Compare the list of Impacted Segments for each pair of Main Segments in this level.
    - If there is an intersection then there lives a common thread
    - If there are no intersections, move to the next pair of Main Segments

3. If there is an intersection in the previous step create a proxy index separately for each of the Main Segment in the pair. The proxy index is simply a product of (absolute drop in KPI from forecast) and (Percentage drop of KPI from forecast). This index tangibly does not have any specific meaning or unit of measurement but higher the value of the index greater is the probability of it being a root cause. This index is used to tie break between the paired segments at each level.
    - The winner of the tiebreaker remains in the list of primary consolidation.
    - The looser is removed from the primary consolidated list.
4. This process is repeated for every pair at each level and across all the levels moving upward.
5. Now, with the truncated list the process starts from the top most level. A Main Segment at the top most level is selected and its Impacted Segments are chosen.
6. The truncated list is looped to check if the Impacted Segments are present as a Main Segment.
    - If it is present the sub level Main Segment corresponding to the matching Impacted Segment is removed.
    - If none of the Impacted Segments in the list match with the Main Segment in the sub levels skip to the next Main Segment in the current level of focus.

At the end of this iteration what remains are mutually exclusive anomalies (Main Segments) and their corresponding sub level variations (Impacted Segments).

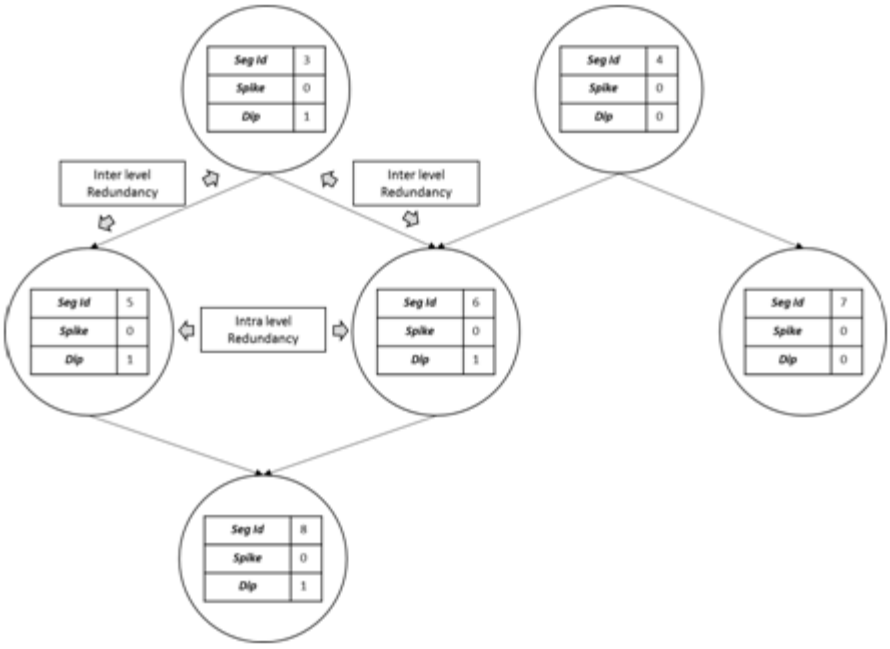

Figure 5: Types of Redundancies

2.3.8 Attribution and Reporting

The final step of the framework is to communicate the anomalies in a readable format to the respective stakeholders. The change in Main Segment in absolute terms is used as the base to calculate the proportion of change at the corresponding sub level segments tagged to the Main Segment as its Impacted Segments. The sub level segments are ordered based on decreasing value of the calculated proportion. The final consolidated list after the sorting is then automatically converted into a presentation where every spotted anomaly is shown separately along with its causes / impact.

2.3.9 Business Preferences and Reporting Customizations

The framework also provides the flexibility to integrate static business filters or preferences to include and exclude certain kinds of anomalies from the final consolidated list. Also the final reporting could be customized according to the needs of the business users by setting up the reporting module accordingly in the framework.

3. RESULTS

The framework was successfully implemented using R and python and tested out on eight different datasets across multiple KPI's. The implementation has resulted positively for businesses in terms of reduction in time latency between report availability and actual actions from the report, spotting variations in smaller segments which were initially neglected and avoiding redundant actions by intimating the right stakeholders based on assigned accountability. On an average 6 out of 20 root cause reports generated during the Beta phase gave conclusive and actionable insights on the anomalies in the data. The time required to forecast and generate reports increases exponentially with the increase in the number of dimensions. However, with the use of parallel processing time taken to generate the results for a dataset with close to 8 dimensions (200,000 columns) was reduced from 9 hours to 1 hour.

With the aid of shiny package in R and PyQt it was possible to create a user-friendly UI for taking inputs and displaying interactive dashboards.

4. CONCLUSIONS

The increasing reliance on data for decision-making has added pressure on analysts to provide quick and accurate reports. The volume of data has increased manifolds in the past decade. If such vast volumes of data can be managed and processed more efficiently it could lead to multi fold gains in all organisations. The framework discussed in this paper finds application in a wide range of domains for KPI tracking, AB testing, campaign tracking and product evaluations.

The methodology can further be improved by adding functionalities like allowing external regressors and multiplicative metrics, interface for integration to Business intelligence tools and models that can handle high frequency data. Distributed computing via big data platforms can further increase the scalability of this approach.

ACKNOWLEDGEMENTS

The authors would like to thank LatentView Analytics for providing the opportunity, inputs and resources to generate this framework. The authors would like to mention their gratitude to co-employees who provided their support in this work. The authors would extend their thanks to Prof. Subhash Subramanian and Prof. Mandeep Sandhu for their guidance and inputs in writing down the paper and proof reading it.

REFERENCES

[1] R implementation by B. D. Ripley and Martin Maechler (spar/lambda, etc). https://www.r-project.org/Licenses/GPL-2

[2] Hyndman, R.J., Akram, Md., and Archibald, B. (2008) "The admissible parameter space for exponential smoothing models". Annals of Statistical Mathematics, 60(2), 407--426.Models: A Roughness Penalty Approach. Chapman and Hall.

[3] Hyndman, R.J., Koehler, A.B., Snyder, R.D., and Grose, S. (2002) "A state space framework for automatic forecasting using exponential smoothing methods", International J. Forecasting, 18(3), 439--454

[4] Hyndman, R.J. and Khandakar, Y. (2008) "Automatic time series forecasting: The forecast package for R",Journal of Statistical Software, 26(3)

[5] Leonard, Michael. "Large-Scale Automatic Forecasting Using Inputs and Calendar Events." White Paper (2005): 1-27.

[6] Thulasiraman, K.; Swamy, M. N. S. (1992), "5.7 Acyclic Directed Graphs", Graphs: Theory and Algorithms, John Wiley and Son, p. 118, ISBN 978-0-471-51356-8.

[7] http://robjhyndman.com/hyndsight/intervals/

AUTHORS

Vivek Sankar is a post graduate in Business Administration from Sri Sathya Sai Institute of Higher Learning and is currently working in LatentView Analytics, Chennai, since September 2012.

Somendra Tripathi received his B.Tech degree in computer science from Vellore Institute of Technology in 2013. He is currently working in LatentView Analytics, Chennai.