Service Assurance in Telecommunication

A Data Science Approach

Sajana Jayathissa
Team Scanvengers
University of Moratuwa
209339V

Hasitha Jayasundara
Team Scanvengers
University of Moratuwa
209400C

***Abstract*— Analyze the fault reporting and clearing patterns of telco service management to increase the service levels.**

Keywords—ML, PCA, XGBoost, TAT

I. Introduction

Internet and Broadband services have transformed businesses and markets and generated significant wealth and economic growth in many countries. They have also empowered individuals and communities with new ways of doing things, as well as transformed our ways of learning and sharing knowledge.

This revolution has been created as a major demand for all telecommunication providers worldwide. Due to that most of these service providers have been investing a lot in bringing the cutting-edge technologies to customers and at the same time, they have put a huge effort on their service levels. Therefore we are trying to analyze the fault reporting and clearing pattern of well-known TSP to get insights about the process and provide suggestions to improve the process.

The Dataset is a combination of two tables which namely the Reported Faults and Cleared faults and the contents of those tables are well described by its name. A more detailed breakdown of these two tables is as below.

A. Data

This data set consists of two tables which can be named as Reported faults and cleared faults. Both these tables have below characteristics

	Reported Faults	*Cleared Faults*
Content	About reported faults	About cleared faults
# of columns	11	6
# of rows	752,058	748,010
Duration	10/1/2019 to 12/31/2019	10/1/2019 to 12/31/2019
Features	PROM_NUMBER, CIRT_DISPLAYNAME, CIRT_SERT_ABBREVIATION, PROM_REPORTED, PROM_CAUSE, PROM_REGN_CODE, MSAN, PROM_WORG_NAME, OLD_FAULT_NO, OLD_PROB_CLEARED_DATE, OLD_PROB_CLEARED_WG, ALTERNATE_NUMBER	PROM_NUMBER, PROM_CLEARED, PROM_PCAT_NAME, PROM_PRSC_NAME, PROM_WORG_NAME, FEEDBACK

Feature Descriptions: Table 01 - Reported Faults

PROM_NUMBER – A unique ID, which is the reference issued for each fault logged by customer. Customers are inquired about the status of the complaint using this.

CIRT_DISPLAYNAME– Connection username to identify the connection of the customer which is faulty

CIRT_SERT_ABBREVIATION- combined attribute describing connection type (Broad Band, TV, Voice) and the connection medium (Copper, Fiber, Wireless)

PROM_REPORTED – Fault Reported Time stamp

PROM_CAUSE – Identified fault type by the contact center officer at the time of fault reporting. (Selected from list of values)

PROM_REGN_CODE- Code used for identifying the relevant geographical area of the country

MSAN- Name of the serving node for the customer in the access network

PROM_WORG_NAME – Name of the team which the fault has initially assigned by the contact center team

OLD_FAULT_NO – Reference no of the most recent fault reported before the current fault

OLD_FAULT_CLEARED_DATE – Cleared Date of the most recent fault before the current fault

OLD_PROB_CLEARED_WG – Name of the team which cleared the most recent fault before the current fault

Feature Descriptions: Table 02 - Cleared Faults

PROM_NUMBER– Reference of the Fault

PROM_CLEARED – Fault Cleared Time stamp

PROM_PCAT_NAME – Identified category of the fault when troubleshooting. selected form list of values (General Category)

PROM_PRSC_NAME – Identified specific issue category of the fault when troubleshooting. selected form list of values according to the CAT_1(More Detailed Version of CAT_1)

PROM_WORG_NAME – Name of the team which fault has been cleared by

FEEDBACK – Feedback of the customer about the fault clearing process which has been collected after the fault has been cleared

Feature Descriptions: Table 03 - Area Mapping

PROM_REGN_CODE- Code used for identifying the relevant geographical area of the country

DISTRICT- Administrative Districts of Sri Lanka

PROVINCE-Administrative Provinces of Sri Lanka

Feature Descriptions: Table 04 - Service Types

CIRT_SERT_ABBREVIATION- combined attribute describing connection type (Broad Band, TV, Voice) and the connection medium (Copper, Fiber, Wireless)

SERVICE_TYPE- General service type of the customer (ADSL, FTTH, IPTV, PSTN, LTE)

B. *Data Preparation*

Frist part of any Data science problem is to clean the data set and reshape it according to the requirements. We have used python with pandas for this purpose.

The shape of each table, this is the most important thing which has to done initially to get an idea about the volume of the dataset

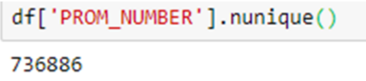
	Number of Rows	*Number of Columns*
Reported faults	752,058	11
Cleared faults	748,010	6

Here we can see that there is a difference between these tables, this is due to no of the faults reported during these 3 months has been attended on the next months or faults reported other than these 3 months has been attended with these 3 months.

Since maintaining 2 tables is hard and records are not identical in these two tables are joined using inner join, then this will result in a table where those outside entries are eliminated.

	Number of Rows	*Number of Columns*
Faults	748,105	11

It was noticed that the resultant table has recorded more than the table which has lesser records (cleared faults table has only 748010 records but the final table has 748105 records). This happened because PROM_NUMBER column has duplicate records, though we consider it as unique.



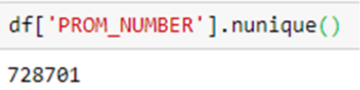
```
df['PROM_NUMBER'].nunique()

736886
```

Therefore, need to remove the duplicate values of PROM_NUMBER column to overcome this issue before merging, after removing the duplicate values using PROM_NUMBER column below table shows the result

	Number of Rows	*Number of Columns*
Reported faults	732,774	11
Cleared faults	747,802	6
Faults	728,701	16

It clearly shows that it has eliminated the above issue after removing duplicates in PROM_NUMBER column, now the PROM_NUMBER column of the final table has all unique values.



```
df['PROM_NUMBER'].nunique()

728701
```

PROM_REGN_CODE is another important column hence it consists of the geographical area data. These geographical areas have been mapped to administrative district and provinces and that table also loaded to the environment

	Number of Rows	*Number of Columns*
PROM_REGN_CODE No of Rows	376	728,701
PROM_REGN_CODE No of Unique Values	376	456

It shows that there are garbage values available in the PROM_REGN_CODE column of the Faults table, inner join with Area mapping table will help to overcome this

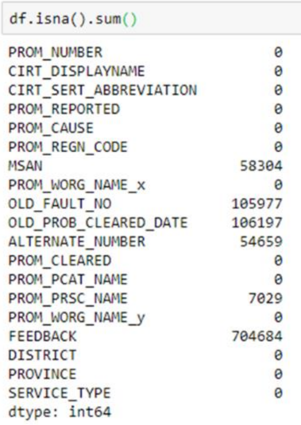
	Faults
PROM_REGN_CODE No of Rows	726,741
PROM_REGN_CODE No of Unique Values	368

After joining those two tables it shows that only 368 unique values are present in the final table though it is 376 unique values that are available in the mapping table. This is due to some codes are obsolete now and not using in the operation.

CIRT_SERT_ABBREVIATION is the next important column because it consists the information about the connection type of the customer since there are large verities of service type we will limit our analysis to the major service types and drop irrelevant service types using an inner join with the service type table which only consists the relevant service types

	Services	*Faults*
CIRT_SERT_ABBREVIATION No of Rows_before_merge	13	726,741
CIRT_SERT_ABBREVIATION No of Unique Values_before_merge	13	53
CIRT_SERT_ABBREVIATION No of Rows_after_merge		718,714
CIRT_SERT_ABBREVIATION No of Rows_after_merge		13

Finally, we can check the null values of the faults table before processing further. It shows that all major columns don’t have any null values and we can proceed with other columns that have null values because most of those columns are not considered for this analysis.



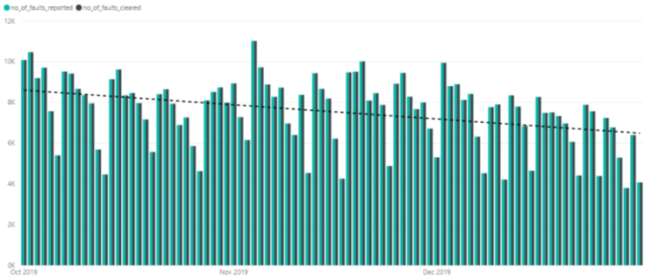
```
df.isna().sum()

PROM_NUMBER                   0
CIRT_DISPLAYNAME              0
CIRT_SERT_ABBREVIATION        0
PROM_REPORTED                 0
PROM_CAUSE                    0
PROM_REGN_CODE                0
MSAN                      58304
PROM_WORG_NAME_x              0
OLD_FAULT_NO             105977
OLD_PROB_CLEARED_DATE    106197
ALTERNATE_NUMBER          54659
PROM_CLEARED                  0
PROM_PCAT_NAME                0
PROM_PRSC_NAME             7029
PROM_WORG_NAME_y              0
FEEDBACK                 704684
DISTRICT                      0
PROVINCE                      0
SERVICE_TYPE                  0
dtype: int64
```

II. Descriptive/ Diagnostic Analytics for Telco Service Management

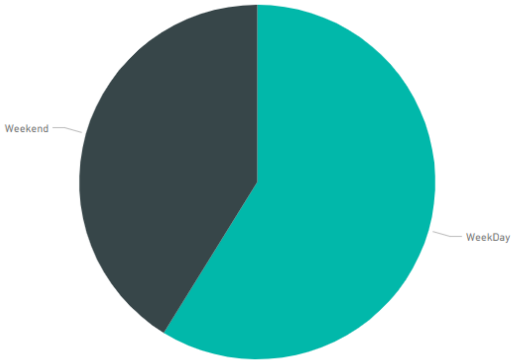
In this context, we have selected incidents for a period of October to December and performed descriptive and diagnostic analytics for the data set. As a result, we managed to observe the trend of support tickets coming in from the customer base, frequency, service types, and other demographics.

A. *Trend Analysis*

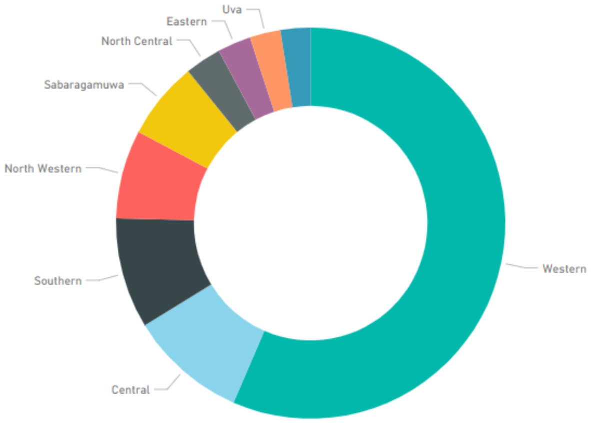


The number of support tickets is higher at the beginning of the period and a slight reduction in the month of December. This was due to the weather patterns, where October to November heavy rain was observed followed by a large number of service requests. However, in the month of December, the weather has come back to the normal status as the incidents reported.

On the weekends the number of support tickets is considerably lower.

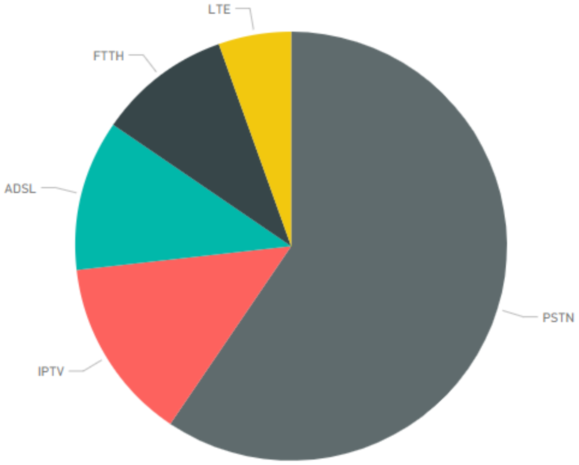


As per the below chart, a bigger chunk of the customers is based on western province. Hence two prominent reasons were diagnosed,



Most of the heavy usage customers are business users. Western province residents are prone to go out on weekends due to urbanization.

B. *Service Type*

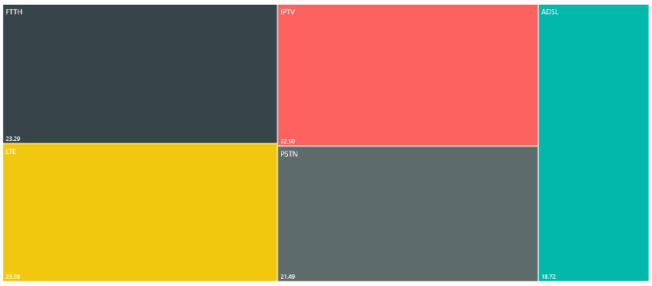


The majority of the tickets are from PSTN, which is explained by the fact that PSTN is the core service for most of the customer base.

C. *Turnaround time (TAT)*

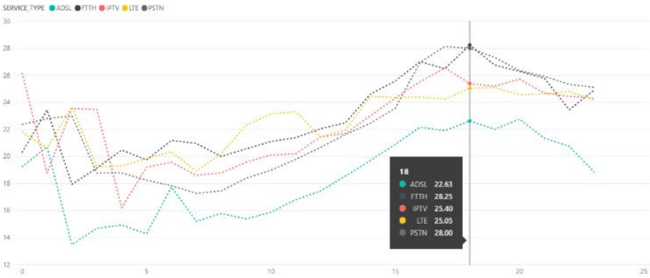
One of the most important KPI of service management is turnaround time. A timely response to customers forms a pivotal part in the overall customer-experience extended to them. Customers ask for a service, make a query or register a request with huge expectations from a company, underneath lies a hope to get the quickest of responses. The length of this TAT goes a long way to make customers stay with a company.

Average TAT by service



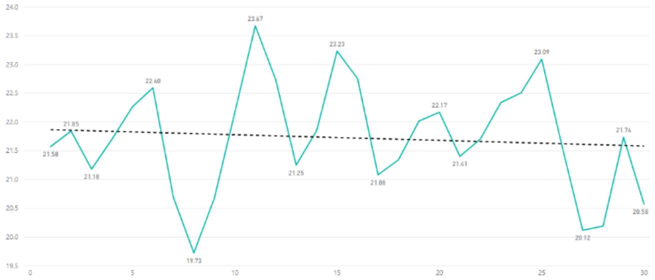
As per the analysis, the lowest turnaround time (18.72 hours) was observed in ADSL as most of the time it is sufficient to re-evaluate the configuration and change accordingly rather than a site visit for the same. Also, for FTTH, maintenance is costly and time-consuming as replacement of the entire portion of the fiber cable is needed rather than fiber joints.

Average TAT by the reported hour and service type.



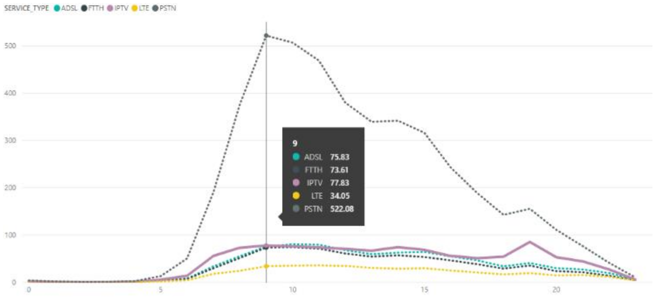
As per the graph, we can observe a peek between 5 PM to 8 PM. This is due to the volume. At this time window, most of the people are home and using most of the services.

Average TAT by the day of the month.



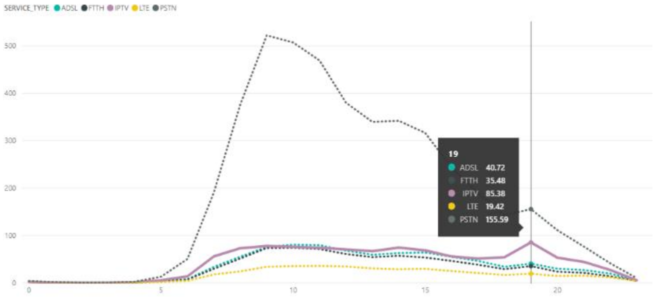
It is observed that the trend of average TAT decreasing as the month goes by. Most of the people use their allocated quota before the month-end. Because of that, a slight volume decrease would be there towards the end of the month and more capacity/ resources available to provide a better service.

Average faults reported by the hour.



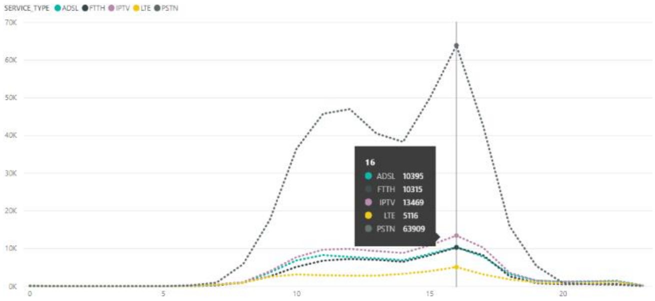
Two observations can be made from the above graph. The peak of reported tickets is around 9 AM. Considering the customer base, most of the heavy users are commercial users and office hours start from 8:30 AM to 9 AM. Hence more volume is visible in that time window.

Incidents for IPTV peak around 7 PM.



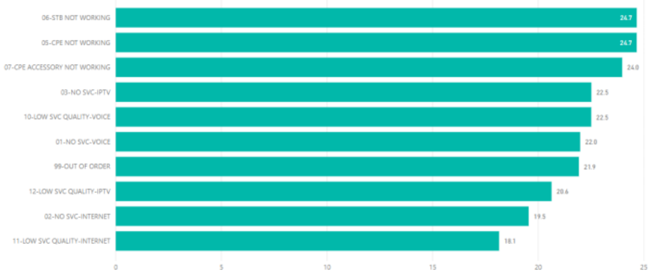
This time slot is known for entertainment among the customer base.

Number of faults cleared by the hour



In this graph, a clear observation can be pointed out where a peek of faults cleared is around 4 PM. This is mainly due to practice, where field visit engineers come to the office and close the incidents. A possible suggestion is to provide a means to close the ticket via mobile by the customer site itself.

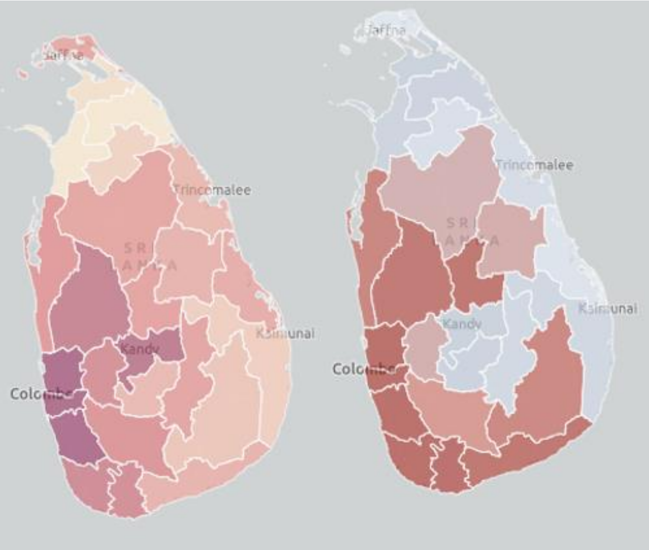
Average TAT for most common incidents.



Most of the issues reflecting in the above graph require a field visit. Therefore, a relatively high turnaround will be there for the customers.

Fault reported and TAT by district.

The first map and the second map respectively show the intensity of fault reported and TAT.



These graphs reflect 3 key observations.

- Even though there are a lot of incidents in Kandy, they have managed to maintain a high efficiency considering the low TAT.
- In Hambantota and Monaragala, even though they have lower volume comparatively, their efficiency is less as per the TAT.
- Comparatively, most of the volume is generated in Colombo and nearby districts.

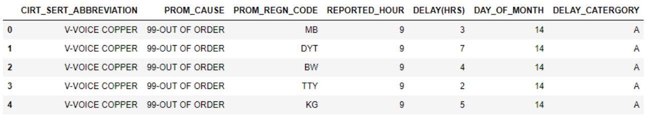
III. Predictive Analytics for Turnaround Time

As the second part of the analysis, we wanted to come up with a predictive model to forecast the TAT, based on service type, problem cause, region, reported hour and the day of the month.

As per the business requirement, TAT should be classified under the below logic.

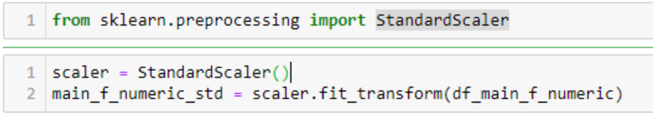
0 <= Category A <= 12, 12 < Category B <= 24, 24 < Category C

A sample data set is as below,



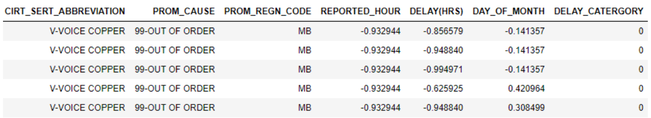
	CIRT_SERT_ABBREVIATION	PROM_CAUSE	PROM_REGN_CODE	REPORTED_HOUR	DELAY(HRS)	DAY_OF_MONTH	DELAY_CATERGORY
0	V-VOICE COPPER	99-OUT OF ORDER	MB	9	3	14	A
1	V-VOICE COPPER	99-OUT OF ORDER	DYT	9	7	14	A
2	V-VOICE COPPER	99-OUT OF ORDER	BW	9	4	14	A
3	V-VOICE COPPER	99-OUT OF ORDER	TTY	9	2	14	A
4	V-VOICE COPPER	99-OUT OF ORDER	KG	9	5	14	A

After the initial basic pre-processing of the data, we have standardized the numerical fields.



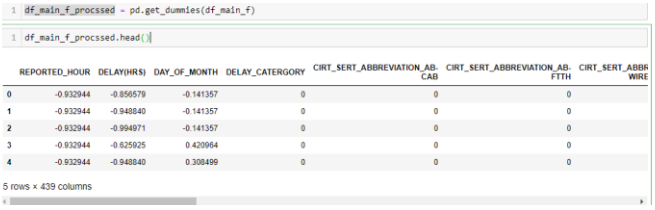
```
from sklearn.preprocessing import StandardScaler
```

```
scaler = StandardScaler()
main_f_numeric_std = scaler.fit_transform(df_main_f_numeric)
```



CIRT_SERT_ABBREVIATION	PROM_CAUSE	PROM_REGN_CODE	REPORTED_HOUR	DELAY(HRS)	DAY_OF_MONTH	DELAY_CATERGORY
V-VOICE COPPER	99-OUT OF ORDER	MB	-0.932944	-0.856579	-0.141357	0
V-VOICE COPPER	99-OUT OF ORDER	MB	-0.932944	-0.948840	-0.141357	0
V-VOICE COPPER	99-OUT OF ORDER	MB	-0.932944	-0.994971	-0.141357	0
V-VOICE COPPER	99-OUT OF ORDER	MB	-0.932944	-0.625925	0.420964	0
V-VOICE COPPER	99-OUT OF ORDER	MB	-0.932944	-0.948840	0.308499	0

Then, we used the one-hot encoding for the categorical variables.



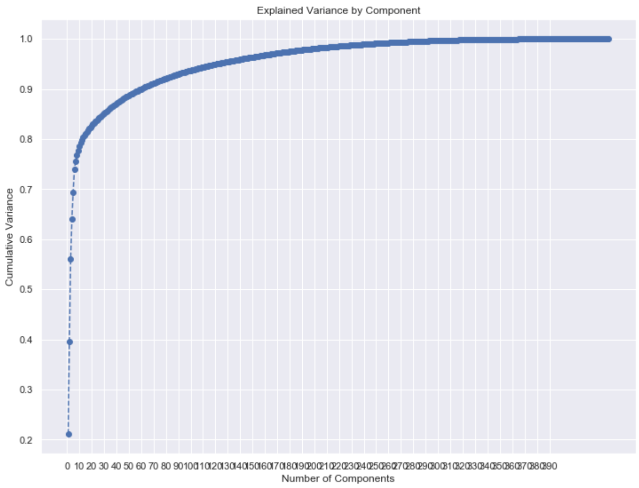
```
df_main_f_procssed = pd.get_dummies(df_main_f)
```

```
df_main_f_procssed.head()
```

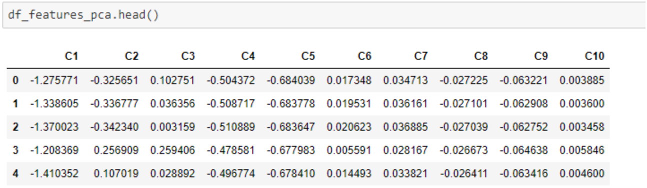
	REPORTED_HOUR	DELAY(HRS)	DAY_OF_MONTH	DELAY_CATERGORY	CIRT_SERT_ABBREVIATION_AB-CAB	CIRT_SERT_ABBREVIATION_AB-FTTH	CIRT_SERT_ABBR WIRE
0	-0.932944	-0.856579	-0.141357	0	0	0	
1	-0.932944	-0.948840	-0.141357	0	0	0	
2	-0.932944	-0.994971	-0.141357	0	0	0	
3	-0.932944	-0.625925	0.420964	0	0	0	
4	-0.932944	-0.948840	0.308499	0	0	0	

5 rows × 439 columns

After the cleaning and pre-processing we have used PCA (Principal Component Analysis) for feature engineering.

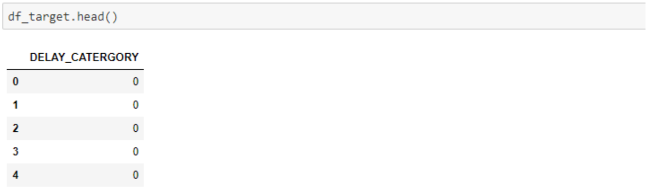


As per the graph, 80% of the data can be described by 10 components. Building on this, we have finalized our features and target before the ML model building.



```
df_features_pca.head()
```

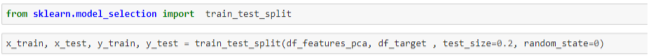
	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
0	-1.275771	-0.325651	0.102751	-0.504372	-0.684039	0.017348	0.034713	-0.027225	-0.063221	0.003885
1	-1.338605	-0.336777	0.036356	-0.508717	-0.683778	0.019531	0.036161	-0.027101	-0.062908	0.003600
2	-1.370023	-0.342340	0.003159	-0.510889	-0.683647	0.020623	0.036885	-0.027039	-0.062752	0.003458
3	-1.208369	0.256909	0.259406	-0.478581	-0.677983	0.005591	0.028167	-0.026673	-0.064638	0.005846
4	-1.410352	0.107019	0.028892	-0.496774	-0.678410	0.014493	0.033821	-0.026411	-0.063416	0.004600



```
df_target.head()
```

	DELAY_CATERGORY
0	0
1	0
2	0
3	0
4	0

For the training set, we have used 80% of the data and 20% for the testing set.



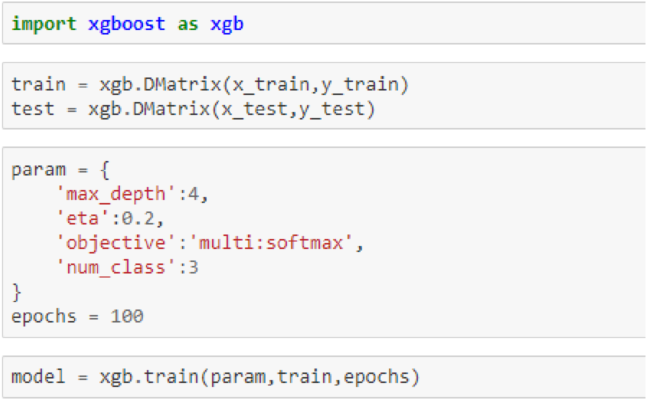
```
from sklearn.model_selection import  train_test_split
```

```
x_train, x_test, y_train, y_test = train_test_split(df_features_pca, df_target , test_size=0.2, random_state=0)
```

For the ML model, we have used XGBoost, which is an implementation of gradient boosted decision trees designed for speed and performance.

As per hyperparameters,

- Maximum depth of a tree = 4
- Learning rate = 0.2
- We need to do a multiclass classification. Hence as the learning objective, we have used ‘multi: softmax’
- The number of passes through the dataset is 10 and the number of classes is 3 as per the business requirement.



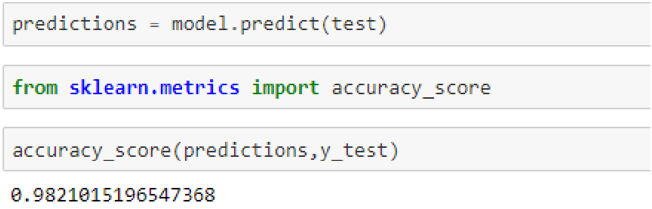
```
import xgboost as xgb
```

```
train = xgb.DMatrix(x_train,y_train)
test = xgb.DMatrix(x_test,y_test)
```

```
param = {
    'max_depth':4,
    'eta':0.2,
    'objective':'multi:softmax',
    'num_class':3
}
epochs = 100
```

```
model = xgb.train(param,train,epochs)
```

After building the model, we have tested the accuracy and the model is a success as per the results.



```
predictions = model.predict(test)
```

```
from sklearn.metrics import accuracy_score
```

```
accuracy_score(predictions,y_test)
```

```
0.9821015196547368
```

IV. Conclusion

The objective of this analysis is to improve the service levels and decrease the fault rates.

On that note, the model we built will be very practical as we would know the moment the incident was reported and how much time it will take beforehand. Then the contact center team can inform the customer of the average time taken to resolve the complaint. This will help to improve the customer satisfaction because 98% time we can keep the promise. Most customers only care about this and it is well known that if we gave a promise and satisfied it, then that's it for them and they feel really satisfied with the operator. Accordingly, we can optimize our customer service and optimize resource deployment based on the incident.

Also, focus on the regions which are low on efficiency, identify physical components and training required for our engineers would be arranged.

It is noted that PROM_REGN_CODE has a significant importance when categorizing the delay, therefore we can suggest increasing the service level by addressing worst regions where the Category C fault percentage is high.

The below table shows a number of faults cleared with each category and top 10 regions according to the percentage fault cleared in CAT_C, here we have only used the regions with total fault count greater than 100.

REGN_CODE	***CAT_A***	***CAT_B***	***CAT_C***	***Total***	***%_CAT_C***
JL	2684	1535	8046	12265	65.60%
NDP	131	72	349	552	63.22%
GVN	144	111	417	672	62.05%
EP	242	229	722	1193	60.52%
TNL	39	17	83	139	59.71%
HPG	187	144	474	805	58.88%
PK	1198	879	2756	4833	57.02%
NL	247	158	521	926	56.26%
GLW	309	212	663	1184	56.00%
AB	475	383	1084	1942	55.82%

If we address the issues of the above regions it will have significant improvement in service levels.

Acknowledgment

We would like to thank Dr. Uthayasanker Thayasivam. PhD and Dr. A. Shehan Perera for their expert advice and encouragement throughout the data science module which greatly helped us build the foundation for this analysis.

References

[1] pandas. 2020. pandas documentation. [ONLINE] Available at: https://pandas.pydata.org/docs/. [Accessed 10 April 2020]

[2] NumPy. 2019. NumPy Documentation. [ONLINE] Available at: https://numpy.org/doc/. [Accessed 10 April 2020].

[3] matplotlib. 2020. matplotlib Documentation. [ONLINE] Available at: https://matplotlib.org/3.2.1/contents.html#. [Accessed 10 April 2020]

[4] seaborn: statistical data visualization. 2020. seaborn Documentation. [ONLINE] Available at: https://seaborn.pydata.org/. [Accessed 10 April 2020].

[5] scikit-learn Machine Learning in Python. 2019. User guide. [ONLINE] Available at: https://scikit-learn.org/stable/user_guide.html. [Accessed 10 April 2020].

[6] XGBoost. 2020. XGBoost Documentation. [ONLINE] Available at: https://xgboost.readthedocs.io/en/latest/#. [Accessed 10 April 2020].