BACKGROUND AND MOTIVATION

Background: In the dynamic and competitive landscape of the telecommunications industry, minimizing customer churn is crucial for sustaining business growth and profitability. Customer churn, referring to the rate at which customers discontinue services, poses a significant challenge for telecom companies. Identifying factors leading to churn and developing predictive models can empower companies to proactively address customer needs, enhance service quality, and implement targeted retention strategies.

Motivation: The motivation behind this data science project stems from the recognition of the substantial impact that customer churn can have on a telecom company's revenue and market share. By leveraging advanced analytics and machine learning techniques, we aim to gain valuable insights into the factors influencing churn among mobile subscribers. This understanding will enable the development of predictive models that forecast potential churn events.

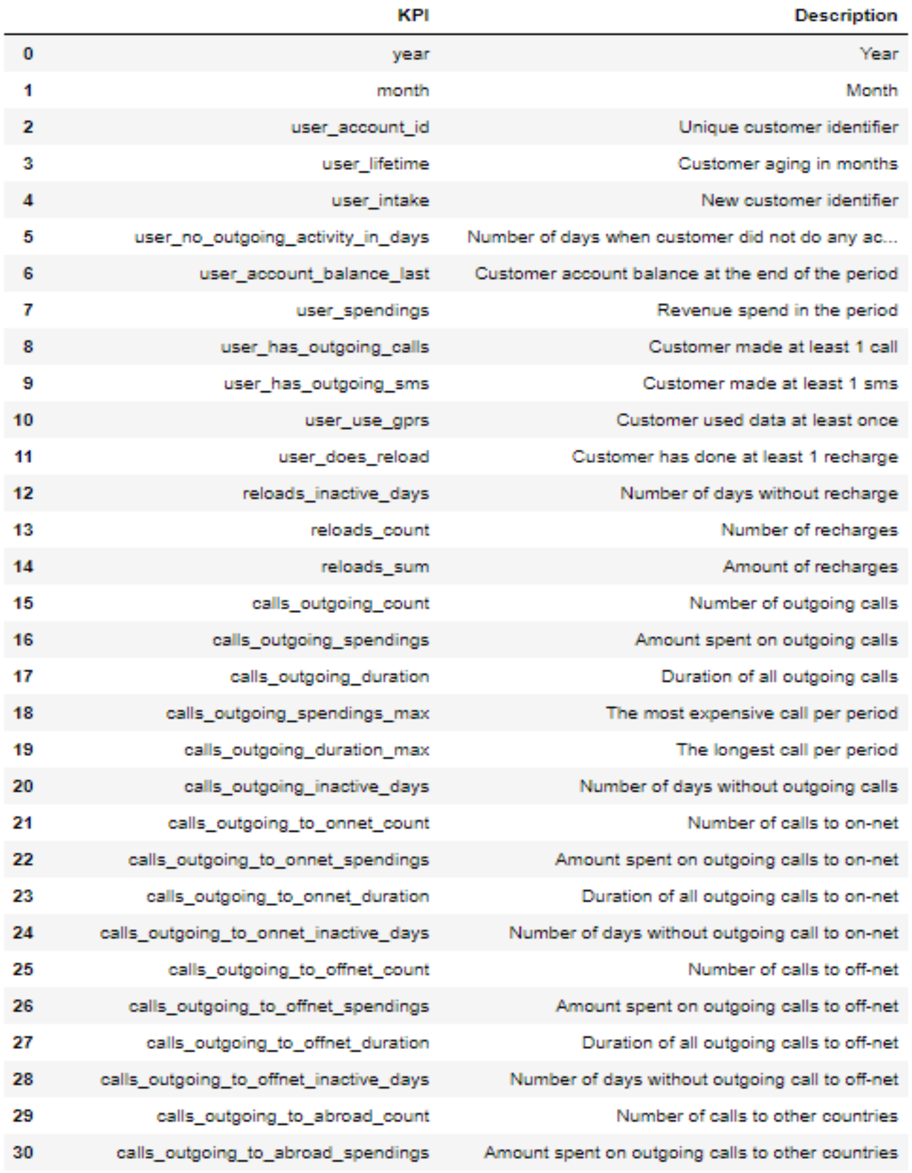
The dataset available at the provided link serves as a valuable resource, offering a comprehensive collection of features related to customer behaviour, service usage, and other relevant metrics. Exploring this dataset provides an opportunity to uncover hidden patterns, correlations, and trends that contribute to customer churn. By harnessing this information, we can develop predictive models that assist telecom companies in identifying at-risk customers early on, allowing for the implementation of targeted retention measures.

The ultimate goal of this data science project is to equip telecom companies with actionable insights derived from predictive modelling, enabling them to implement data-driven strategies to reduce customer churn. By doing so, companies can enhance customer satisfaction, improve service offerings, and maintain a competitive edge in the dynamic telecommunications market.

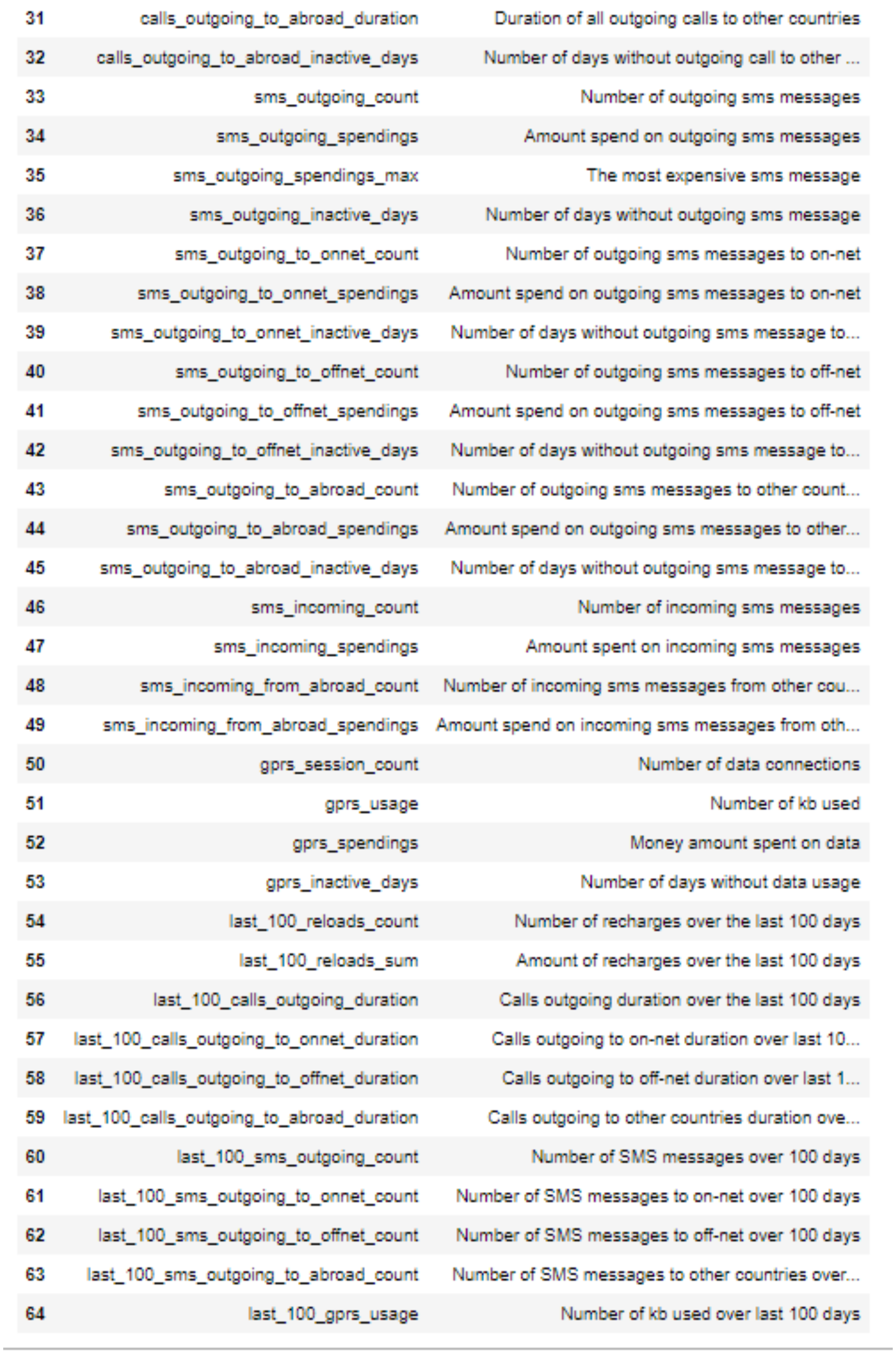
DATA DESCRIPTION

Kaggle is the source of data. The dataset contains 66469 rows and 66 columns including the target variable.

Our variables have been tabulated below:



	KPI	Description
0	year	Year
1	month	Month
2	user_account_id	Unique customer identifier
3	user_lifetime	Customer aging in months
4	user_intake	New customer identifier
5	user_no_outgoing_activity_in_days	Number of days when customer did not do any ac...
6	user_account_balance_last	Customer account balance at the end of the period
7	user_spendings	Revenue spend in the period
8	user_has_outgoing_calls	Customer made at least 1 call
9	user_has_outgoing_sms	Customer made at least 1 sms
10	user_use_gprs	Customer used data at least once
11	user_does_reload	Customer has done at least 1 recharge
12	reloads_inactive_days	Number of days without recharge
13	reloads_count	Number of recharges
14	reloads_sum	Amount of recharges
15	calls_outgoing_count	Number of outgoing calls
16	calls_outgoing_spendings	Amount spent on outgoing calls
17	calls_outgoing_duration	Duration of all outgoing calls
18	calls_outgoing_spendings_max	The most expensive call per period
19	calls_outgoing_duration_max	The longest call per period
20	calls_outgoing_inactive_days	Number of days without outgoing calls
21	calls_outgoing_to_onnet_count	Number of calls to on-net
22	calls_outgoing_to_onnet_spendings	Amount spent on outgoing calls to on-net
23	calls_outgoing_to_onnet_duration	Duration of all outgoing calls to on-net
24	calls_outgoing_to_onnet_inactive_days	Number of days without outgoing call to on-net
25	calls_outgoing_to_offnet_count	Number of calls to off-net
26	calls_outgoing_to_offnet_spendings	Amount spent on outgoing calls to off-net
27	calls_outgoing_to_offnet_duration	Duration of all outgoing calls to off-net
28	calls_outgoing_to_offnet_inactive_days	Number of days without outgoing call to off-net
29	calls_outgoing_to_abroad_count	Number of calls to other countries
30	calls_outgoing_to_abroad_spendings	Amount spent on outgoing calls to other countries



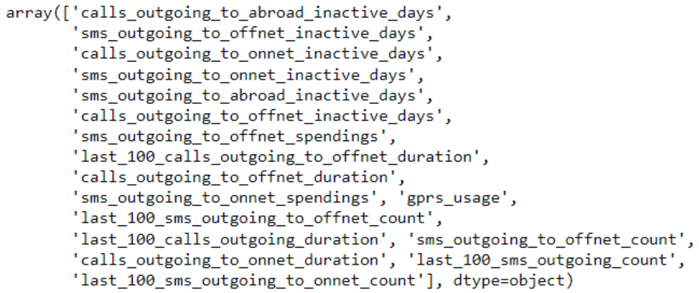
31	calls_outgoing_to_abroad_duration	Duration of all outgoing calls to other countries
32	calls_outgoing_to_abroad_inactive_days	Number of days without outgoing call to other ...
33	sms_outgoing_count	Number of outgoing sms messages
34	sms_outgoing_spendings	Amount spend on outgoing sms messages
35	sms_outgoing_spendings_max	The most expensive sms message
36	sms_outgoing_inactive_days	Number of days without outgoing sms message
37	sms_outgoing_to_onnet_count	Number of outgoing sms messages to on-net
38	sms_outgoing_to_onnet_spendings	Amount spend on outgoing sms messages to on-net
39	sms_outgoing_to_onnet_inactive_days	Number of days without outgoing sms message to...
40	sms_outgoing_to_offnet_count	Number of outgoing sms messages to off-net
41	sms_outgoing_to_offnet_spendings	Amount spend on outgoing sms messages to off-net
42	sms_outgoing_to_offnet_inactive_days	Number of days without outgoing sms message to...
43	sms_outgoing_to_abroad_count	Number of outgoing sms messages to other count...
44	sms_outgoing_to_abroad_spendings	Amount spend on outgoing sms messages to other...
45	sms_outgoing_to_abroad_inactive_days	Number of days without outgoing sms message to...
46	sms_incoming_count	Number of incoming sms messages
47	sms_incoming_spendings	Amount spent on incoming sms messages
48	sms_incoming_from_abroad_count	Number of incoming sms messages from other cou...
49	sms_incoming_from_abroad_spendings	Amount spend on incoming sms messages from oth...
50	gprs_session_count	Number of data connections
51	gprs_usage	Number of kb used
52	gprs_spendings	Money amount spent on data
53	gprs_inactive_days	Number of days without data usage
54	last_100_reloads_count	Number of recharges over the last 100 days
55	last_100_reloads_sum	Amount of recharges over the last 100 days
56	last_100_calls_outgoing_duration	Calls outgoing duration over the last 100 days
57	last_100_calls_outgoing_to_onnet_duration	Calls outgoing to on-net duration over last 10...
58	last_100_calls_outgoing_to_offnet_duration	Calls outgoing to off-net duration over last 1...
59	last_100_calls_outgoing_to_abroad_duration	Calls outgoing to other countries duration ove...
60	last_100_sms_outgoing_count	Number of SMS messages over 100 days
61	last_100_sms_outgoing_to_onnet_count	Number of SMS messages to on-net over 100 days
62	last_100_sms_outgoing_to_offnet_count	Number of SMS messages to off-net over 100 days
63	last_100_sms_outgoing_to_abroad_count	Number of SMS messages to other countries over...
64	last_100_gprs_usage	Number of kb used over last 100 days

DATA PREPROCESSING

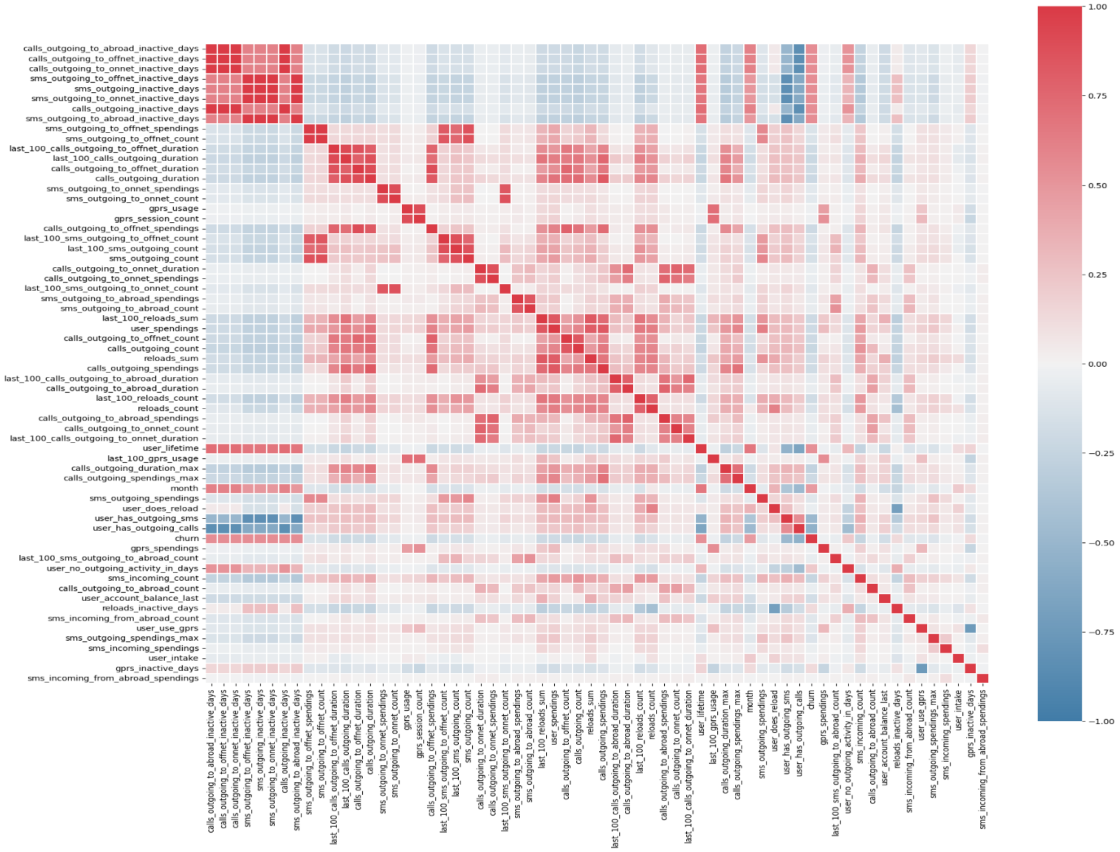
1) First, we checked for null values, and we found no null values in our dataset.

2) We dropped 'user_account_id' as it is unique for every customer and will not contribute towads our prediction. We also dropped 'year' column as it had only one value that is 2013.

3) We checked for correlation and found some highly correlated features therefore we decided to drop features with correlation more than 0.85. Below are the feature names that we dropped and the correlation heatmap.



```
array(['calls_outgoing_to_abroad_inactive_days',
       'sms_outgoing_to_offnet_inactive_days',
       'calls_outgoing_to_onnet_inactive_days',
       'sms_outgoing_to_onnet_inactive_days',
       'sms_outgoing_to_abroad_inactive_days',
       'calls_outgoing_to_offnet_inactive_days',
       'sms_outgoing_to_offnet_spendings',
       'last_100_calls_outgoing_to_offnet_duration',
       'calls_outgoing_to_offnet_duration',
       'sms_outgoing_to_onnet_spendings', 'gprs_usage',
       'last_100_sms_outgoing_to_offnet_count',
       'last_100_calls_outgoing_duration', 'sms_outgoing_to_offnet_count',
       'calls_outgoing_to_onnet_duration', 'last_100_sms_outgoing_count',
       'last_100_sms_outgoing_to_onnet_count'], dtype=object)
```



EXPLORATORY DATA ANALYSIS

SOME OF THE INSIGHTS AND GRAPHS OVER THE VARIABLES ARE SHOWN BELOW

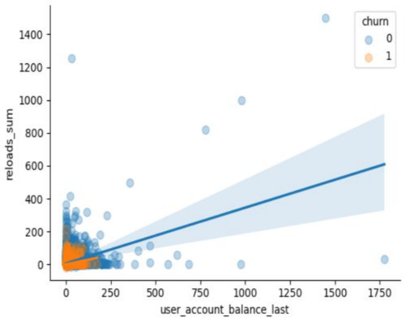


FIG 1

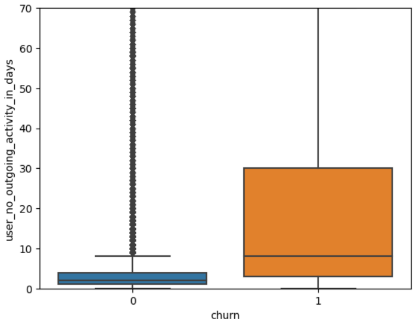


FIG 2

From FIG 1, we can see that if the user with high balance are most likely to not churn and user with low balance are more likely to churn.

From FIG 2, we can see that users with more no outgoing activity are more likely to churn.

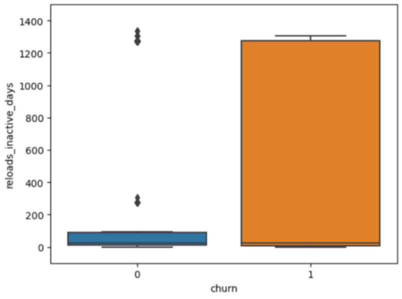


FIG 3

From FIG 3, we can see that higher the reload inactive days higher the churn rate.

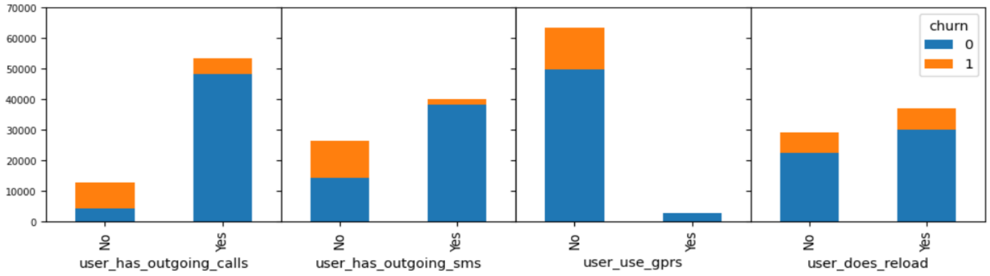


FIG 4

From FIG 4, we can see that:

- The column 'user_has_outgoing_calls' is a good determinant to classify churners and non-churners as the users with no outgoing calls are most likely to churn.
- If the user has outgoing sms and uses gprs, they are less likely to churn.
- The column ‘user_does_reload’ variable doesn't separate well the target variable 'churn'

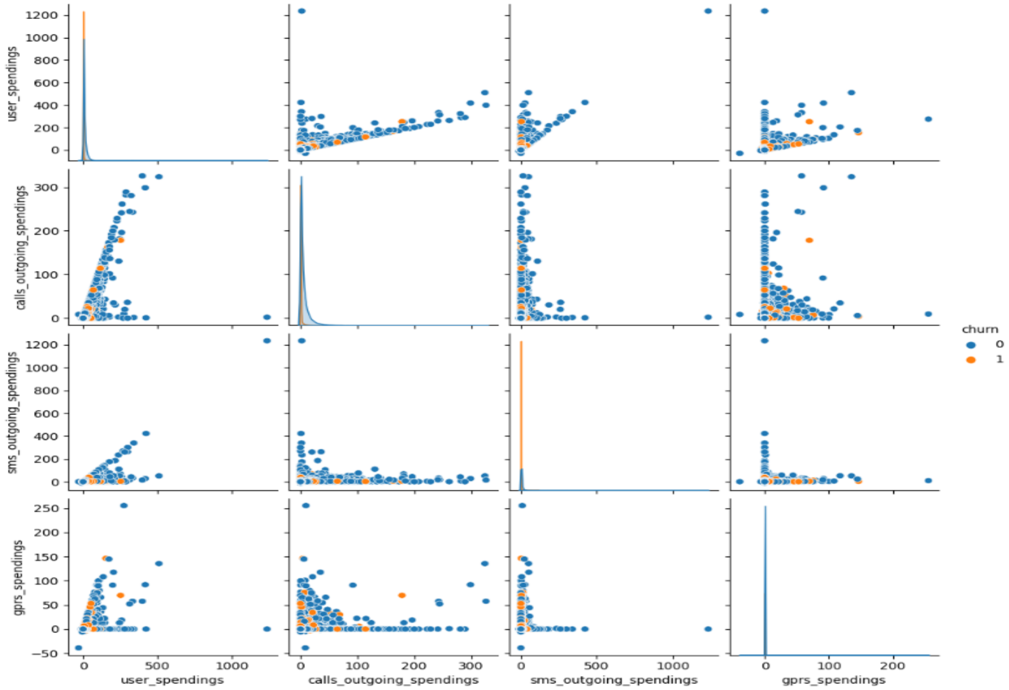


FIG 5

From FIG 5 we can see that high user spending specifically in calls outgoing works well in distinguishing churners and non-churners. similar patterns can be noticed for sms and gprs outgoing services as well.

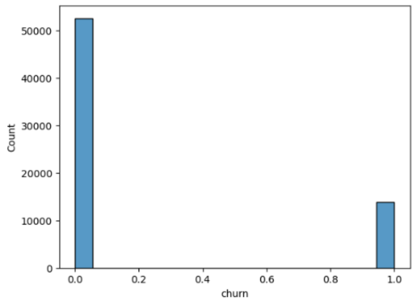
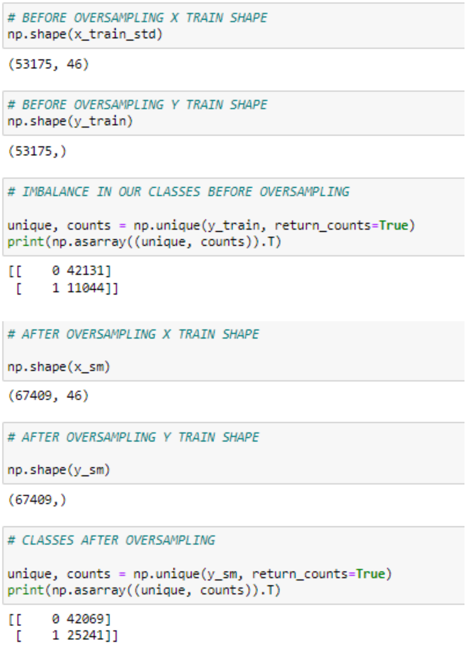


FIG 6

From FIG 6, we can see that our data is imbalanced. we have approximately 52k non churners and 14k churners.

Now as we can see we have an imbalanced dataset so we first spited our dataset into training and testing dataset with 20 percent in test dataset. Then we standardized our training and testing data separately. Now to finally deal with class imbalance we used an oversampling technique known as SMOTE. We used imblearn library to perform SMOTE on our training data. Below are images showing shapes of our training data before and after oversampling.



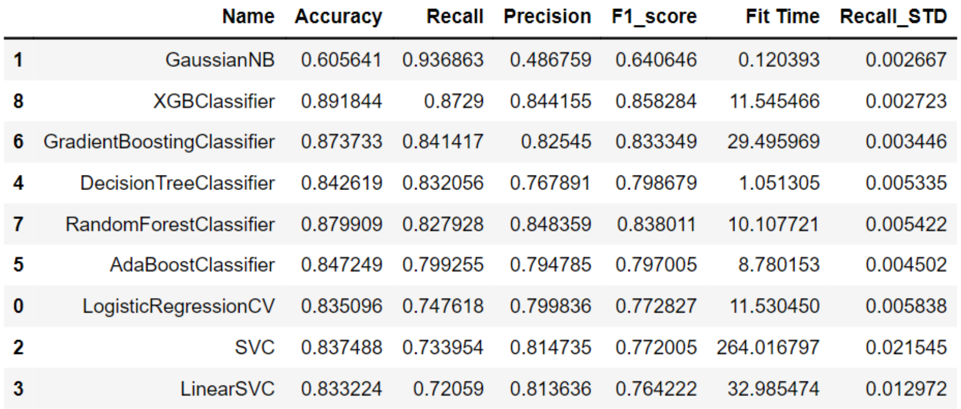
```
# BEFORE OVERSAMPLING X TRAIN SHAPE
np.shape(x_train_std)

(53175, 46)

# BEFORE OVERSAMPLING Y TRAIN SHAPE
np.shape(y_train)

(53175,)

# IMBALANCE IN OUR CLASSES BEFORE OVERSAMPLING

unique, counts = np.unique(y_train, return_counts=True)
print(np.asarray((unique, counts)).T)

[[    0 42131]
 [    1 11044]]

# AFTER OVERSAMPLING X TRAIN SHAPE

np.shape(x_sm)

(67409, 46)

# AFTER OVERSAMPLING Y TRAIN SHAPE

np.shape(y_sm)

(67409,)

# CLASSES AFTER OVERSAMPLING

unique, counts = np.unique(y_sm, return_counts=True)
print(np.asarray((unique, counts)).T)

[[    0 42069]
 [    1 25241]]
```

-→ BEFORE OVERSAMPLING

-→ AFTER OVERSAMPLING

MODEL USED AND PERFORMANCE EVALUATION

- We trained 9 models on training dataset with cross validation. The models we used are listed below with all the mean metrics of our cross validation dataset. Our main metrics here are F1 score and accuracy.
- We calculated Accuracy, precision, recall, F1 score for all of our models.



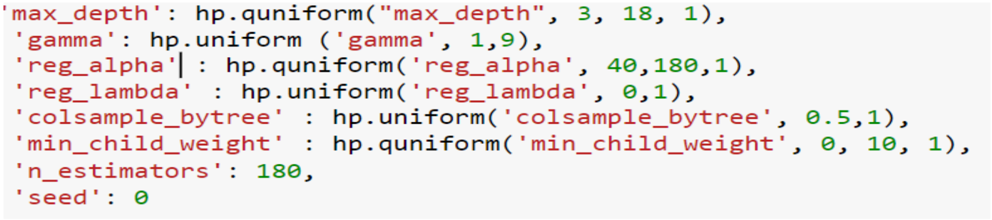
	Name	Accuracy	Recall	Precision	F1_score	Fit Time	Recall_STD
1	GaussianNB	0.605641	0.936863	0.486759	0.640646	0.120393	0.002667
8	XGBClassifier	0.891844	0.8729	0.844155	0.858284	11.545466	0.002723
6	GradientBoostingClassifier	0.873733	0.841417	0.82545	0.833349	29.495969	0.003446
4	DecisionTreeClassifier	0.842619	0.832056	0.767891	0.798679	1.051305	0.005335
7	RandomForestClassifier	0.879909	0.827928	0.848359	0.838011	10.107721	0.005422
5	AdaBoostClassifier	0.847249	0.799255	0.794785	0.797005	8.780153	0.004502
0	LogisticRegressionCV	0.835096	0.747618	0.799836	0.772827	11.530450	0.005838
2	SVC	0.837488	0.733954	0.814735	0.772005	264.016797	0.021545
3	LinearSVC	0.833224	0.72059	0.813636	0.764222	32.985474	0.012972

We can see that many models perform very well. The highest accuracy is achieved by XGBOOST. Highest recall is achieved by Gaussian Naïve bayes. Highest precision by Random Forest model and Highest F1 score by XGBOOST model.

As XGBOOST performs the best in F1 score and is in the top when compared on other models as well. We can see that Ensemble models preform way better than classical models. So we choose XGboost and performed hyperparameter tuning on it.

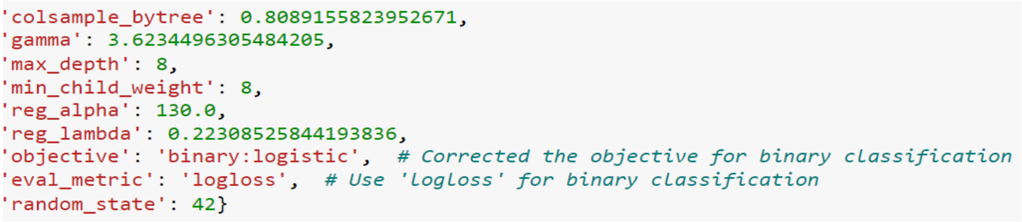
- Hyperparameter Tuning of XGBOOST :

Potential choices of hyperparameter are in the image below -



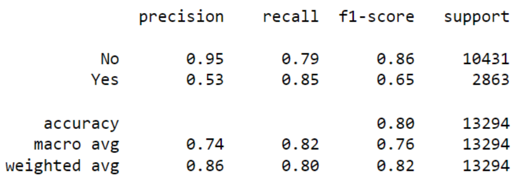
```
'max_depth': hp.quniform("max_depth", 3, 18, 1),
 'gamma': hp.uniform ('gamma', 1,9),
 'reg_alpha' : hp.quniform('reg_alpha', 40,180,1),
 'reg_lambda' : hp.uniform('reg_lambda', 0,1),
 'colsample_bytree' : hp.uniform('colsample_bytree', 0.5,1),
 'min_child_weight' : hp.quniform('min_child_weight', 0, 10, 1),
 'n_estimators': 180,
 'seed': 0
```

The best hyperparameters we got are in the image below –



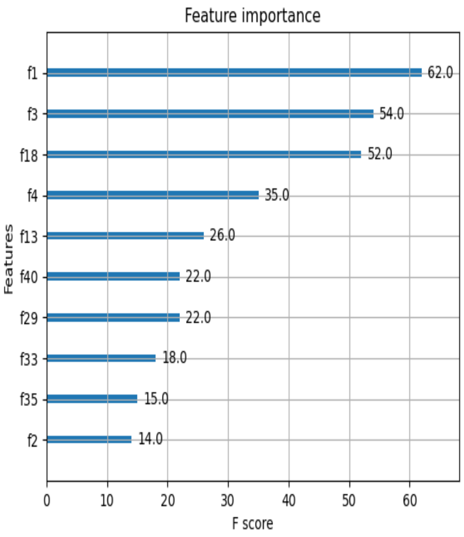
```
'colsample_bytree': 0.8089155823952671,
'gamma': 3.6234496305484205,
'max_depth': 8,
'min_child_weight': 8,
'reg_alpha': 130.0,
'reg_lambda': 0.22308525844193836,
'objective': 'binary:logistic',  # Corrected the objective for binary classification
'eval_metric': 'logloss',  # Use 'logloss' for binary classification
'random_state': 42}
```

- Now we trained another XGBOOST model with the best hyperparameters we got on the whole training dataset and the metrics on test dataset are shown below.

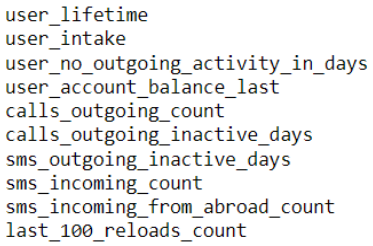


```
              precision    recall  f1-score   support

          No       0.95      0.79      0.86     10431
         Yes       0.53      0.85      0.65      2863

    accuracy                           0.80     13294
   macro avg       0.74      0.82      0.76     13294
weighted avg       0.86      0.80      0.82     13294
```

- Feature Importance



The most important features are f1, f3, f18 etc. which are in respectively shown below.



```
user_lifetime
user_intake
user_no_outgoing_activity_in_days
user_account_balance_last
calls_outgoing_count
calls_outgoing_inactive_days
sms_outgoing_inactive_days
sms_incoming_count
sms_incoming_from_abroad_count
last_100_reloads_count
```

Here we can see that user lifetime, user intake and user no outgoing activity in days are the top three features that prove to be valuable in determining if a customer will stay or not.

CONCLUSION

From the project we conclude that using ensemble learning algorithms, such as XGBoost, performed the best in predicting customer churn with a test accuracy of 80.4%. The XGBClassifier model effectively identified churners and non-churners. The top relevant feature in predicting customer churn rate are user_lifetime, user_intake, user_no_outgoing_activity_in_days, user_account_balance_last and calls_outgoing_count.

Future Scope: We can see that the data was highly imbalanced, therefore there can be a better effort in capturing the data of both churners and non-churners. We can also work on precision of our model for the churners class. If in future we get enough data we can also try using some deep neural net and hope to get better performance metrics.