CSE587 : DATA INTENSIVE COMPUTING

LAB 3 - REPORT

SUBMITTED BY :

DITHYA SRIDHARAN - 50286923

HITESH SANTWANI - 50291123

CUSTOMER RELATIONSHIP PREDICTION FOR MOBILE COMPANY(ORANGE)

ABSTRACT

This project hopes to minimize customer defection by analysing the customer relationship for a mobile company that is spread across Europe by predicting which customers are likely to cancel a subscription. Using data driven methods to better understand and predict customer behaviour is an iterative process which involes :

- Collect by extracting facebook activity, reviews, tweets from ***Facebook and Twitter API*** and compare with historical data collected from ***BigML S3 bucket***.
- Correlating and analysing data across multiple data sources.

For better understanding the customer behaviour, a number of factors are analysed such as :

- Semantic analysis of social media
- Customer usage patterns and geographic usage trends
- Calling-circle data
- Historical data that suggest patterns of churn.

We can use Spark SQL to explore the data, find correlations and feature selection.
Mllib is a new ML library with machine learning routines.
We will use ML pipeline to pass the data through transformers to extract the features and an estimator to produce the model.

PROBLEM STATEMENT

NECESSITY AND IMPORTANCE OF PROBLEM

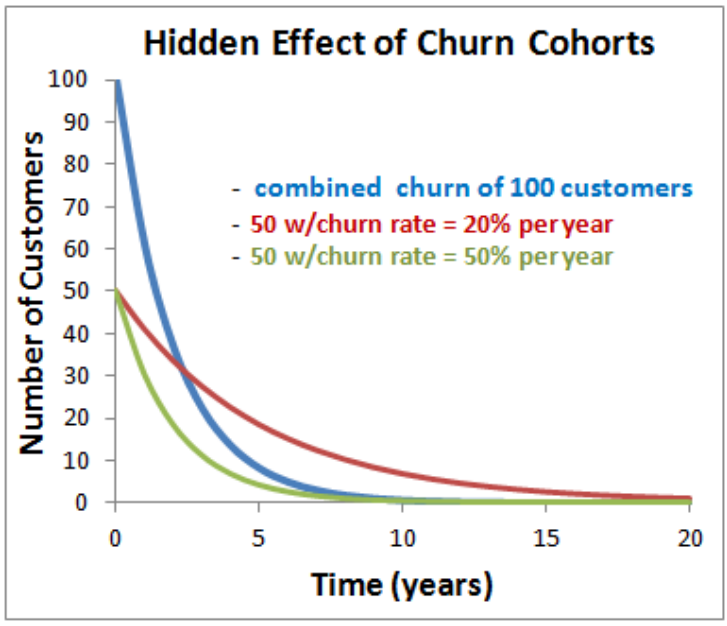
Customer retention is a measure of how many of your customers continue to buy from you over time and are therefore loyal to your brand. ***Churn***, sometimes known as customer attrition, is at the opposite end of the spectrum, i.e. how many customers stop buying from your company.

Industries that use a subscription-based business model have traditionally focused more on churn than others. Banks, telecom companies, insurance firms, energy services companies, are among the many types of businesses that often use customer attrition analysis and customer churn rates as one of their key business metrics.

WHAT WILL BE THE IMPACT

Losing customers is costly for any business. Identifying unhappy customers early on gives you a chance to offer them incentives to stay. Mobile operators have historical records on which customers ultimately ended up churning and which continued using the service and additional data from social media reviews by sematic analysis.

But collecting and measuring social data gives you insights to social shares and engagement. This is especially important for accountability and providing proof to higher-ups on what you need to be doing via social.

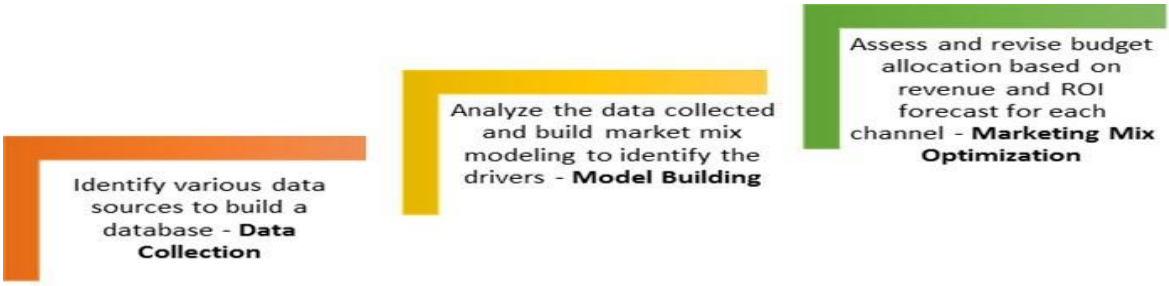


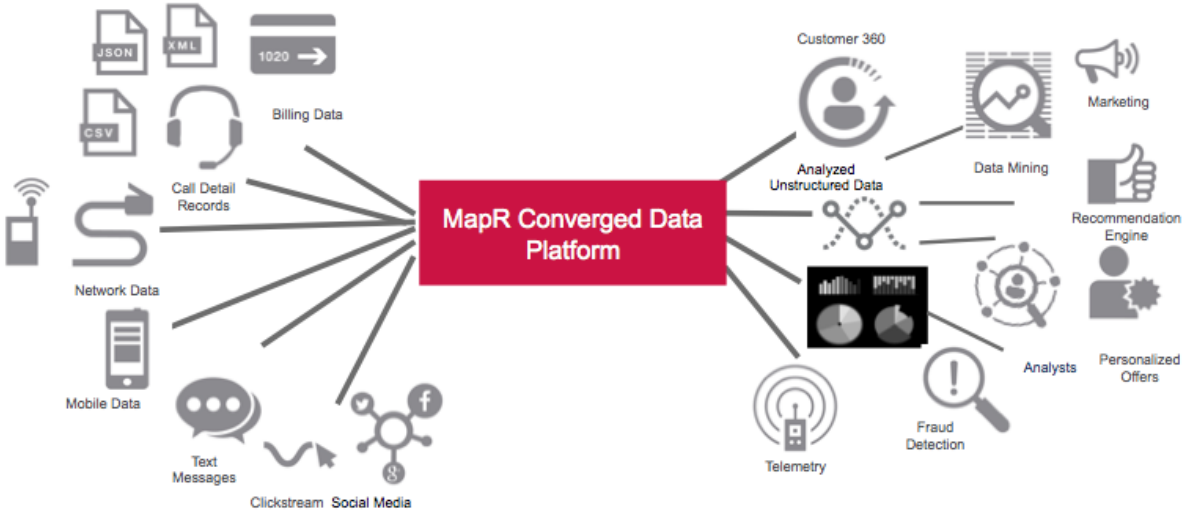
HOW TO SOLVE

To start, choose a time frame for your calculation. Companies typically calculate monthly customer churn rates but you could also do quarterly or annually.
In this timeframe, determine the customers lost and divide it by the total number of customers at the beginning of the month.

We can use this data and information to construct an ML model of one mobile operator's churn using training. After training the model, we can pass the profile information of an arbitrary customer (the same profile information that we used to train the model) to the model, and have the model predict whether this customer is going to churn.

METHODOLOGY

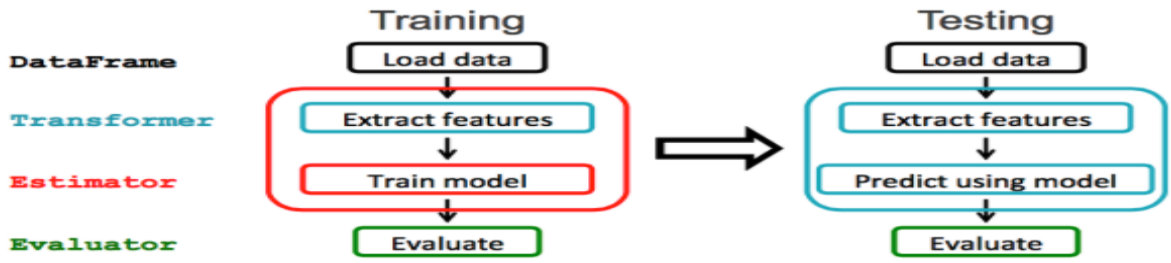


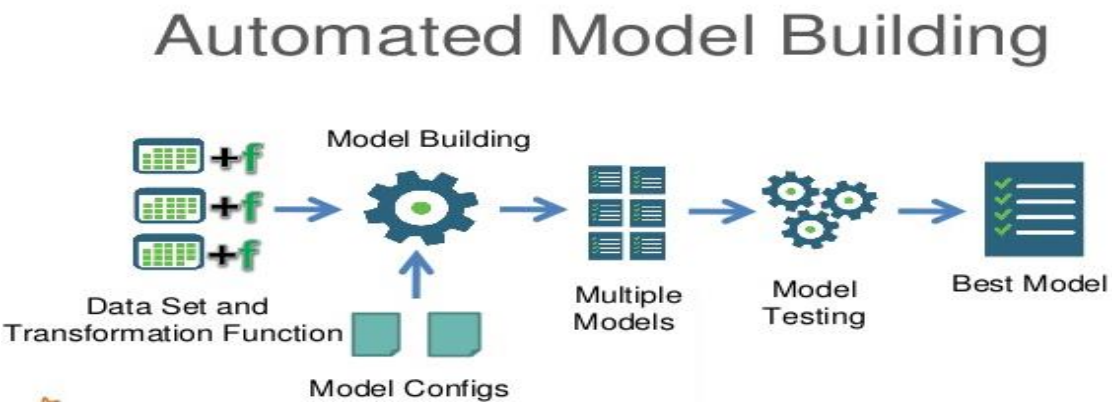


The above diagrams represent the methodologies to be incorporated for customer retention analysis.

PIPELINE ARCHITECTURE

The Mllib package is the newer library of machine learning routines. Spark ML provides a uniform set of high-level APIs built on top of DataFrames.





We will use an ML Pipeline to pass the data through transformers in order to extract the features and an estimator to produce the model.

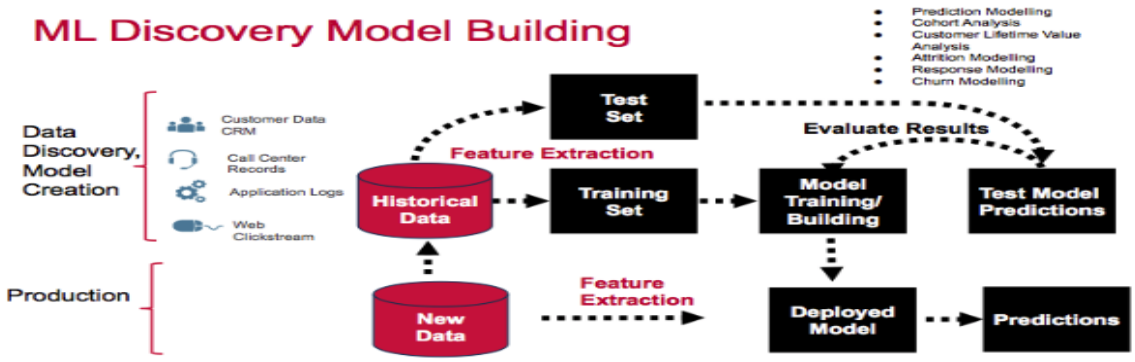
- Transformer: A Transformer is an algorithm which transforms one DataFrame into another DataFrame. We will use a transformer to get a DataFrame with a features vector column.
- Estimator: An Estimator is an algorithm which can be fit on a DataFrame to produce a Transformer (e.g., training/tuning on a DataFrame and producing a model).
- Pipeline: A Pipeline chains multiple Transformers and Estimators together to specify a ML workflow.

ABOUT THE DATA

We'll be using the Orange Telecoms Churn Dataset. It consists of cleaned customer activity data (features), along with a churn label specifying whether the customer canceled the subscription or not. The data can be fetched from ***BigML's S3 bucket***. Along with this historical data, we will be using data driven methods to extract data from social media sites for facebook reviews and tweets from ***Facebook and Twitter API*** to compare and analyse customer satisfaction. We'll use the larger set for training and cross-validation purposes and the smaller set for final testing and model performance evaluation. The historical data has the following following schema.The image shows the first few rows :

State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes	Total eve calls	Total eve charge	Total night minutes	Total night calls	Total night charge	Total intl minutes	Total intl calls	Total intl charge	Customer service calls	Churn
KS	128	415	No	Yes	25	265.1	110	45.07	197.4	99	16.78	244.7	91	11.01	10.0	3	2.7	1	False
OH	107	415	No	Yes	26	161.6	123	27.47	195.5	103	16.62	254.4	103	11.45	13.7	3	3.7	1	False
NJ	137	415	No	No	0	243.4	114	41.38	121.2	110	10.3	162.6	104	7.32	12.2	5	3.29	0	False
OH	84	408	Yes	No	0	299.4	71	50.9	61.9	88	5.26	196.9	89	8.86	6.6	7	1.78	2	False
OK	75	415	Yes	No	0	166.7	113	28.34	148.3	122	12.61	186.9	121	8.41	10.1	3	2.73	3	False

SOLUTION



The above diagram describes ML model building for a Customer360 insights analysis to understand and predict customer behavior using ML techniques.

Classification is a family of supervised machine learning algorithms that identify which category an item belongs to (e.g., whether a transaction is fraud or not fraud), based on labeled examples of known items (e.g., transactions known to be fraud or not). Classification takes a set of data with known labels and pre-determined features and learns how to label new records based on that information.

For the given use case, we are trying to predict :

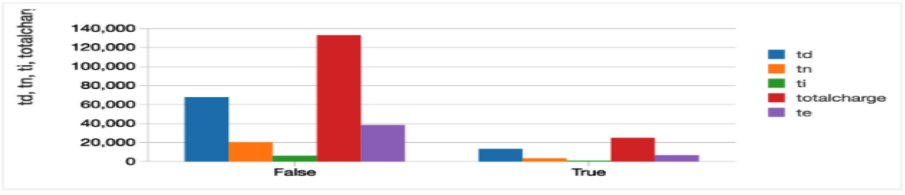
- Whether a customer has a high probability of unsubscribing from the service or not
- Churn is the Label: True or False

What are the "if questions" or properties that you can use to make predictions?

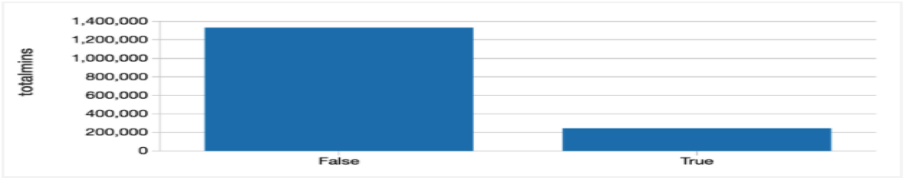
- Call statistics, customer service calls, etc.
- Social media reviews and activity(Sematic analysis)
- To build a classifier model, you extract the features of interest that most contribute to the classification.

This is run on Spark 2.0.1
Data exploration is done by Spark SQL. Here are some example queries using the Scala DataFrame API. This is taken as reference from sample dataset.

```
%sql
SELECT churn,
   SUM(tdcharge) as td,
   SUM(techarge) as te,
   SUM(tncharge) as tn,
   SUM(ticharge) as ti,
   SUM(tdcharge) + SUM(techarge) + SUM(tncharge) + SUM(ticharge) as totalcharge
FROM account
GROUP BY churn
ORDER BY totalcharge DESC
```




```
%sql
SELECT churn,
   SUM(tdmins) + SUM(temins) + SUM(tnmins) + SUM(timins) as totalmins
FROM account
GROUP BY churn
```



SNIPPET SHOTS OF MLLIB DECISION TREE MODELLING RUN ON HISTORICAL DATA FOR CHURN(NOT ENTIRE DATA COLLECTED)

In [6]:

```
"""
Decision Tree Classification Example.
"""
from __future__ import print_function

from pyspark import SparkContext
# $example on$
from pyspark.mllib.tree import DecisionTree, DecisionTreeModel
from pyspark.mllib.util import MLUtils
# $example off$


sc = SparkContext.getOrCreate()

# $example on$
# Load and parse the data file into an RDD of LabeledPoint.
data = MLUtils.loadLibSVMFile(sc, 's3://datacc/sample_libsvm_data.txt')
# Split the data into training and test sets (30% held out for testing)
(trainingData, testData) = data.randomSplit([0.7, 0.3])

# Train a DecisionTree model.
#  Empty categoricalFeaturesInfo indicates all features are continuous.
model = DecisionTree.trainClassifier(trainingData, numClasses=2, categoricalFeaturesInfo={},
                                     impurity='gini', maxDepth=5, maxBins=32)
# Evaluate model on test instances and compute test error
predictions = model.predict(testData.map(lambda x: x.features))
labelsAndPredictions = testData.map(lambda lp: lp.label).zip(predictions)
testErr = labelsAndPredictions.filter(
        lambda lp: lp[0] != lp[1]).count() / float(testData.count())
print('Test Error = ' + str(testErr))
print('Learned classification tree model:')
print(model.toDebugString())

# Save and load model
model.save(sc, "target/tmp/myDecisionTreeClassificationModel")
sameModel = DecisionTreeModel.load(sc, "target/tmp/myDecisionTreeClassificationModel")
```

```
model.save(sc, "target/tmp/myDecisionTreeClassificationModel")
sameModel = DecisionTreeModel.load(sc, "target/tmp/myDecisionTreeClassificationModel")
# $example off$
```

▾ Spark Job Progress

Progress for count at DecisionTreeMetadata.scala:118 Job Progress: 2/2 Tasks Complete

Stage [ID]: name at [source]:[line]	Status	Task Progress	Elapsed Time (seconds)	Failed Task Logs
Stage [4]: count at Decisio...la:118	COMPLETE	2/2	0.081	

▾ Job [5]: collectAsMap at RandomForest.scala:927

Progress for collectAsMap at RandomForest.scala:927 Job Progress: 4/4 Tasks Complete

Stage [ID]: name at [source]:[line]	Status	Task Progress	Elapsed Time (seconds)	Failed Task Logs
Stage [5]: flatMap at Rando...la:919	COMPLETE	2/2	1.024	
Stage [6]: collectAsMap at ...la:927	COMPLETE	2/2	0.468	

▾ Job [6]: collectAsMap at RandomForest.scala:567

```
----------------------------------------
Test Error = 0.0
Learned classification tree model:
DecisionTreeModel classifier of depth 2 with 5 nodes
  If (feature 434 <= 88.5)
   If (feature 100 <= 193.5)
    Predict: 0.0
   Else (feature 100 > 193.5)
    Predict: 1.0
  Else (feature 434 > 88.5)
   Predict: 1.0
```

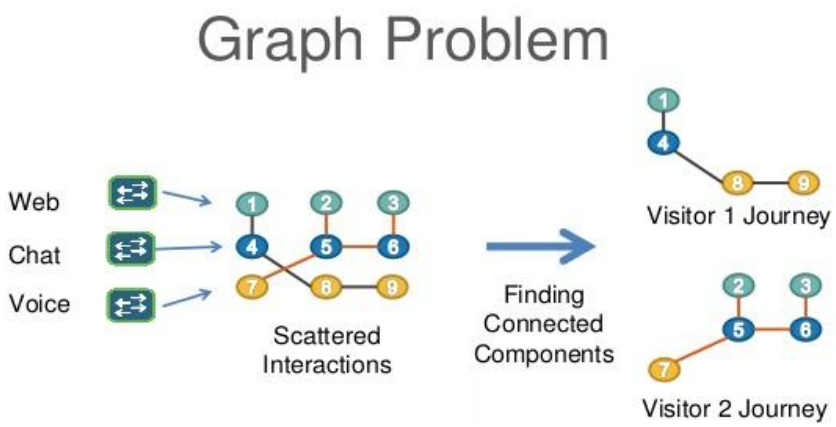
Given the data set at hand, we would like to determine which parameter values of the ***decision tree*** produce the best model. The ML package supports ***k-fold cross validation***, which can be readily coupled with a parameter grid builder and an evaluator to construct a model selection workflow. The prediction probabilities can be very useful in ranking customers by their likeliness to defect. This way, the limited resources available to the business for retention can be focused on the appropriate customers.

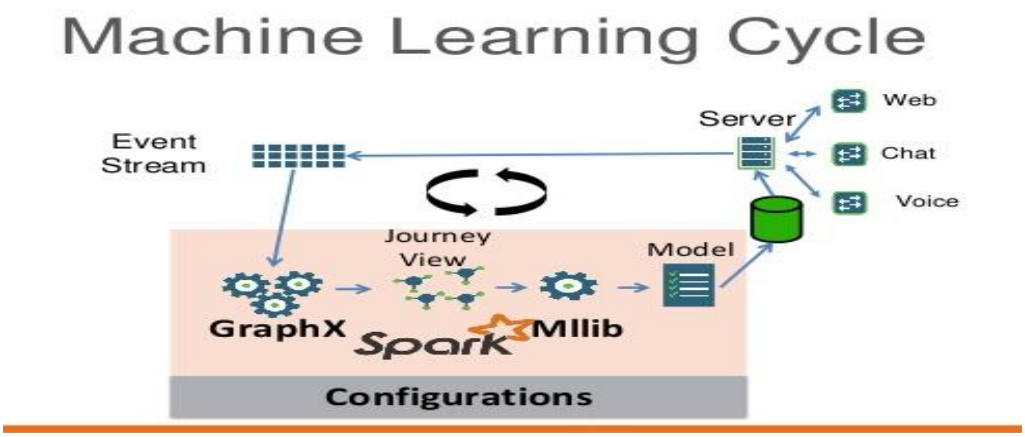
OUTCOMES :

The actual performance of the model can be determined using the test data set that has not been used for any training or cross-validation activities. We'll transform the test set with the model pipeline, which will map the features according to the same recipe. The prediction probabilities can be very useful in ranking customers by their likeliness to defect. This way, the limited resources available to the business for retention can be focused on the appropriate customers. Below, we calculate some more metrics. The number of false/true positive and negative predictions is also useful:

		% Predicted class	
		0	1
% True class	0	TN - Percentage of predictions correctly predicting no churn	FP - Percentage of predictions erroneously predicting churn
	1	FN - Percentage of predictions erroneously predicting no churn	TP - Percentage of predictions correctly predicting churn

VISUALISATIONS OF THE ANALYTICS :





SUMMARY

To reduce customer churn and become customer-obsessed is a technological, organizational and cultural shift. It requires consistent effort and focus over a period of time.

Your training data and your monetary cost assignments could be more complex, or you might have to build multiple models for each type of churn. Regardless of the added complexity, the same principles described in this report will likely apply.

REFERENCES

https://databricks.com/session/automated-machine-learning-using-spark-mllib-to-improve-customer-experience

https://www.pointillist.com/blog/reduce-churn-customer-journey-analytics/

https://sproutsocial.com/insights/social-media-data/

http://chaotic-flow.com/saas-metrics-faqs-what-is-churn/

https://mapr.com/blog/big-data-opportunities-telecommunications/