**Logistic Regression Consulting Project**

**Binary Customer Churn**

A marketing agency has many customers that use their service to produce ads for the client/customer websites. They've noticed that they have quite a bit of churn in clients. They basically randomly assign account managers right now, but want you to create a machine learning model that will help predict which customers will churn (stop buying their service) so that they can correctly assign the customers most at risk to churn an account manager. Luckily they have some historical data, can you help them out? Create a classification algorithm that will help classify whether or not a customer churned. Then the company can test this against incoming data for future customers to predict which customers will churn and assign them an account manager.

The data is saved as customer\_churn.csv. Here are the fields and their definitions:

Name : Name of the latest contact at Company

Age: Customer Age

Total\_Purchase: Total Ads Purchased

Account\_Manager: Binary 0=No manager, 1= Account manager assigned

Years: Totaly Years as a customer

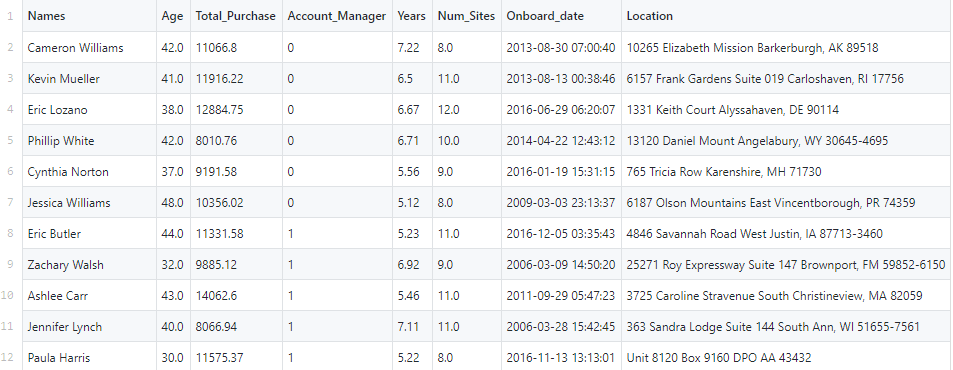
Num\_sites: Number of websites that use the service.

Onboard\_date: Date that the name of the latest contact was onboarded

Location: Client HQ Address

Company: Name of Client Company

Once you've created the model and evaluated it, test out the model on some new data (you can think of this almost like a hold-out set) that your client has provided, saved under new\_customers.csv. The client wants to know which customers are most likely to churn given this data (they don't have the label yet).



**from** **pyspark.sql** **import** SparkSession

spark = SparkSession.builder.appName('logregconsult').getOrCreate()

data = spark.read.csv('customer\_churn.csv',inferSchema=**True**,

header=**True**)

data.printSchema()

root

|-- Names: string (nullable = true)

|-- Age: double (nullable = true)

|-- Total\_Purchase: double (nullable = true)

|-- Account\_Manager: integer (nullable = true)

|-- Years: double (nullable = true)

|-- Num\_Sites: double (nullable = true)

|-- Onboard\_date: timestamp (nullable = true)

|-- Location: string (nullable = true)

|-- Company: string (nullable = true)

|-- Churn: integer (nullable = true)

data.describe().show()

data.columns

### Format for MLlib

We'll ues the numerical columns. We'll include Account Manager because its easy enough, but keep in mind it probably won't be any sort of a signal because the agency mentioned its randomly assigned!

**from** **pyspark.ml.feature** **import** VectorAssembler

assembler = VectorAssembler(inputCols=['Age',

'Total\_Purchase',

'Account\_Manager',

'Years',

'Num\_Sites'],outputCol='features')

output = assembler.transform(data)

final\_data = output.select('features','churn')

**Test Train Split**

train\_churn,test\_churn = final\_data.randomSplit([0.7,0.3])

**Fit the model**

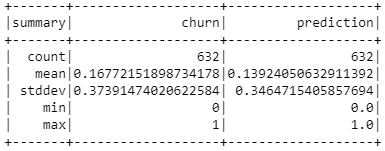
**from** **pyspark.ml.classification** **import** LogisticRegression

lr\_churn = LogisticRegression(labelCol='churn')

fitted\_churn\_model = lr\_churn.fit(train\_churn)

training\_sum = fitted\_churn\_model.summary

training\_sum.predictions.describe().show()



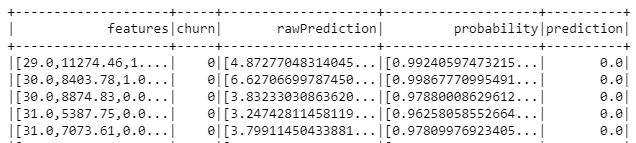
Evaluate results

Let's evaluate the results on the data set we were given (using the test data)

**from** **pyspark.ml.evaluation** **import** BinaryClassificationEvaluator

pred\_and\_labels = fitted\_churn\_model.evaluate(test\_churn)

pred\_and\_labels.predictions.show()



### Using AUC

churn\_eval = BinaryClassificationEvaluator(rawPredictionCol='prediction', labelCol='churn')

auc = churn\_eval.evaluate(pred\_and\_labels.predictions)

auc

### Predict on brand new unlabeled data

We still need to evaluate the new\_customers.csv file!

final\_lr\_model = lr\_churn.fit(final\_data)

new\_customers = spark.read.csv('new\_customers.csv',inferSchema=**True**,

header=**True**)

new\_customers.printSchema()

root

|-- Names: string (nullable = true)

|-- Age: double (nullable = true)

|-- Total\_Purchase: double (nullable = true)

|-- Account\_Manager: integer (nullable = true)

|-- Years: double (nullable = true)

|-- Num\_Sites: double (nullable = true)

|-- Onboard\_date: timestamp (nullable = true)

|-- Location: string (nullable = true)

|-- Company: string (nullable = true)

test\_new\_customers = assembler.transform(new\_customers)

test\_new\_customers.printSchema()

root

|-- Names: string (nullable = true)

|-- Age: double (nullable = true)

|-- Total\_Purchase: double (nullable = true)

|-- Account\_Manager: integer (nullable = true)

|-- Years: double (nullable = true)

|-- Num\_Sites: double (nullable = true)

|-- Onboard\_date: timestamp (nullable = true)

|-- Location: string (nullable = true)

|-- Company: string (nullable = true)

|-- features: vector (nullable = true)

final\_results = final\_lr\_model.transform(test\_new\_customers)

final\_results.select('Company','prediction').show()