Customer Churn

Customer churn refers to the loss of customers of any given company or product. This project aims to predict the likelihood of voluntary customer churn at Telco, a large telecommunications company based in the United States. Using this information, the company would be able to focus efforts on a small subset of customers predicted to churn, saving them valuable time and resources while increasing their effective retention rate.

Data

Telco Customer Churn

The data was downloaded from IBM Sample Data Sets (now on Kaggle): https://www.kaggle.com/datasets/blastchar/telco-customer-churn

Each row represents a customer, each column contains customer's attributes described as below:

- **customerID**: Customer ID
- **gender**: Customer gender (female, male)
- **SeniorCitizen**: Whether the customer is a senior citizen or not (1, 0)
- **Partner**: Whether the customer has a partner or not (Yes, No)
- **Dependents**: Whether the customer has dependents or not (Yes, No)
- **tenure**: Number of months the customer has stayed with the company
- **PhoneService**: Whether the customer has a phone service or not (Yes, No)
- **MultipleLines**: Whether the customer has multiple lines or not (Yes, No, No phone service)
- **InternetService**: Customer's internet service provider (DSL, Fiber optic, No)
- **OnlineSecurity**: Whether the customer has online security or not (Yes, No, No internet service)
- **OnlineBackup**: Whether the customer has online backup or not (Yes, No, No internet service)
- **DeviceProtection**: Whether the customer has device protection or not (Yes, No, No internet service)
- **TechSupport**: Whether the customer has tech support or not (Yes, No, No internet service)
- **StreamingTV**: Whether the customer has streaming TV or not (Yes, No, No internet service)
- **StreamingMovies**: Whether the customer has streaming movies or not (Yes, No, No internet service)
- **Contract**: The contract term of the customer (Month-to-month, One year, Two year)
- **PaperlessBilling**: Whether the customer has paperless billing or not (Yes, No)

- **PaymentMethod**: The customer's payment method (Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic))
- **MonthlyCharges**: The amount charged to the customer monthly
- **TotalCharges**: The total amount charged to the customer
- **Churn**: Whether the customer churned or not (Yes or No)

The data set includes information about:

- Customers who left - the column is called `Churn`
- Services that each customer has signed up for - phone, multiple lines, internet, online security, online backup, device protection, tech support, and streaming TV and movies
- Customer account information - how long they've been a customer, contract, payment method, paperless billing, monthly charges, and total charges
- Demographic info about customers - gender, age range, and if they have partners and dependents

Importing libraries:

In [1]:
```
%matplotlib inline
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from pyspark.sql.functions import *

# the following line gets the bucket name attached to our cluster
bucket = spark._jsc.hadoopConfiguration().get("fs.gs.system.bucket")

# specifying the path to our bucket where the data is located (no need to edit this
data = "gs://" + bucket + "/notebooks/jupyter/data/"
print(data)
```

```
gs://pstat135-wtran/notebooks/jupyter/data/
```

Get the data from here: https://github.com/UCSB-PSTAT-135-235/Winter2023/blob/public/01-Introduction/data/Telco-Customer-Churn.csv

Importing data:

In [2]:
```
df = spark.read.format("csv")\
  .option("header", "true")\
  .option("inferschema", True)\
  .load(data + "Telco-Customer-Churn.csv")\
  .coalesce(5)

df = df.drop('customerID')  # Dropping customerID
df.cache()
df.show(5)
df.printSchema()
print("This datasets consists of {} rows.".format(df.count()))
```

```
+------+-------------+-------+----------+------+------------+----------------+----
-----------+--------------+------------+----------------+-----------+-----------+-
--------------+--------------+----------------+--------------------+--------------
+------------+-----+
|gender|SeniorCitizen|Partner|Dependents|tenure|PhoneService|   MultipleLines|Inte
rnetService|OnlineSecurity|OnlineBackup|DeviceProtection|TechSupport|StreamingTV|S
treamingMovies|      Contract|PaperlessBilling|       PaymentMethod|MonthlyCharges
|TotalCharges|Churn|
+------+-------------+-------+----------+------+------------+----------------+----
-----------+--------------+------------+----------------+-----------+-----------+-
--------------+--------------+----------------+--------------------+--------------
+------------+-----+
|Female|            0|    Yes|        No|     1|          No|No phone service|
DSL|            No|         Yes|              No|         No|         No|
No|Month-to-month|            Yes|    Electronic check|         29.85|       29.8
5|   No|
|  Male|            0|     No|        No|    34|         Yes|              No|
DSL|           Yes|          No|             Yes|         No|         No|
No|      One year|             No|        Mailed check|         56.95|      1889.
5|   No|
|  Male|            0|     No|        No|     2|         Yes|              No|
DSL|           Yes|         Yes|              No|         No|         No|
No|Month-to-month|            Yes|        Mailed check|         53.85|      108.1
5|  Yes|
|  Male|            0|     No|        No|    45|          No|No phone service|
DSL|           Yes|          No|             Yes|        Yes|         No|
No|      One year|             No|Bank transfer (au...|          42.3|     1840.7
5|   No|
|Female|            0|     No|        No|     2|         Yes|              No|
Fiber optic|           No|          No|              No|         No|         No|
No|Month-to-month|            Yes|    Electronic check|          70.7|      151.6
5|  Yes|
+------+-------------+-------+----------+------+------------+----------------+----
-----------+--------------+------------+----------------+-----------+-----------+-
--------------+--------------+----------------+--------------------+--------------
+------------+-----+
only showing top 5 rows

root
 |-- gender: string (nullable = true)
 |-- SeniorCitizen: integer (nullable = true)
 |-- Partner: string (nullable = true)
 |-- Dependents: string (nullable = true)
 |-- tenure: integer (nullable = true)
 |-- PhoneService: string (nullable = true)
 |-- MultipleLines: string (nullable = true)
 |-- InternetService: string (nullable = true)
 |-- OnlineSecurity: string (nullable = true)
 |-- OnlineBackup: string (nullable = true)
 |-- DeviceProtection: string (nullable = true)
 |-- TechSupport: string (nullable = true)
 |-- StreamingTV: string (nullable = true)
 |-- StreamingMovies: string (nullable = true)
 |-- Contract: string (nullable = true)
 |-- PaperlessBilling: string (nullable = true)
 |-- PaymentMethod: string (nullable = true)
```

```
 |-- MonthlyCharges: double (nullable = true)
 |-- TotalCharges: string (nullable = true)
 |-- Churn: string (nullable = true)

This datasets consists of 7043 rows.
```

Checking for missing values:

In [3]:
```
df = df.withColumn("TotalCharges", df.TotalCharges.cast('double'))

[(c, df.where(col(c).isNull()).count()) for c in df.columns]
```

Out[3]:
```
[('gender', 0),
 ('SeniorCitizen', 0),
 ('Partner', 0),
 ('Dependents', 0),
 ('tenure', 0),
 ('PhoneService', 0),
 ('MultipleLines', 0),
 ('InternetService', 0),
 ('OnlineSecurity', 0),
 ('OnlineBackup', 0),
 ('DeviceProtection', 0),
 ('TechSupport', 0),
 ('StreamingTV', 0),
 ('StreamingMovies', 0),
 ('Contract', 0),
 ('PaperlessBilling', 0),
 ('PaymentMethod', 0),
 ('MonthlyCharges', 0),
 ('TotalCharges', 11),
 ('Churn', 0)]
```

Filling missing values with zeroes:

In [4]:
```
df = df.fillna(0)
```

Defining an RFormula that uses all of the columns as features:

In [5]:
```
# Your answer goes here
from pyspark.ml.feature import RFormula

supervised = RFormula(formula="Churn ~ .")
```

Fitting the RFormula transformer:

In [6]:
```
fittedRF = supervised.fit(df)
```

Using `fittedRF` transform our `df` DataFrame:

In [7]:
```
preparedDF = fittedRF.transform(df)
```

Printing the first couple of rows of `preparedDF`:

In [8]:
```
preparedDF.show(truncate=False)
```

```
23/03/07 20:59:08 WARN org.apache.spark.sql.catalyst.util.package: Truncated the s
tring representation of a plan since it was too large. This behavior can be adjust
ed by setting 'spark.sql.debug.maxToStringFields'.
```

```
+------+-------------+-------+----------+------+------------+---------------+----
-----------+------------------+------------------+------------------+----------
---------+------------------+------------------+-------------+---------------+
------------------------+-------------+------------+-----+--------------------
--------------------------------------------------------------------------------
------------------------------+-----+
|gender|SeniorCitizen|Partner|Dependents|tenure|PhoneService|MultipleLines   |Inte
rnetService|OnlineSecurity     |OnlineBackup       |DeviceProtection   |TechSuppor
t        |StreamingTV        |StreamingMovies    |Contract      |PaperlessBilling|
PaymentMethod            |MonthlyCharges|TotalCharges|Churn|features
|label|
+------+-------------+-------+----------+------+------------+---------------+----
-----------+------------------+------------------+------------------+----------
---------+------------------+------------------+-------------+---------------+
------------------------+-------------+------------+-----+--------------------
--------------------------------------------------------------------------------
------------------------------+-----+
|Female|0            |Yes    |No        |1     |No          |No phone service|DSL
|No                |Yes               |No                |No                |N
o                 |No                |Month-to-month|Yes             |Electronic
check         |29.85         |29.85       |No   |(30,[3,4,9,10,13,14,16,18,20,22,2
4,25,28,29],[1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,29.85,29.85])
|0.0  |
|Male  |0            |No     |No        |34    |Yes         |No              |DSL
|Yes               |No                |Yes               |No                |N
o                 |No                |One year      |No              |Mailed chec
k             |56.95         |1889.5      |No   |(30,[0,2,3,4,5,6,9,11,12,15,16,1
8,20,26,28,29],[1.0,1.0,1.0,34.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,56.95,188
9.5])              |0.0  |
|Male  |0            |No     |No        |2     |Yes         |No              |DSL
|Yes               |Yes               |No                |No                |N
o                 |No                |Month-to-month|Yes             |Mailed chec
k             |53.85         |108.15      |Yes  |(30,[0,2,3,4,5,6,9,11,13,14,16,1
8,20,22,24,26,28,29],[1.0,1.0,1.0,2.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,
1.0,53.85,108.15])  |1.0  |
|Male  |0            |No     |No        |45    |No          |No phone service|DSL
|Yes               |No                |Yes               |Yes               |N
o                 |No                |One year      |No              |Bank transf
er (automatic)|42.3          |1840.75     |No   |(30,[0,2,3,4,9,11,12,15,17,18,20,
27,28,29],[1.0,1.0,1.0,45.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,42.3,1840.75])
|0.0  |
|Female|0            |No     |No        |2     |Yes         |No              |Fibe
r optic    |No                |No                |No                |No
|No                |No                |Month-to-month|Yes             |Electroni
c check         |70.7          |151.65      |Yes  |(30,[2,3,4,5,6,8,10,12,14,16,1
8,20,22,24,25,28,29],[1.0,1.0,2.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,
70.7,151.65])         |1.0  |
|Female|0            |No     |No        |8     |Yes         |Yes             |Fibe
r optic    |No                |No                |Yes               |No
|Yes               |Yes               |Month-to-month|Yes             |Electroni
c check         |99.65         |820.5       |Yes  |(30,[2,3,4,5,7,8,10,12,15,16,1
9,21,22,24,25,28,29],[1.0,1.0,8.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,
99.65,820.5])         |1.0  |
|Male  |0            |No     |Yes       |22    |Yes         |Yes             |Fibe
r optic    |No                |Yes               |No                |No
|Yes               |No                |Month-to-month|Yes             |Credit ca
```

```
rd (automatic)  |89.1          |1949.4      |No   |(30,[0,2,4,5,7,8,10,13,14,16,1
9,20,22,24,28,29],[1.0,1.0,22.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,89.1,1
949.4])               |0.0  |
|Female|0            |No     |No        |10    |No          |No phone service|DSL
|Yes                |No                |No                  |No                |N
o                  |No                |Month-to-month|No             |Mailed chec
k             |29.75         |301.9       |No   |(30,[2,3,4,9,11,12,14,16,18,20,2
2,26,28,29],[1.0,1.0,10.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,29.75,301.9])
|0.0  |
|Female|0            |Yes    |No        |28    |Yes         |Yes             |Fibe
r optic    |No                |No                |Yes               |Yes
|Yes                |Yes               |Month-to-month|Yes             |Electroni
c check         |104.8         |3046.05     |Yes  |(30,[3,4,5,7,8,10,12,15,17,19,2
1,22,24,25,28,29],[1.0,28.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,104.8,
3046.05])            |1.0  |
|Male  |0            |No     |Yes       |62    |Yes         |No              |DSL
|Yes                |Yes               |No                  |No                |N
o                  |No                |One year      |No             |Bank transf
er (automatic)|56.15         |3487.95     |No   |(30,[0,2,4,5,6,9,11,13,14,16,18,2
0,27,28,29],[1.0,1.0,62.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,56.15,3487.95])
|0.0  |
|Male  |0            |Yes    |Yes       |13    |Yes         |No              |DSL
|Yes                |No                |No                  |No                |N
o                  |No                |Month-to-month|Yes             |Mailed chec
k             |49.95         |587.45      |No   |(30,[0,4,5,6,9,11,12,14,16,18,20,
22,24,26,28,29],[1.0,13.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,49.95,58
7.45])             |0.0  |
|Male  |0            |No     |No        |16    |Yes         |No              |No
|No internet service|No internet service|No internet service|No internet service|N
o internet service|No internet service|Two year      |No             |Credit card
(automatic)  |18.95         |326.8       |No   |(30,[0,2,3,4,5,6,23,28,29],[1.0,1.
0,1.0,16.0,1.0,1.0,1.0,18.95,326.8])
|0.0  |
|Male  |0            |Yes    |No        |58    |Yes         |Yes             |Fibe
r optic    |No                |No                |Yes               |No
|Yes                |Yes               |One year      |No              |Credit ca
rd (automatic)  |100.35        |5681.1      |No   |(30,[0,3,4,5,7,8,10,12,15,16,1
9,21,28,29],[1.0,1.0,58.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,100.35,5681.1])
|0.0  |
|Male  |0            |No     |No        |49    |Yes         |Yes             |Fibe
r optic    |No                |Yes               |Yes               |No
|Yes                |Yes               |Month-to-month|Yes             |Bank tran
sfer (automatic)|103.7         |5036.3      |Yes  |(30,[0,2,3,4,5,7,8,10,13,15,16,
19,21,22,24,27,28,29],[1.0,1.0,1.0,49.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.
0,1.0,103.7,5036.3]) |1.0  |
|Male  |0            |No     |No        |25    |Yes         |No              |Fibe
r optic    |Yes               |No                |Yes               |Yes
|Yes                |Yes               |Month-to-month|Yes             |Electroni
c check         |105.5         |2686.05     |No   |(30,[0,2,3,4,5,6,8,11,12,15,17,
19,21,22,24,25,28,29],[1.0,1.0,1.0,25.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.
0,1.0,105.5,2686.05])|0.0  |
|Female|0            |Yes    |Yes       |69    |Yes         |Yes             |Fibe
r optic    |Yes               |Yes               |Yes               |Yes
|Yes                |Yes               |Two year      |No              |Credit ca
rd (automatic)  |113.25        |7895.15     |No   |(30,[4,5,7,8,11,13,15,17,19,21,
23,28,29],[69.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,113.25,7895.15])
```

```
|0.0  |
|Female|0            |No     |No        |52    |Yes         |No              |No
|No internet service|No internet service|No internet service|No internet service|N
o internet service|No internet service|One year      |No              |Mailed chec
k             |20.65         |1022.95     |No   |(30,[2,3,4,5,6,26,28,29],[1.0,1.
0,52.0,1.0,1.0,1.0,20.65,1022.95])
|0.0  |
|Male  |0            |No     |Yes       |71    |Yes         |Yes             |Fibe
r optic    |Yes               |No                |Yes               |No
|Yes                |Yes               |Two year      |No              |Bank tran
sfer (automatic)|106.7         |7382.25     |No   |(30,[0,2,4,5,7,8,11,12,15,16,1
9,21,23,27,28,29],[1.0,1.0,71.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,106.7,
7382.25])             |0.0  |
|Female|0            |Yes    |Yes       |10    |Yes         |No              |DSL
|No                 |No                |Yes               |Yes               |N
o                  |No                |Month-to-month|No              |Credit card
(automatic)  |55.2          |528.35      |Yes  |(30,[4,5,6,9,10,12,15,17,18,20,22,
28,29],[10.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,55.2,528.35])
|1.0  |
|Female|0            |No     |No        |21    |Yes         |No              |Fibe
r optic    |No                |Yes               |Yes               |No
|No                 |Yes               |Month-to-month|Yes             |Electroni
c check         |90.05         |1862.9      |No   |(30,[2,3,4,5,6,8,10,13,15,16,1
8,21,22,24,25,28,29],[1.0,1.0,21.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.
0,90.05,1862.9])       |0.0  |
+------+------------+-------+----------+------+------------+----------------+----
-----------+------------------+------------------+------------------+----------
---------+------------------+------------------+--------------+----------------+
-------------------------+--------------+------------+-----+-------------------
-------------------------------------------------------------------------------
------------------------------+-----+
only showing top 20 rows
```

Retrieving the name of the columns used to make our feature vector and storing them in a pandas DataFrame:

In [9]:
```
featureCols = pd.DataFrame(preparedDF.schema["features"].metadata["ml_attr"]["attrs
  preparedDF.schema["features"].metadata["ml_attr"]["attrs"]["numeric"]).sort_value

featureCols = featureCols.set_index('idx')
featureCols.head()
```

Out[9]:

idx	name
0	gender_Male
1	SeniorCitizen
2	Partner_No
3	Dependents_No
4	tenure

Splitting the transformed data into `train` and `test` with a 30% ratio and a seed value for reproducibility:

In [10]:
```
train, test = preparedDF.randomSplit([0.7, 0.3], seed = 13)
```

Instantiating an instance of `LogisticRegression` and checking the parameters' default values:

In [11]:
```
from pyspark.ml.classification import LogisticRegression

lr = LogisticRegression()
```

In [12]:
```
lr.explainParams()
```

Out[12]:
```
"aggregationDepth: suggested depth for treeAggregate (>= 2). (default: 2)\nelastic
NetParam: the ElasticNet mixing parameter, in range [0, 1]. For alpha = 0, the pen
alty is an L2 penalty. For alpha = 1, it is an L1 penalty. (default: 0.0)\nfamily:
The name of family which is a description of the label distribution to be used in
the model. Supported options: auto, binomial, multinomial (default: auto)\nfeature
sCol: features column name. (default: features)\nfitIntercept: whether to fit an i
ntercept term. (default: True)\nlabelCol: label column name. (default: label)\nlow
erBoundsOnCoefficients: The lower bounds on coefficients if fitting under bound co
nstrained optimization. The bound matrix must be compatible with the shape (1, num
ber of features) for binomial regression, or (number of classes, number of feature
s) for multinomial regression. (undefined)\nlowerBoundsOnIntercepts: The lower bou
nds on intercepts if fitting under bound constrained optimization. The bounds vect
or size must beequal with 1 for binomial regression, or the number oflasses for mu
ltinomial regression. (undefined)\nmaxBlockSizeInMB: maximum memory in MB for stac
king input data into blocks. Data is stacked within partitions. If more than remai
ning data size in a partition then it is adjusted to the data size. Default 0.0 re
presents choosing optimal value, depends on specific algorithm. Must be >= 0. (def
ault: 0.0)\nmaxIter: max number of iterations (>= 0). (default: 100)\npredictionCo
l: prediction column name. (default: prediction)\nprobabilityCol: Column name for
predicted class conditional probabilities. Note: Not all models output well-calibr
ated probability estimates! These probabilities should be treated as confidences,
not precise probabilities. (default: probability)\nrawPredictionCol: raw predictio
n (a.k.a. confidence) column name. (default: rawPrediction)\nregParam: regularizat
ion parameter (>= 0). (default: 0.0)\nstandardization: whether to standardize the
training features before fitting the model. (default: True)\nthreshold: Threshold
in binary classification prediction, in range [0, 1]. If threshold and thresholds
are both set, they must match.e.g. if threshold is p, then thresholds must be equa
l to [1-p, p]. (default: 0.5)\nthresholds: Thresholds in multi-class classificatio
n to adjust the probability of predicting each class. Array must have length equal
to the number of classes, with values > 0, excepting that at most one value may be
0. The class with largest value p/t is predicted, where p is the original probabil
ity of that class and t is the class's threshold. (undefined)\ntol: the convergenc
e tolerance for iterative algorithms (>= 0). (default: 1e-06)\nupperBoundsOnCoeffi
cients: The upper bounds on coefficients if fitting under bound constrained optimi
zation. The bound matrix must be compatible with the shape (1, number of features)
for binomial regression, or (number of classes, number of features) for multinomia
l regression. (undefined)\nupperBoundsOnIntercepts: The upper bounds on intercepts
if fitting under bound constrained optimization. The bound vector size must be equ
al with 1 for binomial regression, or the number of classes for multinomial regres
sion. (undefined)\nweightCol: weight column name. If this is not set or empty, we
treat all instance weights as 1.0. (undefined)"
```

Fitting the model on our training data:

In [13]:
```
lrModel = lr.fit(train)
```

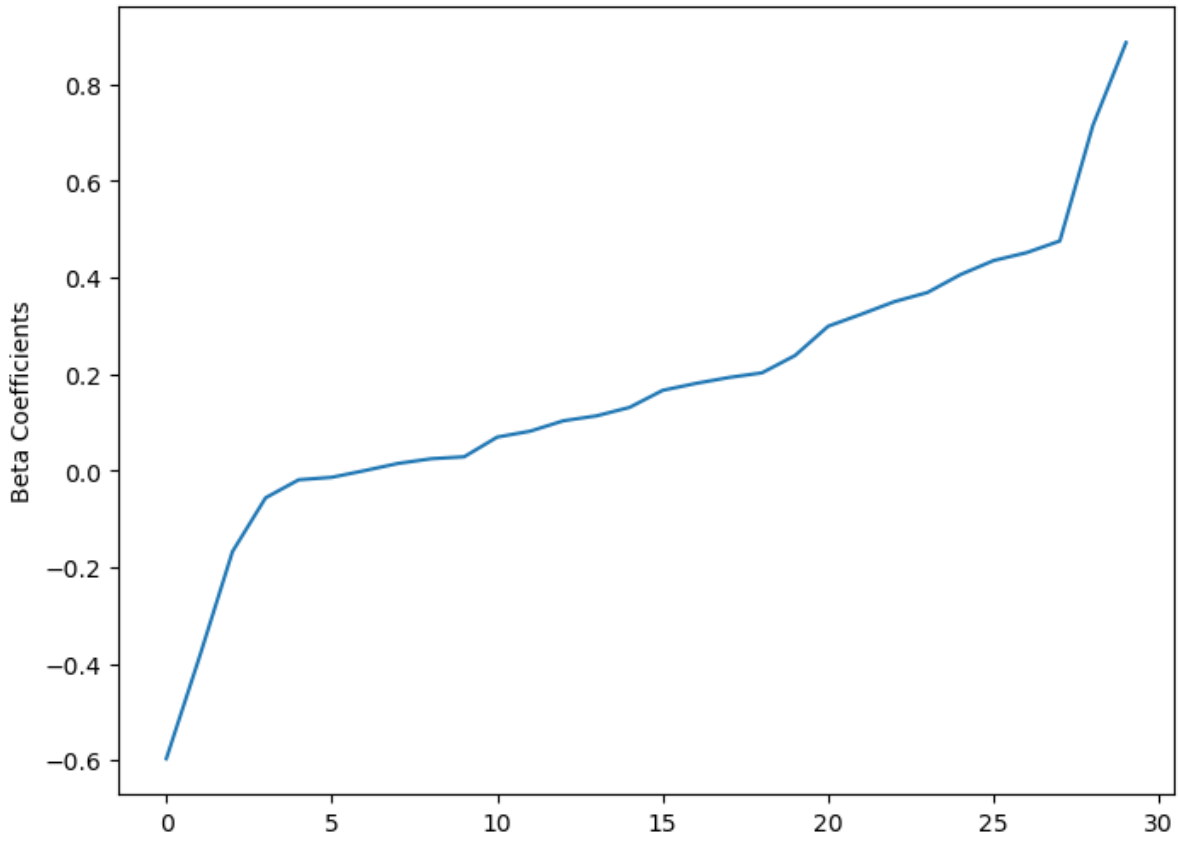
```
23/03/07 20:59:13 WARN com.github.fommil.netlib.BLAS: Failed to load implementatio
n from: com.github.fommil.netlib.NativeSystemBLAS
23/03/07 20:59:14 WARN com.github.fommil.netlib.BLAS: Failed to load implementatio
n from: com.github.fommil.netlib.NativeRefBLAS
```

Plotting the coefficients of our trained model in a sorted manner:

In [14]:
```
plt.rcParams["figure.figsize"] = (8,6)
```

In [15]:
```
beta = np.sort(lrModel.coefficients)
plt.plot(beta)
plt.ylabel('Beta Coefficients')
```

Out[15]:
```
Text(0, 0.5, 'Beta Coefficients')
```



Feature importance

After sorting the coefficients, we need to rejoin them with their names in order to identify the ones with higher absolute values. This is done below:

In [16]:
```
coefsArray = np.array(lrModel.coefficients)  # convert to np.array
coefsDF = pd.DataFrame(coefsArray, columns=['coefs'])  # to pandas

coefsDF = coefsDF.merge(featureCols, left_index=True, right_index=True)  # join it
coefsDF.sort_values('coefs', inplace=True)  # Sort them
coefsDF.head()
```

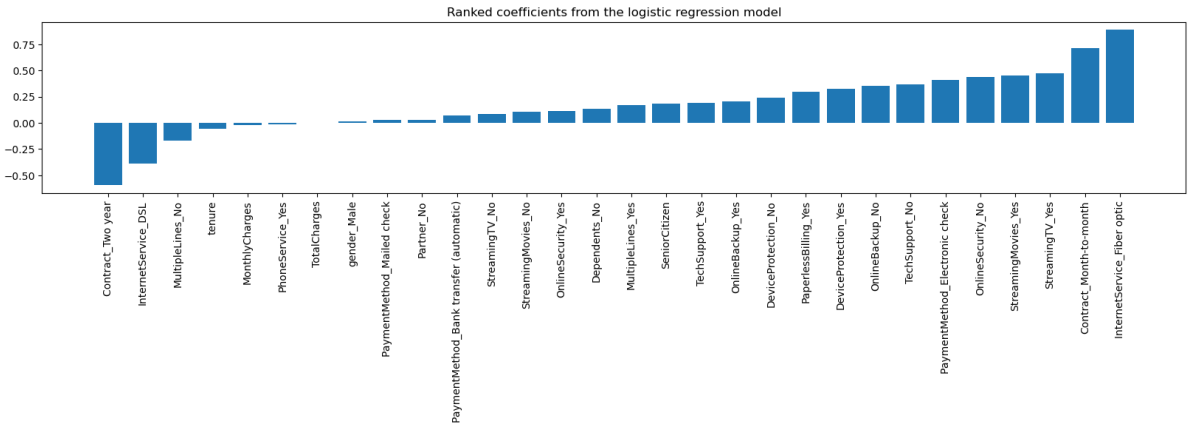
Out[16]:

	coefs	name
23	-0.596668	Contract_Two year
9	-0.386310	InternetService_DSL
6	-0.167465	MultipleLines_No
4	-0.056157	tenure
28	-0.018830	MonthlyCharges

Plotting a sorted bar chart of all coefficients:

In [17]:
```
plt.rcParams["figure.figsize"] = (20,3)
```

In [18]:
```
plt.xticks(rotation=90)
plt.bar(coefsDF.name, coefsDF.coefs)
plt.title('Ranked coefficients from the logistic regression model')
plt.show()
```



Extracting the summary from our fitted model:

In [19]:
```
summary = lrModel.summary
```

Extracting the area under curve (AUC) of our fitted model from the training data:

In [20]:
```
summary.areaUnderROC
```

Out[20]:
```
0.8472021124344006
```

Extracting ROC raw values from our summary:

In [21]:
```
roc = summary.roc.toPandas()
roc
```

Out[21]:

	FPR	TPR
0	0.000000	0.000000
1	0.000000	0.003046
2	0.000275	0.005331
3	0.000275	0.008378
4	0.000275	0.011424
...	...	...
1231	0.996703	1.000000
1232	0.997802	1.000000
1233	0.998901	1.000000
1234	1.000000	1.000000
1235	1.000000	1.000000

1236 rows × 2 columns

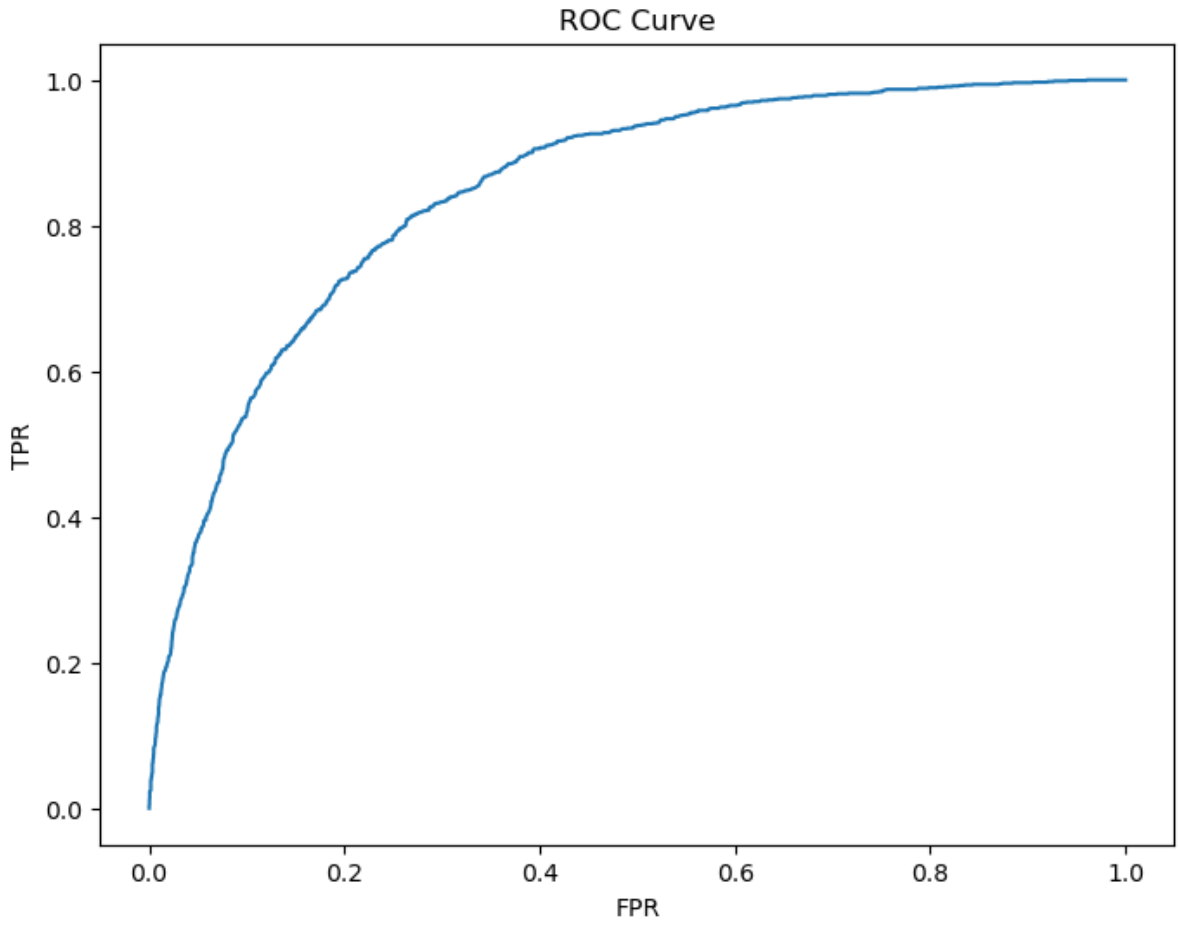
Plotting the ROC curve:

In [22]:
```
plt.rcParams["figure.figsize"] = (8,6)
```

In [23]:
```
roc.plot(x='FPR', y='TPR', style='-', legend=False)
plt.title('ROC Curve')
plt.ylabel('TPR')

print('Train AUC:', summary.areaUnderROC)
```

```
Train AUC: 0.8472021124344006
```



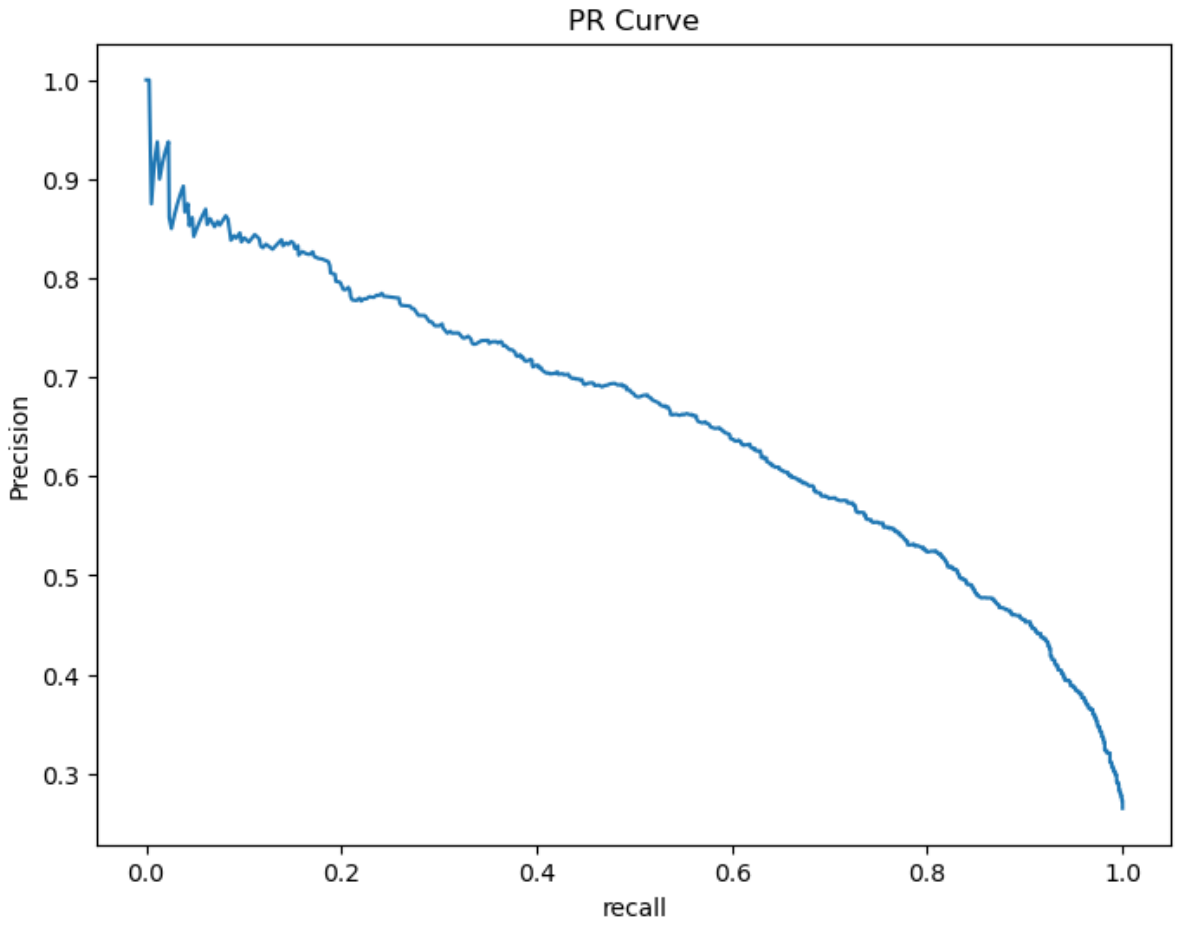
Plotting the PR curve in a similar manner:

In [24]:
```
pr = summary.pr.toPandas()

pr.plot(x='recall', y='precision', style='-', legend=False)
plt.title('PR Curve')
plt.ylabel('Precision')

print('Train AUC:', summary.areaUnderROC)
```

```
Train AUC: 0.8472021124344006
```



From the results above we can gather that the baseline model looks adequate. We can now confidently move onto predicting using the testing dataset.

In [25]:
```
fittedTest = lrModel.transform(test)
```

Printing the first few rows of our fitted model, showing 'label', 'prediction', and 'rawPrediction':

In [26]:
```
fittedTest.select('label', 'prediction', 'rawPrediction').show(5)
```

```
+-----+----------+--------------------+
|label|prediction|       rawPrediction|
+-----+----------+--------------------+
|  1.0|       1.0|[-0.3404532976936...|
|  0.0|       1.0|[-0.2525892987373...|
|  0.0|       0.0|[0.32770103173280...|
|  0.0|       0.0|[0.59633508895060...|
|  0.0|       0.0|[0.60375957562736...|
+-----+----------+--------------------+
only showing top 5 rows
```

Making an evaluator that calculates the AUC of this fitted model as a way to measure performance:

In [27]:
```
from pyspark.ml.evaluation import BinaryClassificationEvaluator

aucEvaluator = BinaryClassificationEvaluator()
```

Using our `aucEvaluator` find out the AUC on the `test` set:

In [28]:
```
aucEvaluator.evaluate(fittedTest, {aucEvaluator.metricName: "areaUnderROC"})
```

Out[28]:
```
0.8494338167015298
```

The test and train AUC's are very close in value, allowing us to conclude that this model will provide a high-performing tool to predict customer churn at Telco.