Customer Segmentation

Ashish Jha

Lakshmi Ravichandran

Problem Statement

- E-commerce customer segmentation and classification of new customer with the goal of improving retention rate
  - Exploratory data analysis
    - Cohort analysis shows customer and revenue retention
    - Product popularity distribution
  - Feature engineering
  - K-means clustering for customer segmentation
  - Classification of new customers into above clusters

Problem Statement (contd)

- Mobile Subscriber segmentation and classification of customer for active retention
- Exploratory data analysis
  - Feature engineering
  - Logistic Regression
- Classification of customer with high revenue and high churn probability as to maximize revenue

E-Commerce Data Set – UC Irvine

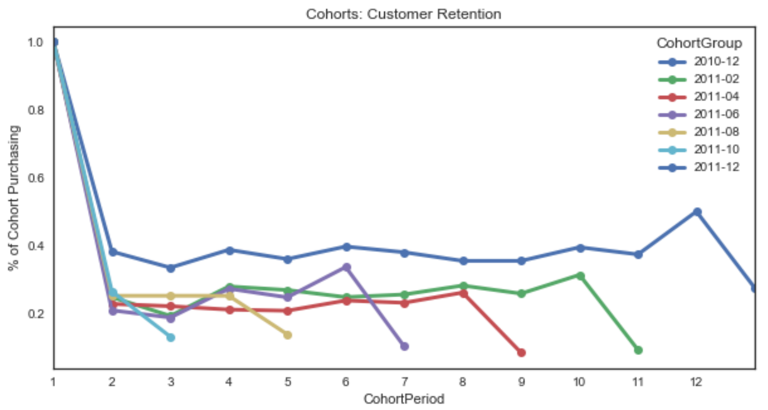
Customer transaction data for 13 months for an UK based online gift retail store

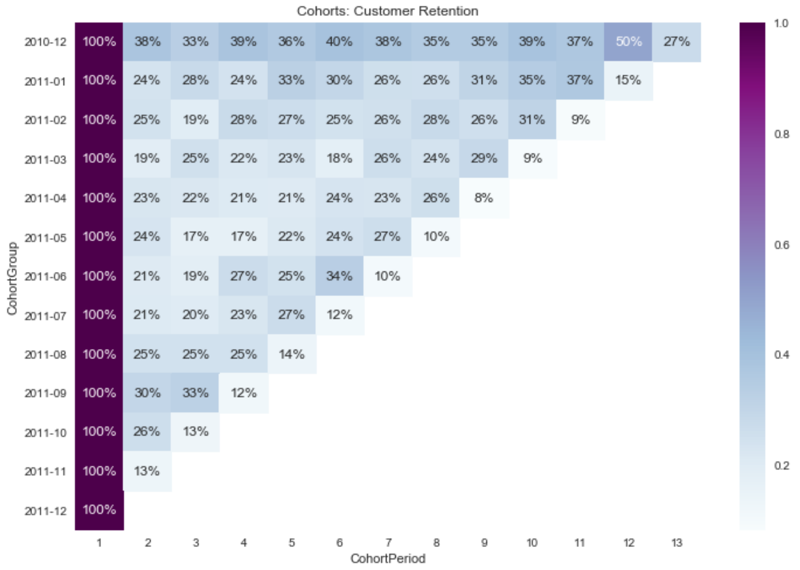
	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
5	536365	22752	SET 7 BABUSHKA NESTING BOXES	2	12/1/2010 8:26	7.65	17850.0	United Kingdom
6	536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER	6	12/1/2010 8:26	4.25	17850.0	United Kingdom
7	536366	22633	HAND WARMER UNION JACK	6	12/1/2010 8:28	1.85	17850.0	United Kingdom

Stats after data cleaning

Country	Customers	Products	Transactions
37	4372	3896	22190

Exploratory Data Analysis – Customer Retention

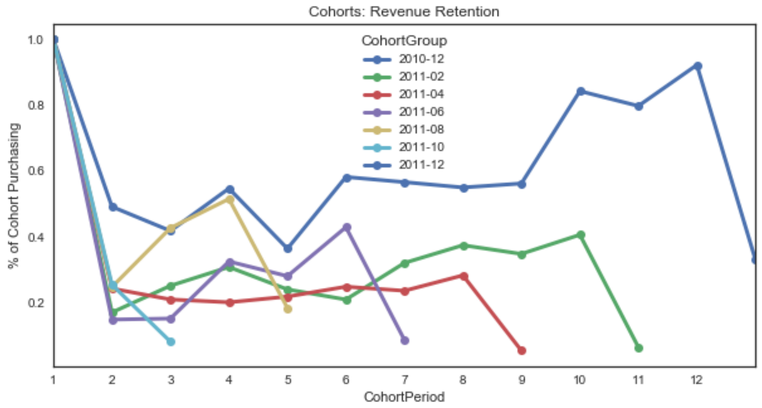


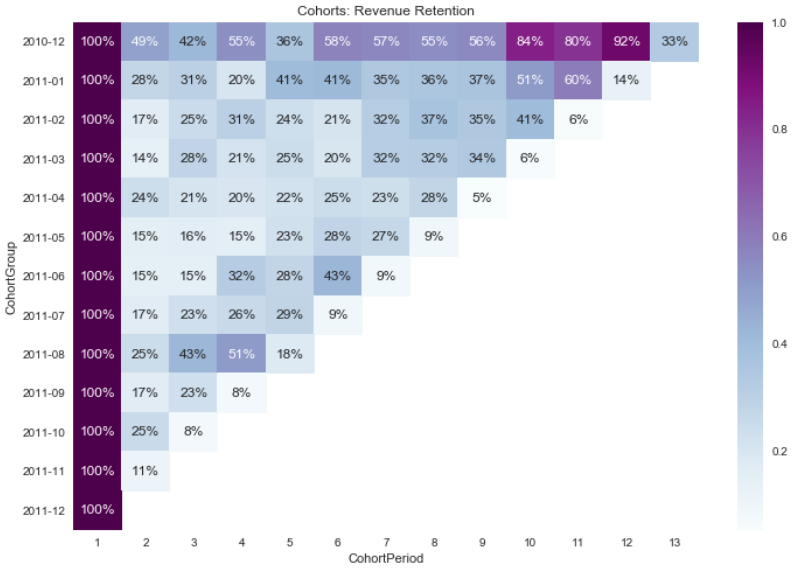


Cohort Analysis

- Group customers by first purchase month
- For each group, count unique customers from first purchase month to last month in dataset
- Normalize the count by the number of unique customers in each cohort group
- Graph shows
  - Low customer retention across all groups
  - First group higher retention than other groups

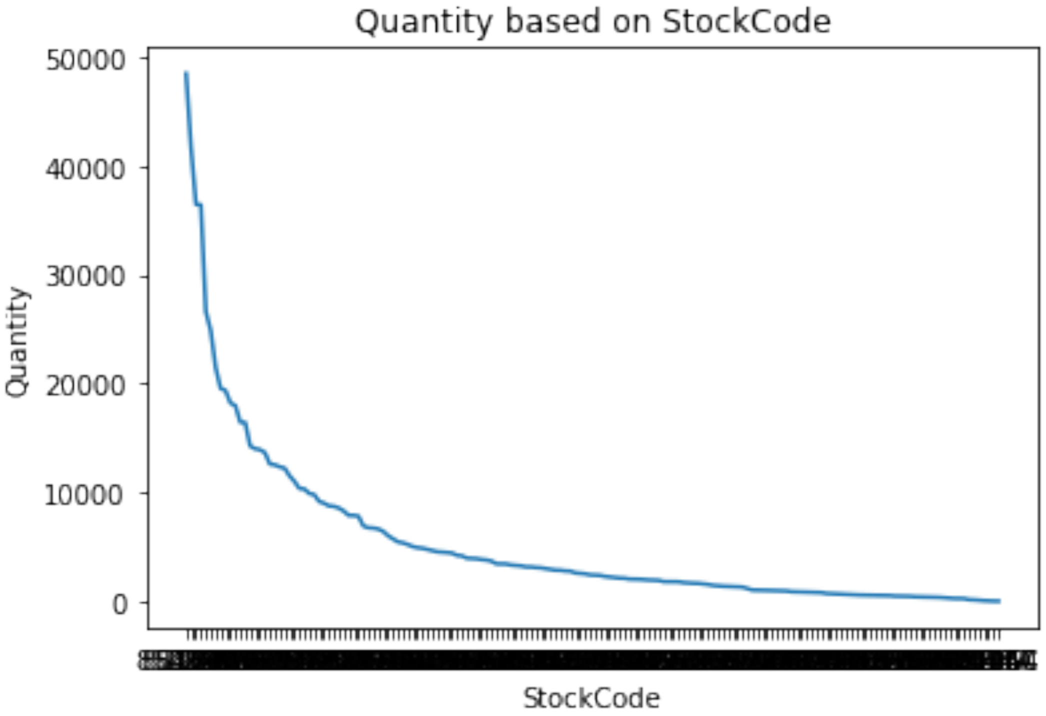
Exploratory Data Analysis – Revenue Retention





- Graph shows similar trends for revenue retention
- Exception : Seasonality in revenue (November higher sales)

Exploratory Data Analysis – Product Popularity



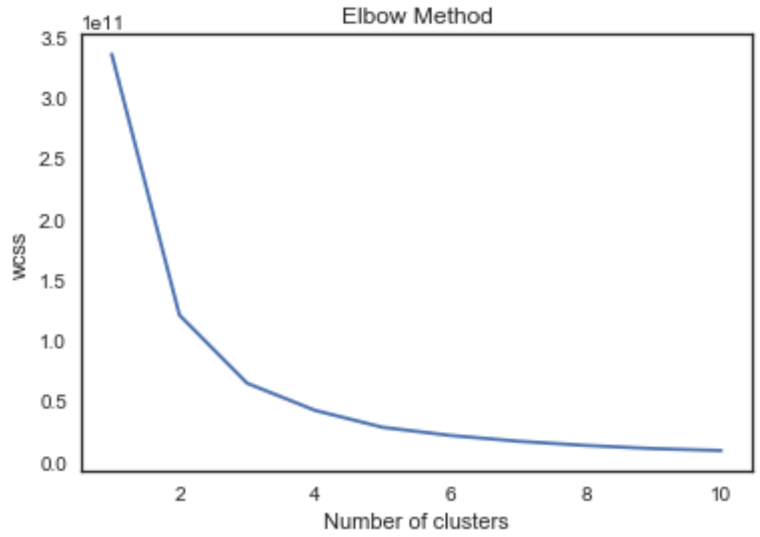
- Graph shows total quantity sold for each product code
- Trend : Zipf distribution showing some products sell in large quantities
- Implication : categorize products into 3 popularity groups used for customer segmentation

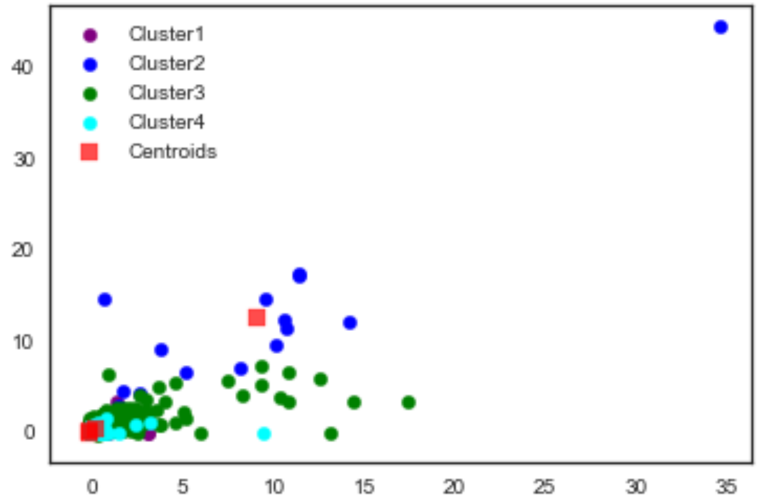
Feature Engineering

For each customer -

- Product interests
  - Quantity of products purchased classified into 3 popularity groups
- Purchasing power
  - Total amount spent
- Frequency of purchase
  - Active number of days of transaction
  - Mean transaction rate per month

CustomerID	Prod_Category1_Qnty	Prod_Category2_Qnty	Prod_Category3_Qnty	Amount_Spent	DaySinceFirstPurchase	DaySinceLastPurchase	ActiveDays	TransactionRate
12346	0	0	0	0.00	325	325	1	0.005128
12347	471	1082	905	4310.00	367	2	7	0.466667
12348	720	1477	144	1797.24	358	75	4	0.079487
12349	13	432	186	1757.55	18	18	1	0.187179
12350	12	73	112	334.40	310	310	1	0.043590
12352	47	219	204	1545.41	296	36	7	0.243590
12353	0	0	20	89.00	204	204	1	0.010256
12354	78	323	129	1079.40	232	232	1	0.148718
12355	26	174	40	459.40	214	214	1	0.033333
12356	373	461	757	2811.43	325	22	3	0.151282



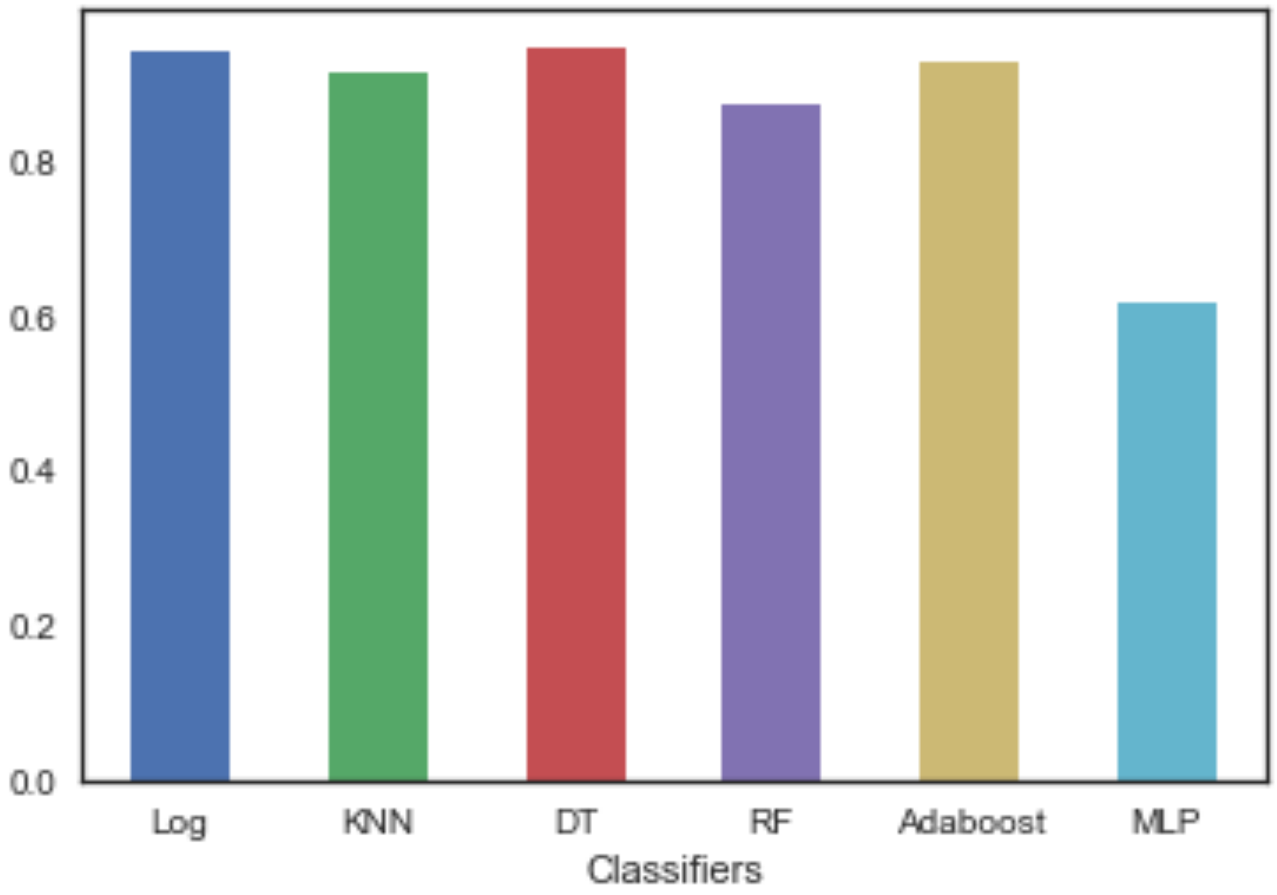


Customer Segmentation using K-means Clustering

- Elbow plot - find the best number of clusters for data
- Our data n=4 is the knee of the curve beyond which no additional benefits
- Scatter plot showing K-means with n=4
- Num customers in each cluster – 1687, 1604, 1066, 15
  - **Used as label for prediction**

Classification of new customers

- Data labeled using k-means clustering
- Split into training and testing
- Multiple classification algorithms tried
- Overall over 90% accuracy for all algorithms except MLP
- Currently tuning parameters and understanding the prediction results further



The mobile Subscriber data (as we got it)

Data Details

- #Features:81
- #Observation:62297
- #Features with NA:45
- Target Feature: ‘Churn’
  - Did not churn: 50438
  - Churn out: 15859
- Unique Identifier: ‘Customer_id’

```
Console    Terminal ×
C:/Users/ISHUHOME/PythonMLClass/FinalProj/RAnalysis
> nrow(dat)
[1] 66297
> ncol(dat)
[1] 81
> naCol = colSums(is.na(dat)) > 0
> table(naCol)
naCol
FALSE  TRUE
   36    45
>
```

```
> table(dat2$churn)

    0     1
50438 15859
>
```

Data preparation (Everyone like clean data but hates cleaning it! – Old Legend)

```
> #Removing Row with NA more than 10%
> NaRemover = colSums(is.na(dat))/nrow(dat) < 0.1
> table(NaRemover)
NaRemover
FALSE  TRUE
   14    67
> ncol(dat2)
[1] 67
>
```

Dealing with Enemy #1: NA values

Removal

- Removing Columns with percentage of NA values is more than 10%
- After removal of 14 features, we have 67 features for the analysis

Imputing missing values

Continuous Case(Simple scenario):

Simple Strategy: Mean Imputation

- Replacing NA values with average value of the feature

```
colNames_withNA = names(dat2[colSums(is.na(dat2))>0])

for (nam in colNames_withNA) {

  if (class(dat2[[nam]]) =='factor'||class(dat2[[nam]]) =='character') {

  }else{
    col_avg = mean(dat2[[nam]],na.rm = T)
    print(col_avg)
    dat2[is.na(dat2[[nam]]),nam] = col_avg
  }

}
```

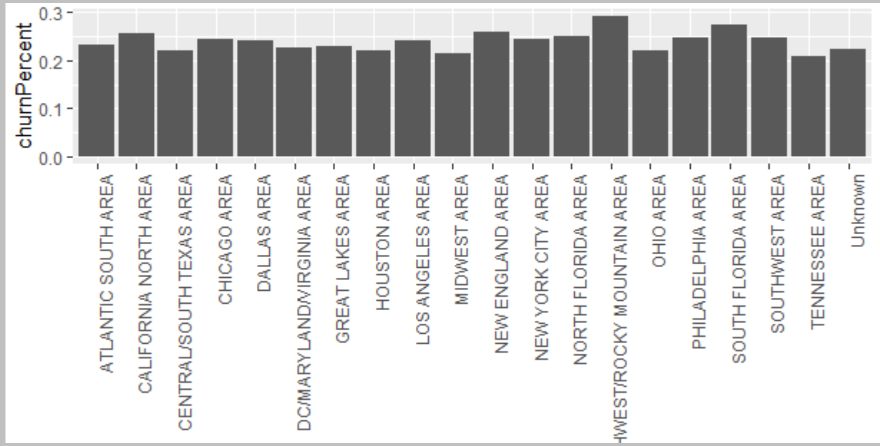
Categorical Case(This was fun to do!):

Replacing NA with factor which have similar level of target percentage (churn percentage).

Sum of churn in level/count in level

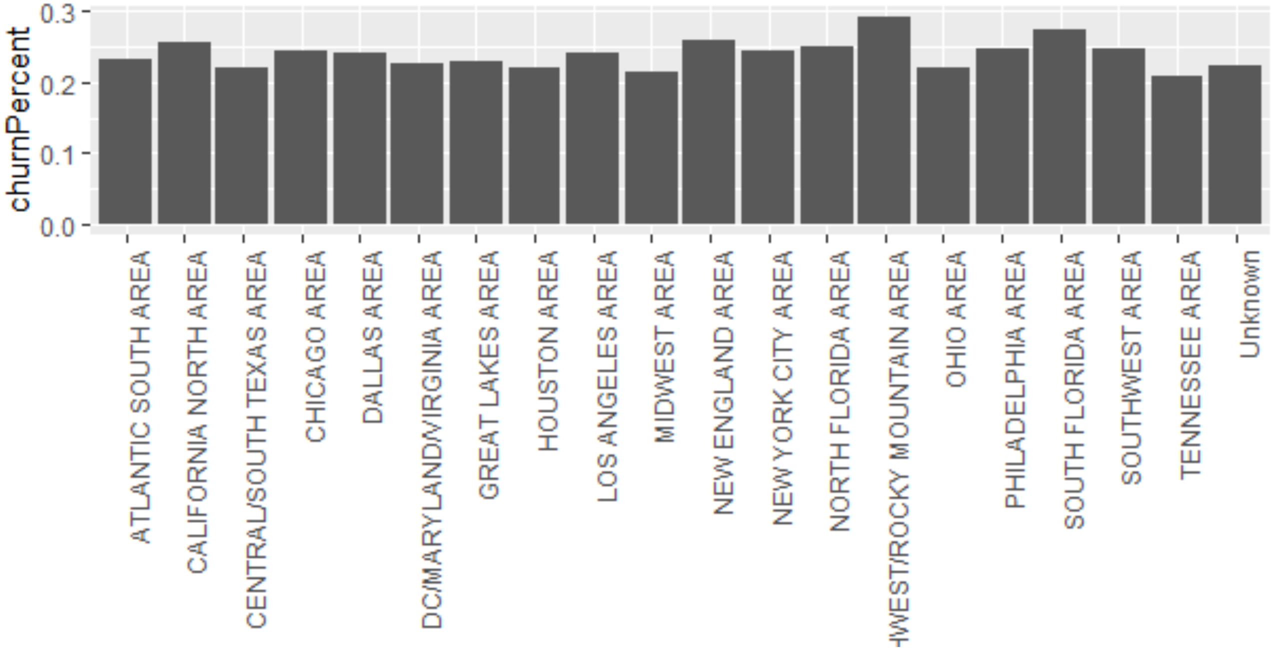
Sample with categorical Imputation(Area)

- Replacing NA levels values with Label for plotting.
- Since NA bar has churn percentage which is like factor Ohio Area.
- Later we will merge all level with similar levels




```
> dat2['area'] = fct_explicit_na(dat2[['area']],na_level = "Unknown")
> Impact <- dat2[c('area','churn')] %>% group_by(area) %>% summarise(churnPercent = sum(chur
n)/n())
> ggplot(Impact,aes(x = area,y = churnPercent)) + geom_bar(stat = 'identity')
> ggplot(Impact,aes(x = area,y = churnPercent)) + geom_bar(stat = 'identity')+ theme(axis.te
xt.x = element_text(angle = 90, hjust = 1))
> |
```

Sample with categorical Imputation(Area)

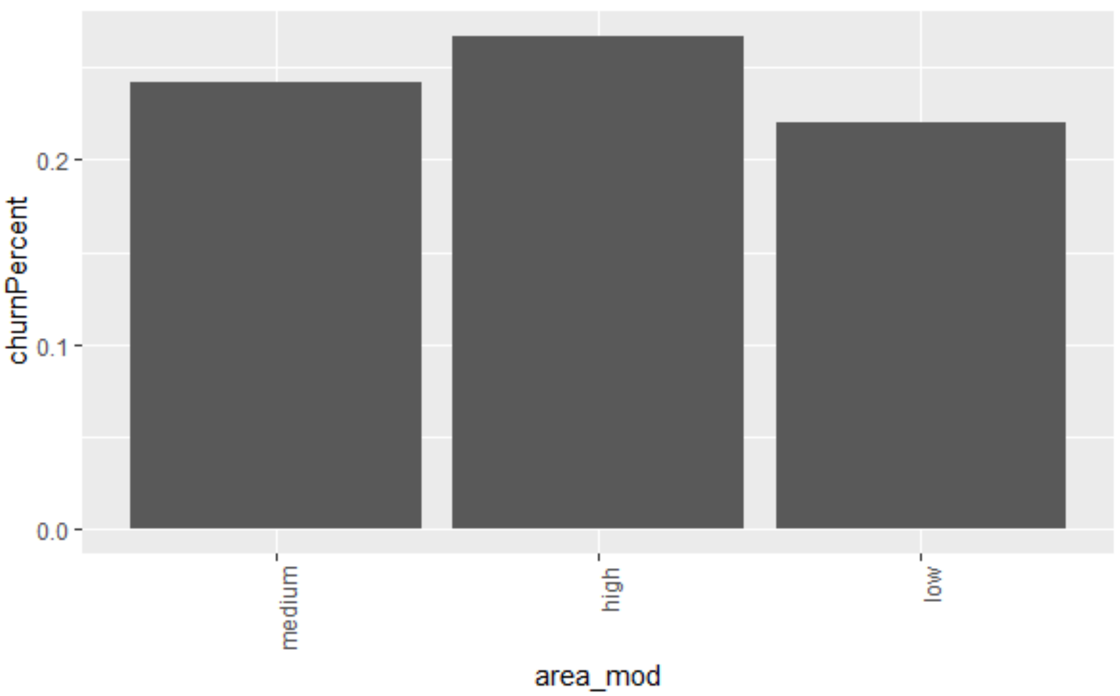


- Replacing NA levels values with Label for plotting.
- Since NA bar has churn percentage which is like factor Ohio Area.
- Later we will merge all level with similar levels

Reducing levels for all the categorical feature

SEVERAL CATEGORICAL IN THE DATASET HAVE MORE THAN 4 LEVELS.

MERGING SEVERAL FEATURE INTO FACTORS WITH LEVEL OF 3 OR 4 HAVING SIMILAR CHURN PERCENTAGE.




```
> dat2$area_mod <- dat2$area %>% fct_collapse(
+   high = c("NORTHWEST/ROCKY MOUNTAIN AREA",    "SOUTH FLORIDA AREA",    "NEW ENGLAND AREA"
,
+            "CALIFORNIA NORTH AREA",    "NORTH FLORIDA AREA"),
+   medium = c("NORTH FLORIDA AREA",     "PHILADELPHIA AREA",     "SOUTHWEST AREA",
+              "NEW YORK CITY AREA",     "CHICAGO AREA", "LOS ANGELES AREA",
+              "DALLAS AREA",    "ATLANTIC SOUTH AREA"),
+   low = c("GREAT LAKES AREA", "DC/MARYLAND/VIRGINIA AREA",     "Unknown",      "OHIO AREA
",
+           "HOUSTON AREA",     "CENTRAL/SOUTH TEXAS AREA",     "MIDWEST AREA", "TENNESSEE
 AREA")
+ )
> Impact <- dat2[c('area_mod','churn')] %>% group_by(area_mod) %>%
+   summarise(churnPercent = sum(churn)/n())
> ggplot(Impact,aes(x = area_mod,y = churnPercent)) +
+   geom_bar(stat = 'identity')+ theme(axis.text.x = element_text(angle = 90, hjust = 1))
> |
```

Engineering Area(contd.)

Now it is time for One Hot Dummy Encoding

- Now we have reduced the number of levels in the categorical variables.
- But our algorithm like to keep it simple (Ones and Zeros) Dummy!!!!
- All the factors level will have the column of their own with ones and zeros

	asl_flag	hnd_webcap	area_mod	crclscod_mod	models_mod	uniqsubs_mod	ethnic_mod	age1_mod
1	N	WCMB	medium	b	high	low	medium	age30_54
2	N	WCMB	medium	a	high	low	low	age54_70
3	Y	WCMB	low	b	high	low	medium	age30_54
4	N	WCMB	medium	a	high	low	medium	age54_70
5	N	WCMB	low	b	high	low	medium	age30_54
6	N	WCMB	low	b	high	low	low	youth
7	N	WCMB	medium	b	high	low	medium	age30_54
8	N	WCMB	high	b	high	low	medium	age30_54
9	N	WCMB	medium	b	high	low	medium	age30_54
10	N	UNKNOWN	low	a	high	low	medium	age30_54
11	N	WCMB	high	b	high	low	medium	age30_54
12	N	WCMB	low	b	high	low	medium	age30_54
13	N	WCMB	medium	a	high	low	high	age30_54

Showing 1 to 14 of 66,297 entries

53	asl_flag_Y
54	hnd_webcap_WCMB
55	hnd_webcap_UNKNOWN
56	hnd_webcap_WC
57	hnd_webcap_UNKW
58	area_mod_medium
59	area_mod_low
60	area_mod_high
61	crclscod_mod_b
62	crclscod_mod_a
63	crclscod_mod_c
64	models_mod_high
65	models_mod_med
66	models_mod_low
67	uniqsubs_mod_low
68	uniqsubs_mod_medium
69	uniqsubs_mod_high
70	ethnic_mod_medium
71	ethnic_mod_low
72	ethnic_mod_high
73	age1_mod_age30_54
74	age1_mod_age54_70
75	age1_mod_youth
76	age1_mod_late20
77	age1_mod_age70more

After dummy encoding

- All the levels in factor variable are nicely encoded.
- Combination of feature and feature level is a unique feature in the data to be used by Algorithms

BF	BG	BH	BI	BJ	BK	BL	BM	BN	BO	BP	BQ	BR	BS	BT	BU	BV	BW	BX	BY	BZ
area_mod	area_mod	area_mod	crclscod_r	crclscod_r	crclscod_r	models_m	models_m	models_m	uniqsubs_	uniqsubs_	uniqsubs_	ethnic_mo	ethnic_mo	ethnic_mo	age1_moc	age1_moc	age1_moc	age1_moc	age1_mod_age7	
1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	
1	0	0	0	1	0	1	0	0	1	0	0	0	1	0	0	1	0	0	0	
0	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	
1	0	0	0	1	0	1	0	0	1	0	0	1	0	0	0	1	0	0	0	
0	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	
0	1	0	1	0	0	1	0	0	1	0	0	0	1	0	0	0	1	0	0	
1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	

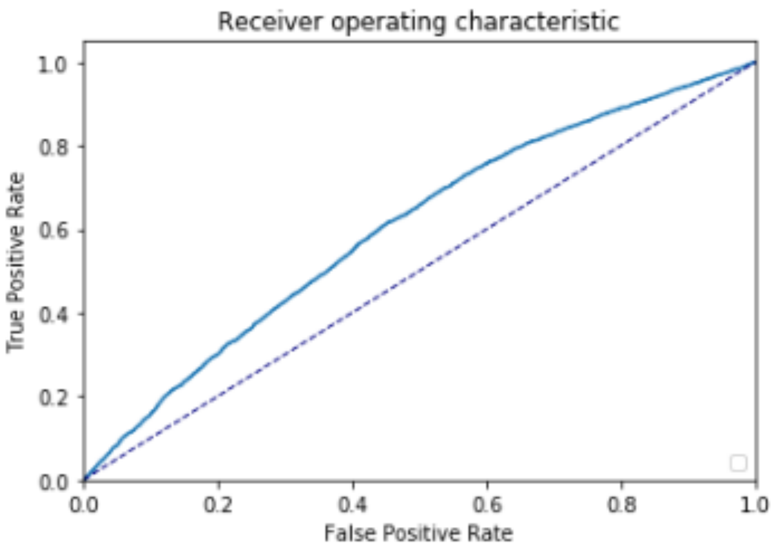
Machine learning (finally!)

- Writing clean data to csv
- Now importing data into python
- Splitting the data in train test split
- Running Logistic Regression on the train data
- Checking accuracy on the test set

```
In [8]: logReg = LogisticRegression()
        mod = logReg.fit(X_train,Y_train)
        #mod.score(X_test,Y_test)
```

```
In [7]: print("Accuracy on train data - ", mod.score(X_train,Y_train))
        print("Accuracy on test data - ", mod.score(X_test,Y_test))

        Accuracy on train data -  0.7621173725916094
        Accuracy on test data -  0.7568627450980392
```




```
In [24]: # calculate AUC
         auc = roc_auc_score(Y_test, test_probs[:,1])
         print('AUC: %.3f' % auc)

AUC: 0.599
```

Model details

- Accuracy on train set: 0.762
- Accuracy on test set: 0.756
- AUC = 0.599 (almost 0.6, welcome to real world)

Using churn probability to target customer proactively for retention

- We have probability score if the customer is going to change the subscriber.
- In the real world with limited resources we may not be able to target all the customer for retention.
- Thus we segment customers into group with high revenue and high probability to churn
- Thus model can be used to predict customers with high probability of Churn and extract the target list using their "Customer ID".

Probability of Churn (Score)/Revenue	Low (Y1-Y2)	Medium (Y2-Y3)	High (Y3-Y4)
Low (X1-X2)			
Medium (X2-X3)			Target
High(X3-X4)		Target	Target

Conclusions

Lesson : As always 80-20 rule triumphed in the end

80% of time spent in finding dataset, cleaning data, feature engineering

Engineering the right features with domain knowledge critical

Common problems in two different domains (customer segmentation, churn prediction for E-commerce and Telecom domains)