UE19CS334 - NLP
Assignment 1

Team Members:

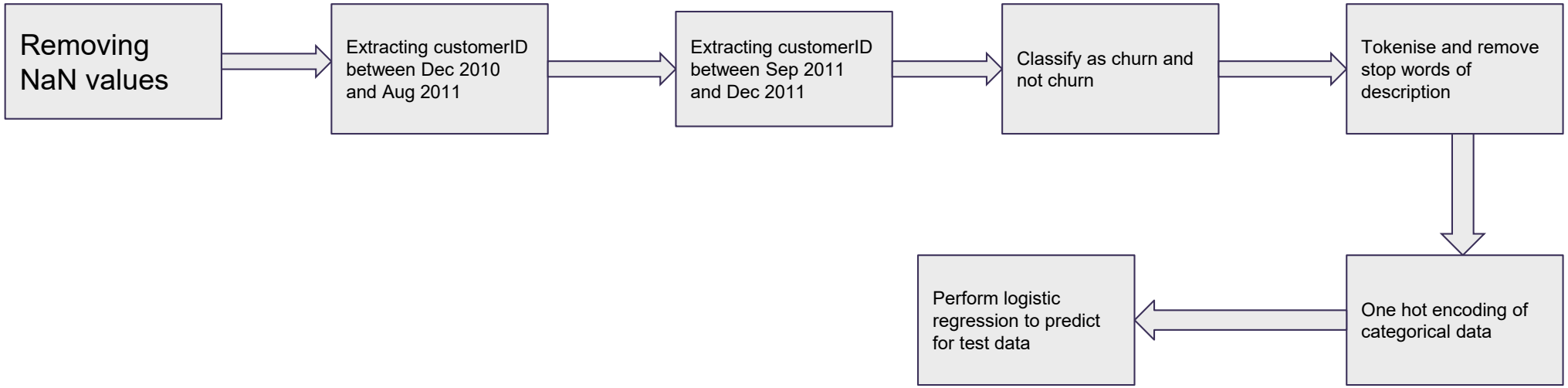
NAME	SRN	SECTION
B PRAVENA	PES2UG19CS076	B
VARNA SATYANARAYANA	PES2UG19CS448	G
SPOORTHI R	PES1UG19CS500	E

Abstract

A company has been experiencing high customer churn and a reduction in repeat customers. As part of this case study, we have built a predictive model to predict the likelihood of a customer churning.

We have performed Target variable creation (based on churn definition provided), Customer level feature creation from transaction dataset, EDA on the features thus created a Logistic Regression model and performed Model training, validation and evaluation.

Flowchart


Removing NaN values
Extracting customerID between Dec 2010 and Aug 2011
Extracting customerID between Sep 2011 and Dec 2011
Classify as churn and not churn
Tokenise and remove stop words of description
One hot encoding of categorical data
Perform logistic regression to predict for test data

Packages and Libraries used

We used -:

- Pandas
- Numpy
- NLTK - stopwords, word_tokenize
- seaborn
- matplotlib.pyplot
- sklearn - model_selection, linear_model

Output Screenshots

NLPAssignment.ipynb ☆

File Edit View Insert Runtime Tools Help Last saved at 18:37

+ Code + Text

```
[ ] import pandas as pd
    import numpy as np
```

```
[ ] df=pd.read_excel('NLP_Assignment1_Online Retail.xlsx')
```

```
df.isna().sum()
```

```
InvoiceNo          0
StockCode          0
Description     1454
Quantity           0
InvoiceDate        0
UnitPrice          0
CustomerID    135080
Country            0
dtype: int64
```

```
p.describe()
```

	Quantity	UnitPrice	CustomerID
count	541909.000000	541909.000000	406829.000000
mean	9.552250	4.611114	15287.690570
std	218.081158	96.759853	1713.600303
min	-80995.000000	-11062.060000	12346.000000
25%	1.000000	1.250000	13953.000000
50%	3.000000	2.080000	15152.000000
75%	10.000000	4.130000	16791.000000
max	80995.000000	38970.000000	18287.000000

```
p.dtypes
```

```
InvoiceNo              object
StockCode              object
Description            object
Quantity                int64
InvoiceDate    datetime64[ns]
UnitPrice             float64
CustomerID            float64
Country                object
dtype: object
```

```
[278] p.describe()
```

	Quantity	UnitPrice	CustomerID
count	541909.000000	541909.000000	406829.000000
mean	9.552250	4.611114	15287.690570
std	218.081158	96.759853	1713.600303
min	-80995.000000	-11062.060000	12346.000000
25%	1.000000	1.250000	13953.000000
50%	3.000000	2.080000	15152.000000
75%	10.000000	4.130000	16791.000000
max	80995.000000	38970.000000	18287.000000

```
[280] p.dtypes
```

```
InvoiceNo              object
StockCode              object
Description            object
Quantity                int64
InvoiceDate    datetime64[ns]
```

```
p.dtypes
```

```
InvoiceNo              object
StockCode              object
Description            object
Quantity                int64
InvoiceDate    datetime64[ns]
UnitPrice             float64
CustomerID            float64
Country                object
dtype: object
```

```
p.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541909 entries, 0 to 541908
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    541909 non-null  object
 1   StockCode    541909 non-null  object
 2   Description  540455 non-null  object
 3   Quantity     541909 non-null  int64
 4   InvoiceDate  541909 non-null  datetime64[ns]
 5   UnitPrice    541909 non-null  float64
 6   CustomerID   406829 non-null  float64
 7   Country      541909 non-null  object
dtypes: datetime64[ns](1), float64(2), int64(1), object(4)
memory usage: 33.1+ MB
```

```
[284] p.isna().any()
```

```
InvoiceNo      False
StockCode      False
Description     True
Quantity       False
InvoiceDate    False
UnitPrice      False
CustomerID      True
Country        False
dtype: bool
```

```
[200] p.groupby('CustomerID')['InvoiceNo'].unique()
```

```
CustomerID
12346.0                                        [541431, C541433]
12347.0    [537626, 542237, 549222, 556201, 562032, 57351...
12348.0                     [539318, 541998, 548955, 568172]
12349.0                                              [577609]
12350.0                                              [543037]
                                  ...
18280.0                                              [545712]
18281.0                                              [556464]
18282.0                               [562525, C562808, 580173]
18283.0    [540350, 541854, 545079, 550957, 554157, 55673...
18287.0                                [554065, 570715, 573167]
Name: InvoiceNo, Length: 4372, dtype: object
```

NLPAssignment.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment

+ Code + Text

RAM Disk

[205] df.head()

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	Target	TokensDesc	isChurn
0	536365	85123A	white hanging heart t-light holder	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom	Churn	white hanging heart t-light holder	Churn
1	536365	71053	white metal lantern	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	Churn	white metal lantern	Churn
2	536365	84406B	cream cupid hearts coat hanger	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom	Churn	cream cupid hearts coat hanger	Churn
3	536365	84029G	knitted union flag hot water bottle	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	Churn	knitted union flag hot water bottle	Churn
4	536365	84029E	red woolly hottie white heart.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	Churn	red woolly hottie white heart.	Churn

```
df.isna().sum()
```

```
InvoiceNo      0
StockCode      0
Description    0
Quantity       0
InvoiceDate    0
UnitPrice      0
CustomerID     0
Country        0
Target         0
TokensDesc     0
isChurn        0
dtype: int64
```

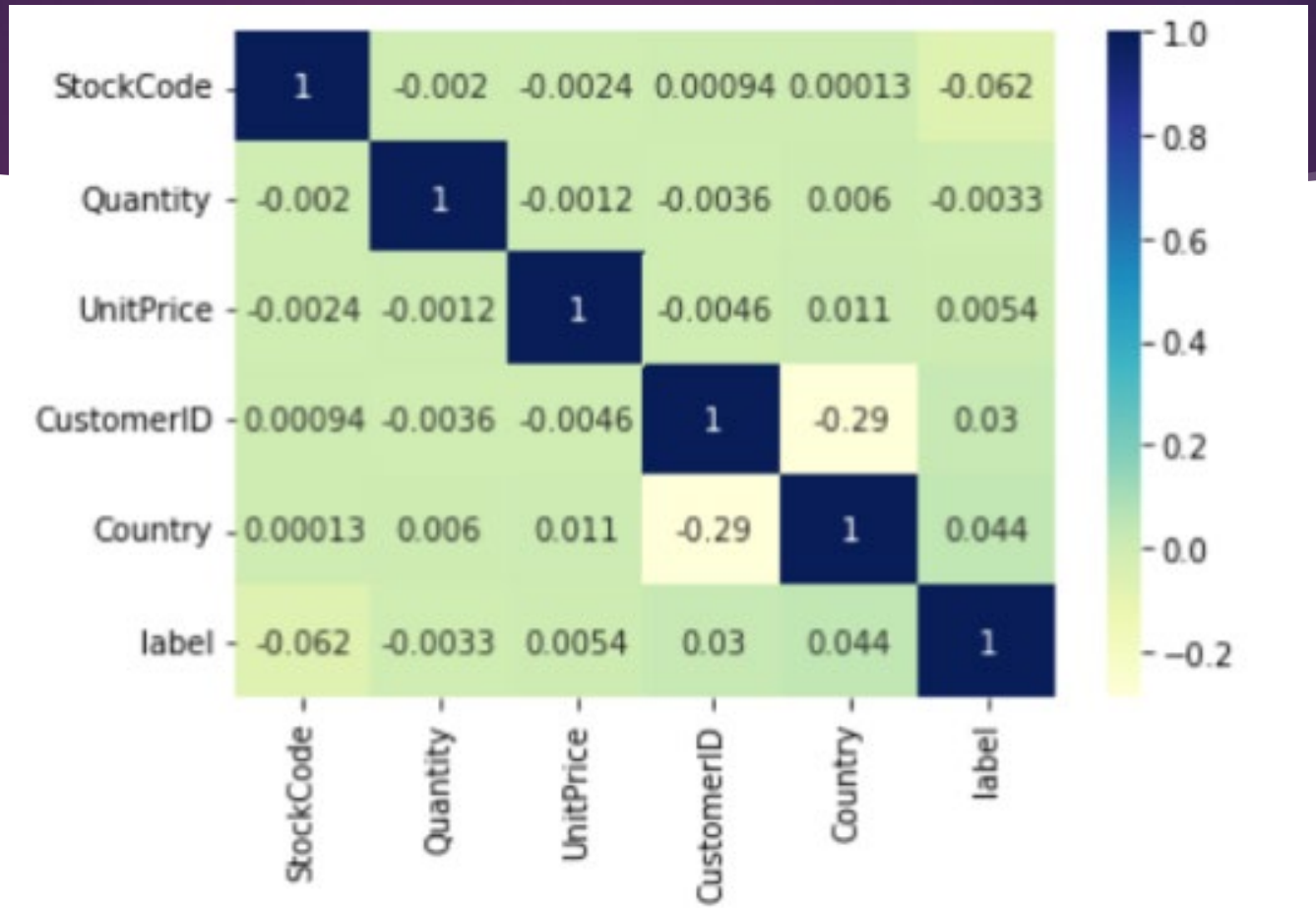
```
df[df['Target']=='Churn'].count()
```

```
InvoiceNo      50510
StockCode      50510
Description    50510
Quantity       50510
InvoiceDate    50510
UnitPrice      50510
CustomerID     50510
Country        50510
Target         50510
TokensDesc     50510
label          50510
dtype: int64
```

```
[289] df[df['Target']=='Not Churn'].count()
```

```
InvoiceNo      356319
StockCode      356319
Description    356319
Quantity       356319
InvoiceDate    356319
UnitPrice      356319
CustomerID     356319
Country        356319
Target         356319
TokensDesc     356319
label          356319
dtype: int64
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	Target	TokensDesc
0	536365	85123A	white hanging heart t-light holder	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom	Churn	[white, hanging, heart, t-light, holder]
1	536365	71053	white metal lantern	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	Churn	[white, metal, lantern]
2	536365	84406B	cream cupid hearts coat hanger	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom	Churn	[cream, cupid, hearts, coat, hanger]
3	536365	84029G	knitted union flag hot water bottle	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	Churn	[knitted, union, flag, hot, water, bottle]
4	536365	84029E	red woolly hottie white heart.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	Churn	[red, woolly, hottie, white, heart, .]


StockCode
Quantity
UnitPrice
CustomerID
Country
label
1
-0.002
-0.0024
0.00094
0.00013
-0.062
-0.002
1
-0.0012
-0.0036
0.006
-0.0033
-0.0024
-0.0012
1
-0.0046
0.011
0.0054
0.00094
-0.0036
-0.0046
1
-0.29
0.03
0.00013
0.006
0.011
-0.29
1
0.044
-0.062
-0.0033
0.0054
0.03
0.044
1
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

```
logisticRegr.fit(x_train, y_train)
```

```
LogisticRegression()
```

```
# Returns a NumPy Array
# Predict for One Observation (image)
logisticRegr.predict(x_test)
```

```
array([0, 0, 0, ..., 0, 0, 0])
```

```
score = logisticRegr.score(x_test, y_test)
print(score)
```

```
0.8759291304518818
```

Thank
You