MATH2319 Machine Learning Project Phase 1
Predicting A Binary Label

Reshma Taruni Gadala

April 28, 2019

Contents

Chapter 1

Intro to churn in banking data

The data is sourced from Kaggle [1].

Introduction:

One of the most important metrics in growing business today is customer churn. It's the hard truth that a company faces about customer retention. Customer churn is defined as the percentage of customers, who have stopped using a company's product in a given time frame. This has been one of the hardest hurdles that a financial institution faces.

Objectives:

The main of objective of this machine learning project is to get insights about the customer based on the data available with good percentage of accuracy. This data set contains details of a bank's customers and the target variable is a binary variable reflecting the fact whether the customer left the bank (closed his account) or he continues to be a customer.

Data Legend

RowNumber Row Numbers from 1 to 10000, **CustomerId** Unique Ids for bank customer identification, **Surname** Customer's last name, **CreditScore** Credit score of the customer, **Geography** The country from which the customer belongs, **Gender** Male or Female, **Age** Age of the customer, **Tenure** Number of years for which the customer has been with the bank, **Balance** Bank balance of the customer, **NumOfProducts** Number of bank products the customer is utilizing, **HasCrCard** Binary Flag for whether the customer holds a credit card with the bank or not, **IsActiveMember** Binary Flag for whether the customer is an active member with the bank or not, **EstimatedSalary** Estimated salary of the customer in Dollars, **Exited** Binary flag 1 if the customer closed account with bank and 0 if the customer is retained.

Chapter 2

Descriptive Statistics and processing

We import all the required libraries in this section. And we read the CSV file required to do this task using read_csv fuction available in python. We have set 'sep' attribute to ',' to get the comma separted data in grid layout or data frame. Later we show onlt the first five rows from the data set so show that we have loaded the data set appropriately.

In [8]:
```
import pandas as pd #importing pandas library
import seaborn as sns
import matplotlib.pyplot as plt
churn_data = pd.read_csv("Churn_Modelling.csv",sep=',',decimal='.')
churn_data.head()
```

Out[8]:
```
   RowNumber  CustomerId   Surname  CreditScore Geography  Gender  Age  \
0          1    15634602  Hargrave          619    France  Female   42
1          2    15647311      Hill          608     Spain  Female   41
2          3    15619304      Onio          502    France  Female   42
3          4    15701354      Boni          699    France  Female   39
4          5    15737888  Mitchell          850     Spain  Female   43

   Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember\  0
        2       0.00              1          1               1
1       1   83807.86              1          0               1
2       8  159660.80              3          1               0
3       1       0.00              2          0               0
4       2  125510.82              1          1               1

   EstimatedSalary  Exited
0        101348.88       1
1        112542.58       0
2        113931.57       1
3         93826.63       0
4         79084.10       0
```

In [9]:
```
#To find the no=umber of rows and columns we use shape function
churn_data.shape
```

Out[9]: (10000, 14)

After executing this statement it is observed that the dataset has no 'Nan' values because even after executing dropna function we haven't lost any number of rows or columns. Previously : (10000, 14) After dropna : 10000 rows Œ 14 columns.

The below statement is used to check if there are any null values in any of the columns. The output suggests that there are no null values.

In [10]:
```
churn_data.isnull().any()
```

Out[10]:
```
RowNumber          False
CustomerId         False
Surname            False
CreditScore        False
Geography          False
Gender             False
Age                False
Tenure             False
Balance            False
NumOfProducts      False
HasCrCard          False
IsActiveMember     False
EstimatedSalary    False
Exited             False
dtype: bool
```

In [11]:
```
# Checking basic details of our dataset that is all
# the data type of columns, missing values etc.
churn_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 14 columns):
RowNumber          10000  non-null  int64
CustomerId         10000  non-null  int64
Surname            10000 non-null object
CreditScore        10000 non-null  int64
Geography          10000  non-null  object
Gender             10000  non-null  object
Age                10000  non-null  int64
Tenure             10000  non-null  int64
Balance            10000 non-null float64
NumOfProducts      10000 non-null int64
HasCrCard          10000 non-null int64
IsActiveMember     10000 non-null int64
EstimatedSalary    10000 non-null float64
Exited             10000 non-null int64
dtypes:  float64(2),  int64(9),  object(3)
memory usage: 1.1+ MB
```

Data Preparation: **Non numerical data:** To check whether all the object type attributes in the dataset have appropriate categories.

In [12]:
```
#Since all the categorical varibles are the type
#object hence we try to find all the varibles with object type and
#to find the number of factors in it.
churn_data_categorical = churn_data.select_dtypes
                         (include=['object']).copy() # Getting only categorical variable
for column in churn_data_categorical:
```

```
print("The Column '{columnName}' has
        '{numOfCategories}' categories".format(columnName = column,
         numOfCategories=len(churn_data_categorical[column].unique())))
```

```
The Column 'Surname' has '2932' categories
The Column 'Geography' has '3' categories
The Column 'Gender' has '2' categories
```

From our description above in section 1, we know that there are only two categories for Gender i.e Male and Female and we have only 3 categories for Geography i.e Germany, Spain and France. All the categorical varibles have no outliers. Hence we can say the data has only legit values. **Numerical data:** We are removing all the data which are not required for visualization. However we can identify some features are objects data (non numerical data) and we will need to delete them or encode them into numbers. RowNumber, Surname, CustomerId contain personal informations which is not required, thus they should also be removed from the dataset.

In [13]:
```
# Now removing Surname, RowNumber, CustomerId because it cannot
# be considered as a categorical variable
churn_data=churn_data.drop(["Surname","RowNumber","CustomerId"], axis=1)
```

To analyze the statistical distributions i.e, count, mean, standard deviation, median etc we use describe function. We can aslo check whether our numerical datatypes are within believable range. For example if the age cloumn had 200 as their maximun age we can say that the dataset has to be further cleaned to find outliers.

In [14]:
```
churn_data.describe()
```

Out[14]:

	CreditScore	Age	Tenure	Balance	NumOfProducts \
count	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000
mean	650.528800	38.921800	5.012800	76485.889288	1.530200
std	96.653299	10.487806	2.892174	62397.405202	0.581654
min	350.000000	18.000000	0.000000	0.000000	1.000000
25%	584.000000	32.000000	3.000000	0.000000	1.000000
50%	652.000000	37.000000	5.000000	97198.540000	1.000000
75%	718.000000	44.000000	7.000000	127644.240000	2.000000
max	850.000000	92.000000	10.000000	250898.090000	4.000000

	HasCrCard	IsActiveMember	EstimatedSalary	Exited
count	10000.00000	10000.000000	10000.000000	10000.000000
mean	0.70550	0.515100	100090.239881	0.203700
std	0.45584	0.499797	57510.492818	0.402769
min	0.00000	0.000000	11.580000	0.000000
25%	0.00000	0.000000	51002.110000	0.000000
50%	1.00000	1.000000	100193.915000	0.000000
75%	1.00000	1.000000	149388.247500	0.000000
max	1.00000	1.000000	199992.480000	1.000000

Visualizations: Here is a pivot table demonstrating the percentile of different genders and geographic locations exiting the bank.

In [8]:
```
visualization= churn_data.pivot_table("Exited", index="Gender", columns="Geography")
visualization
```

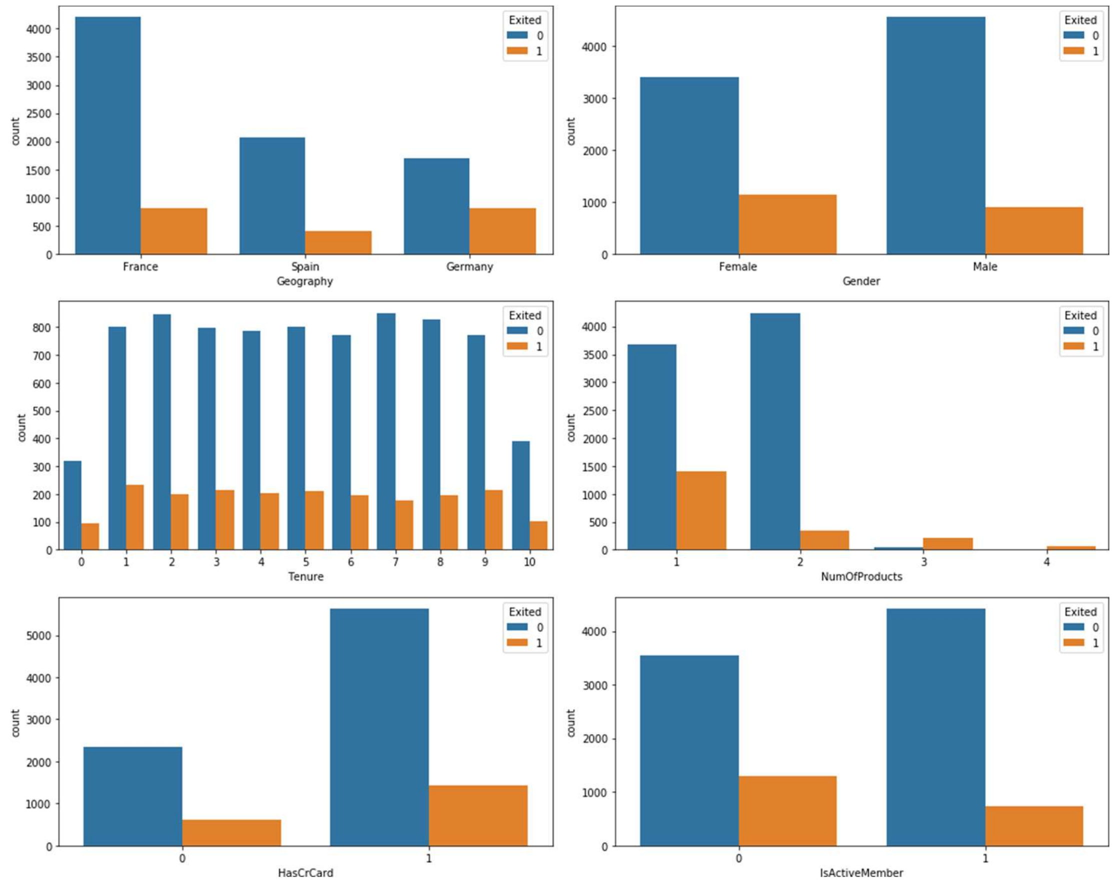
Out[8]:

Geography	France	Germany	Spain
Gender			
Female	0.203450	0.375524	0.212121
Male	0.127134	0.278116	0.131124

This pivot table explains two major trends, one is more number of females have have exited than the males in all geographic locations and the other one is country with most number of exits is Germany.

Target variables Vs descriptive Variables: The bleow visulaisation speaks about the number of people who exited the bank. The visualisation is categorised based on demographics such as geography,gender. The categorisation is aslo based on various financial features such as tenure, no: of products, has a credit card or not and is an active member ot not. The blue bars report the no: of people who still saty with the bank and the orange bar reports the no: of people who exited the bank.

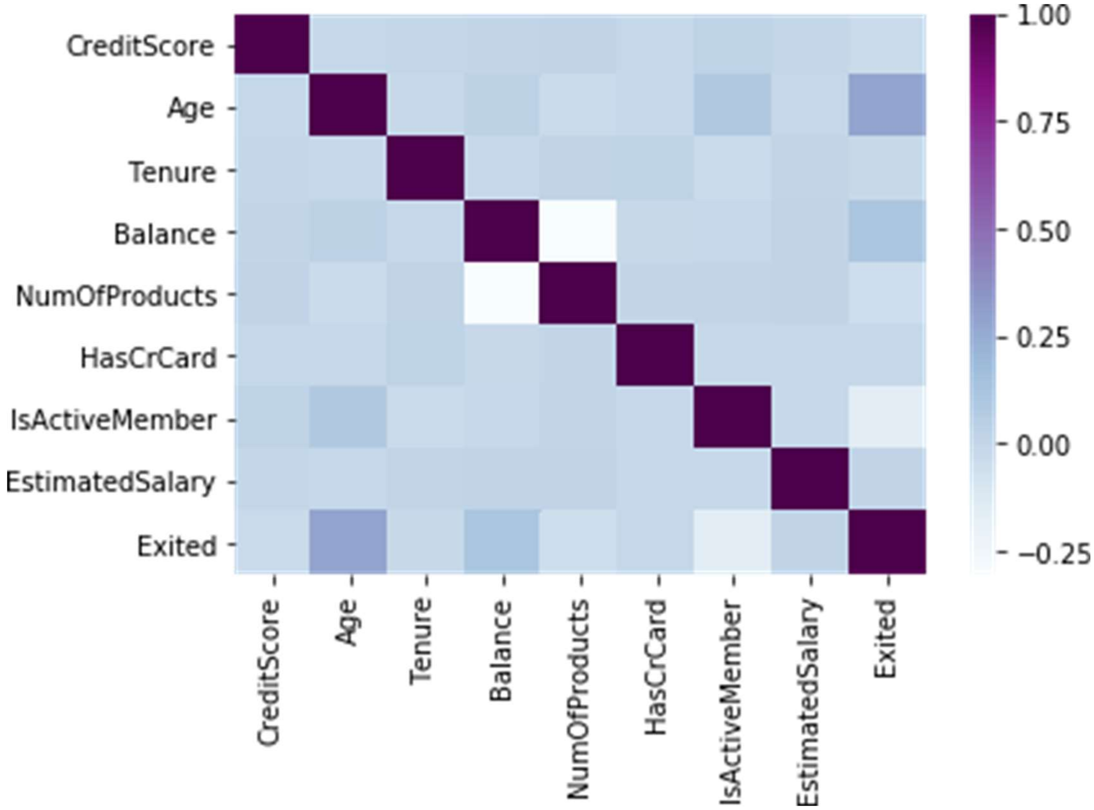
In [20]:
```
# using seaborn library for visualization
plot, axes = plt.subplots(3, 2, figsize=(15, 12))
axes = axes.flatten()

# extracting features with unique values that are between 2 and 49
unique_data = churn_data.nunique()

# array of categorical features
categorical_features = [column for column in churn_data.columns
                        if unique_data[column] >= 1 and unique_data[column] < 50]

# array of non-categorical non_categorical
numerical_features = [column for column in churn_data.columns
                      if unique_data[column] > 50]

# looping through the array of categorical features and
# plotting their counts with the target variable
for axis, categoricalplot in zip(axes, churn_data.dtypes[categorical_features].index):
    sns.countplot(x=categoricalplot, hue = 'Exited', data=churn_data, ax=axis)
plt.tight_layout()
plt.show()
```



Heat Map For Correlation: The Heat map shows the correlation amonst all the variables avaliable in the data set except the dropped variables. Its observeg that all the variables have weak and strong corelation with the target variable. Due to this all variables must be taken into consideration for bulding the model.NumOfProducts, IsActiveMember, CreditScore, Age, Balance,are the variables with significant level of correlation.

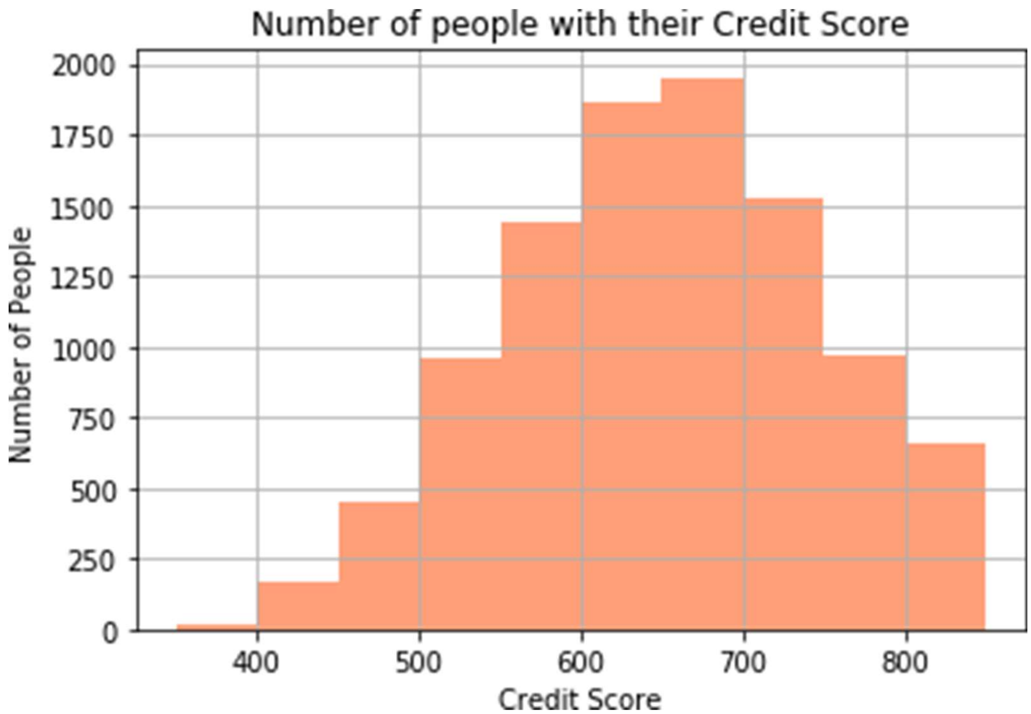
In [10]:
```
churn_data = churn_data.drop(["Geography", "Gender"], axis=1)
correlation = churn_data.corr()
sns.heatmap(correlation.T, cmap="BuPu")
```

Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0xc8fc550>



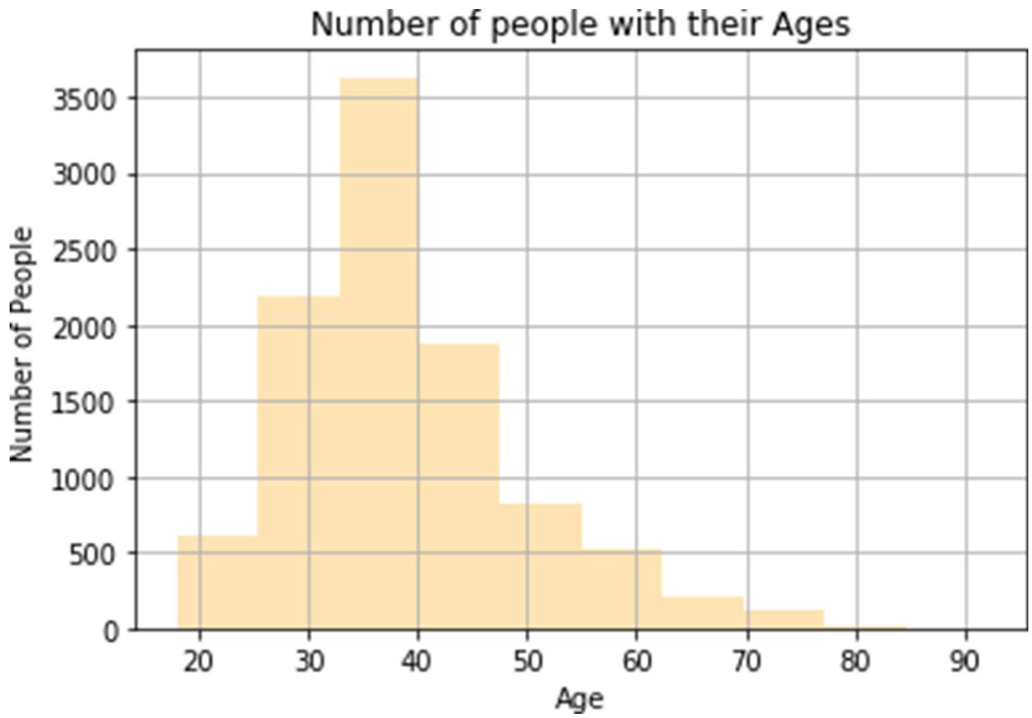
Credit Score VS No: of customers: The below visualised histogram is a comparison between credit score and no: of people. Its observed that majority of the customers have a credit score lying between 550 to 750.

In [17]:
```
churn_data['CreditScore'].hist(bins=10,color='lightsalmon')
# Histogram showing Number of people with their Credit Score
plt.title("Number of people with their Credit Score") #setting title
plt.xlabel('Credit Score') #setting x-axis label
plt.ylabel('Number of People') #setting y-axis label
plt.show() #show plot
```



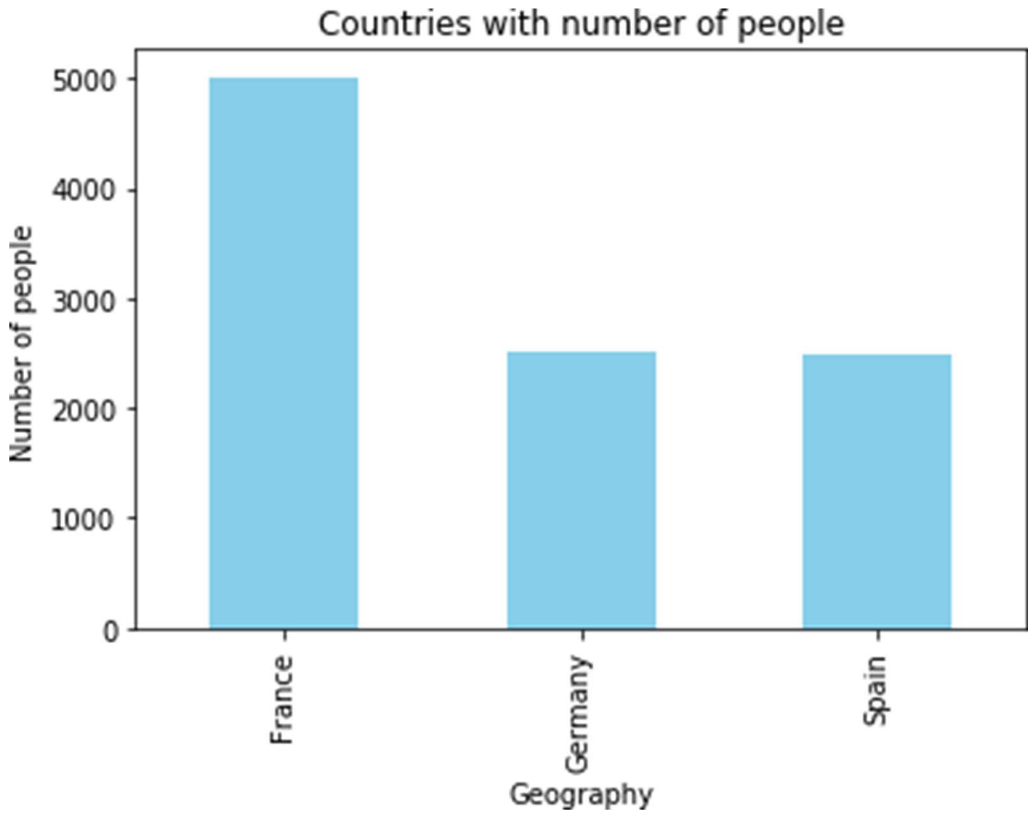
Age VS No: of customers: The below visualised histogram is a comparison between age and no: of people. Its observed that majority of the customers have their ages lying between 25 to 50.

In [18]:
```
churn_data['Age'].hist(bins=10,color='moccasin')
# Histogram showing Number of people with their Ages
plt.title("Number of people with their Ages") #setting title
plt.xlabel('Age') #setting x-axis label
plt.ylabel('Number of People') #setting y-axis label
plt.show() #show plot
```



Geography VS France: From the below visulaisation it is observed that majority of the customers reside in France and also same amount of customers reside in Germany and Spain.

In [19]:
```
churn_data['Geography'].value_counts().plot(kind='bar',color='skyblue')
# Histogram showing Countries with number of people
plt.title("Countries with number of people")#setting title
plt.ylabel('Number of people')#setting x-axis label
plt.xlabel('Geography');#setting y-axis label
plt.show() #show plot
```


Countries with number of people
Number of people
5000
4000
3000
2000
1000
0
France
Germany
Spain
Geography

In [ ]:

Bibliography

[1] Shruti Iyer. Churn Modelling: Classification Data Set.