Data Mining - Business Report

Rohan R. Khade

Table of Contents

List of Tables

List of Figures

Chapter 1. Problem 1: Clustering

1.1 Problem Statement

A leading bank wants to develop a customer segmentation to give promotional offers to its customers. They collected a sample that summarizes the activities of users during the past few months. You are given the task to identify the segments based on credit card usage.

1.2 Introduction

The dataset has 210 rows and 7 colums. The columns of the dataset include spending, advance payments, probability of full payment, current balance, credit limit, minimum (min) payment amount (amt), and maximum (max) spent in single shopping. The dataset provides a list of customers surveyed to understand the best promotional offer that can be offered by the bank to them.

1.2.1 Read the data, do the necessary initial steps, and exploratory data analysis (Univariate, Bi-variate, and multivariate analysis).

Table 1 Dataframe: bank (with head function)

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	19.94	16.92	0.8752	6.675	3.763	3.252	6.550
1	15.99	14.89	0.9064	5.363	3.582	3.336	5.144
2	18.95	16.42	0.8829	6.248	3.755	3.368	6.148
3	10.83	12.96	0.8099	5.278	2.641	5.182	5.185
4	17.99	15.86	0.8992	5.890	3.694	2.068	5.837
5	12.70	13.41	0.8874	5.183	3.091	8.456	5.000
6	12.02	13.33	0.8503	5.350	2.810	4.271	5.308
7	13.74	14.05	0.8744	5.482	3.114	2.932	4.825
8	18.17	16.26	0.8637	6.271	3.512	2.853	6.273
9	11.23	12.88	0.8511	5.140	2.795	4.325	5.003

Back to Top

Table 2 Dataframe: bank (with describe function)

	count	mean	std	min	25%	50%	75%	max
spending	210.0	14.847524	2.909699	10.5900	12.27000	14.35500	17.305000	21.1800
advance_payments	210.0	14.559286	1.305959	12.4100	13.45000	14.32000	15.715000	17.2500
probability_of_full_payment	210.0	0.870999	0.023629	0.8081	0.85690	0.87345	0.887775	0.9183
current_balance	210.0	5.628533	0.443063	4.8990	5.26225	5.52350	5.979750	6.6750
credit_limit	210.0	3.258605	0.377714	2.6300	2.94400	3.23700	3.561750	4.0330
min_payment_amt	210.0	3.700201	1.503557	0.7651	2.56150	3.59900	4.768750	8.4560
max_spent_in_single_shopping	210.0	5.408071	0.491480	4.5190	5.04500	5.22300	5.877000	6.5500

Figure 1. Dataset information

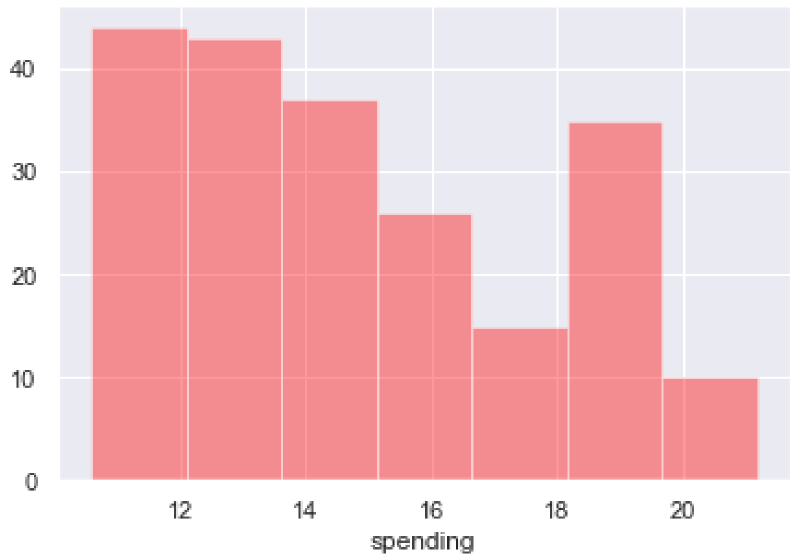
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 210 entries, 0 to 209
Data columns (total 7 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   spending                      210 non-null    float64
 1   advance_payments              210 non-null    float64
 2   probability_of_full_payment   210 non-null    float64
 3   current_balance               210 non-null    float64
 4   credit_limit                  210 non-null    float64
 5   min_payment_amt               210 non-null    float64
 6   max_spent_in_single_shopping  210 non-null    float64
dtypes: float64(7)
memory usage: 11.6 KB
```

The dataset has no null values as the total number of rows are 210 and all the data types are in float form. However, when we look at the columns, we can assume that as there are no columns to uniquely identify the customers, all 210 entries are unique entries and no duplicates.

1.2.1.1 Univariate Analysis

Figure 2. Spending data series: Description & graphical representation

```
Description of spending
---------------------------------------------------------
count    210.000000
mean      14.847524
std        2.909699
min       10.590000
25%       12.270000
50%       14.355000
75%       17.305000
max       21.180000
Name: spending, dtype: float64 Distribution of spending
---------------------------------------------------------
```



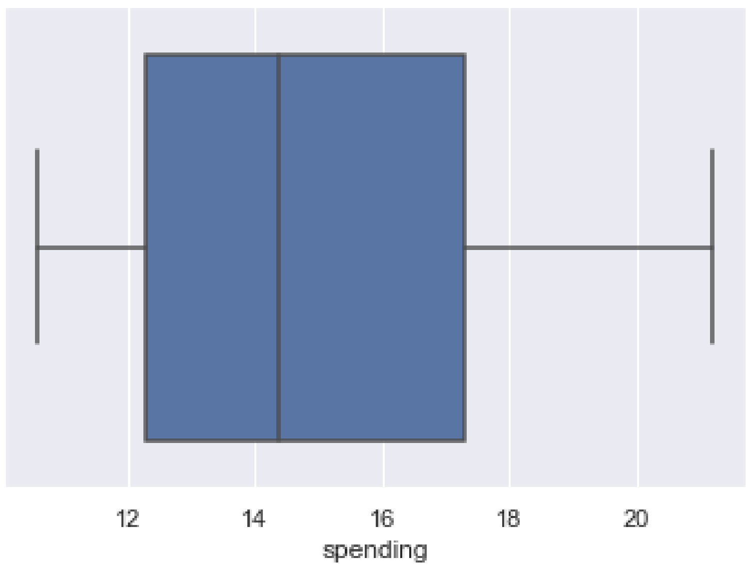
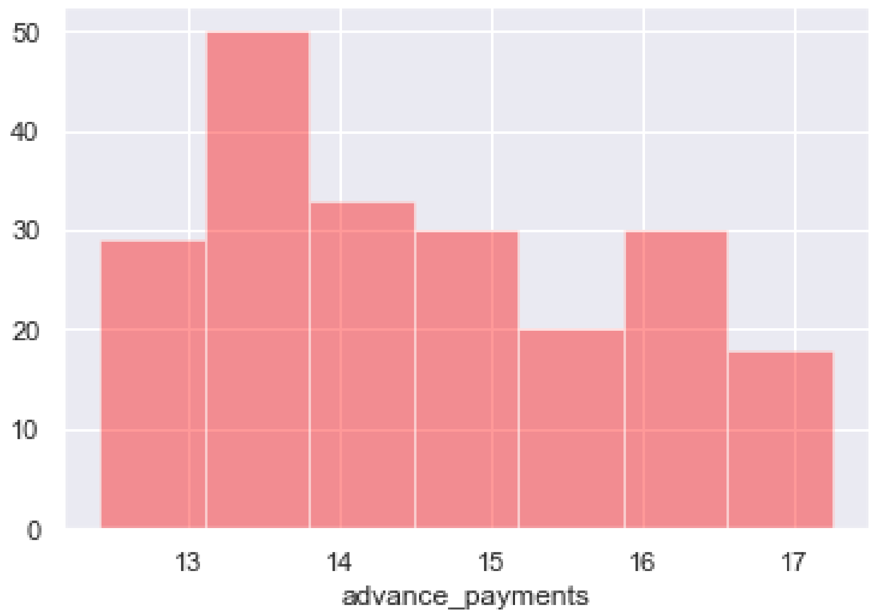


Figure 3. **Advance payments data series: Description & graphical representation**

```
Description of advance_payments
--------------------------------------------------------------------------
count     210.000000
mean       14.559286
std         1.305959
min        12.410000
25%        13.450000
50%        14.320000
75%        15.715000
max        17.250000
Name: advance_payments, dtype: float64 Distribution of advance_payments
--------------------------------------------------------------------------
```



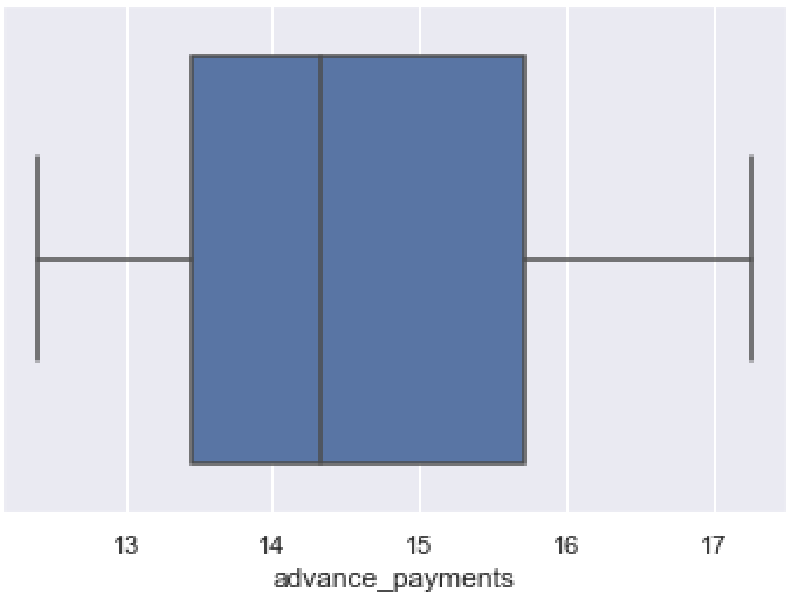
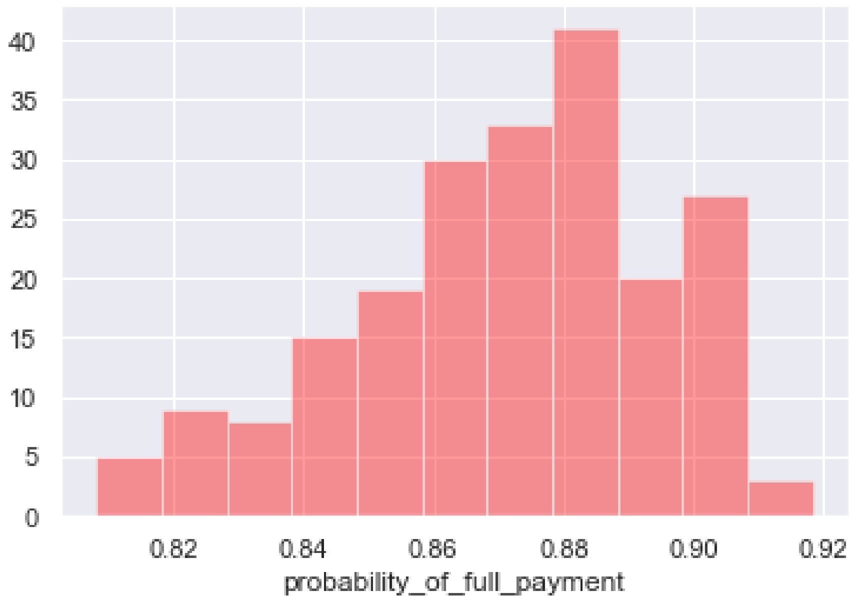


Figure 4. Probability of full payment data series: Description & graphical representation

```
Description of probability_of_full_payment
-----------------------------------------------------------------------------
count    210.000000
mean       0.870999
std        0.023629
min        0.808100
25%        0.856900
50%        0.873450
75%        0.887775
max        0.918300
Name: probability_of_full_payment, dtype: float64 Distribution of probability_of_full_payment
-----------------------------------------------------------------------------
```



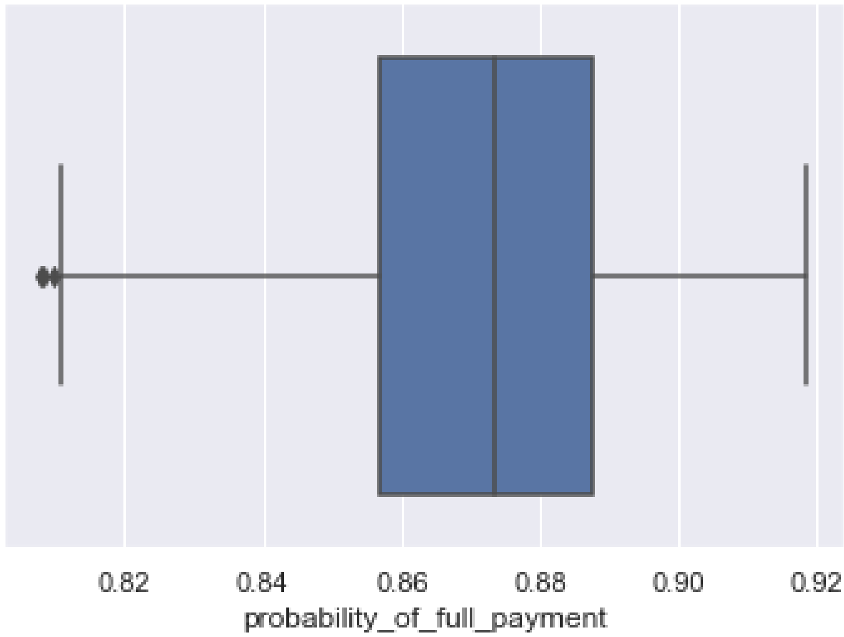
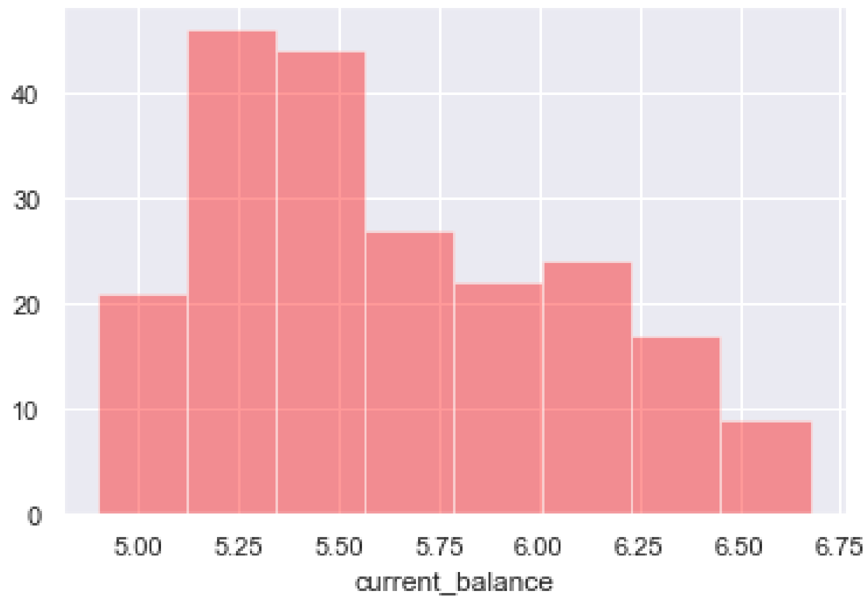
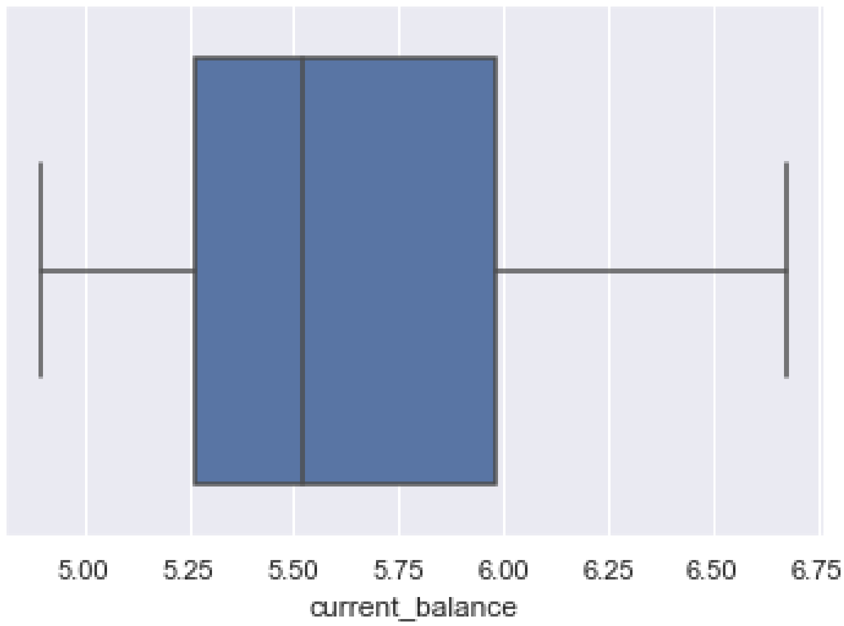


Figure 5. Current balance data series: Description & graphical representation

```
Description of current_balance
----------------------------------------------------------------------------
count    210.000000
mean       5.628533
std        0.443063
min        4.899000
25%        5.262250
50%        5.523500
75%        5.979750
max        6.675000
Name: current_balance, dtype: float64 Distribution of current_balance
----------------------------------------------------------------------------
```

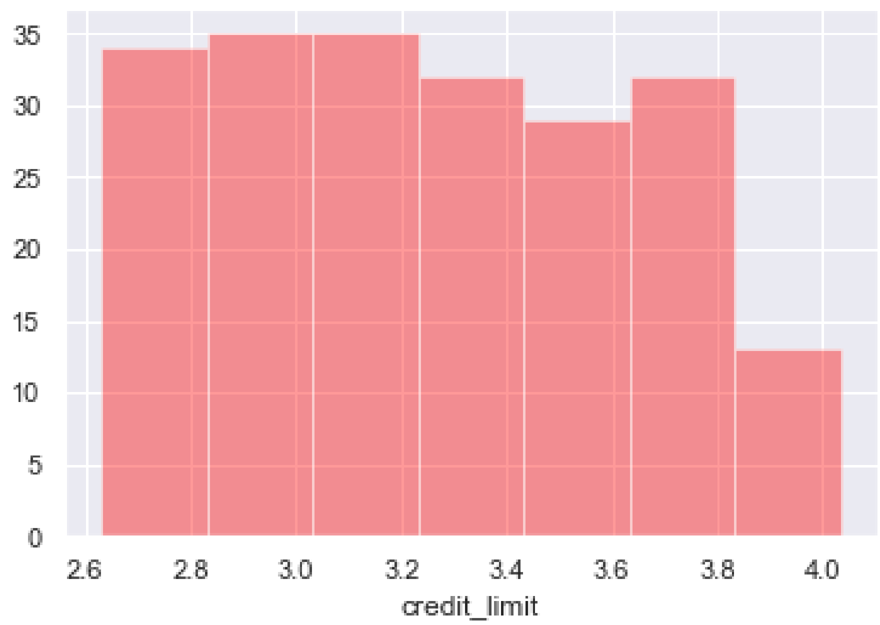


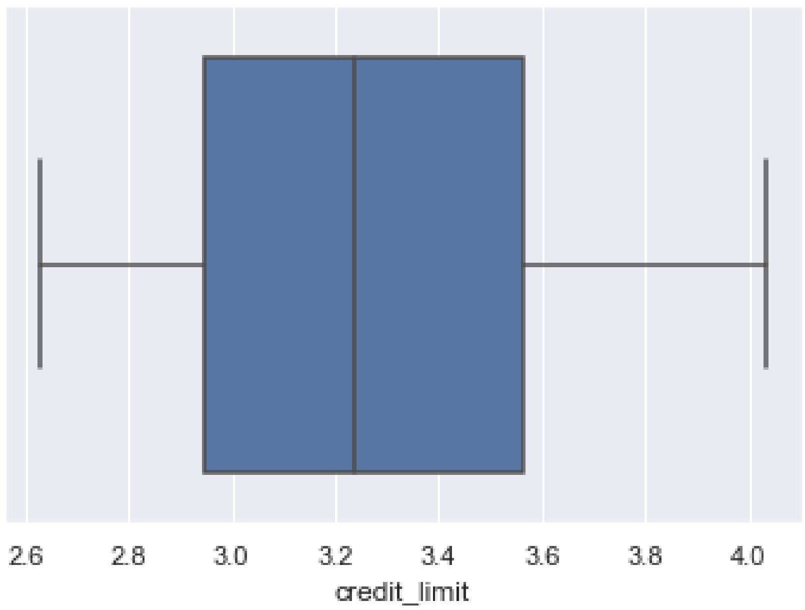


Back to Top

Figure 6. **Credit limit data series: Description & graphical representation**

```
Description of credit_limit
----------------------------------------------------------------
count    210.000000
mean       3.258605
std        0.377714
min        2.630000
25%        2.944000
50%        3.237000
75%        3.561750
max        4.033000
Name: credit_limit, dtype: float64 Distribution of credit_limit
----------------------------------------------------------------
```

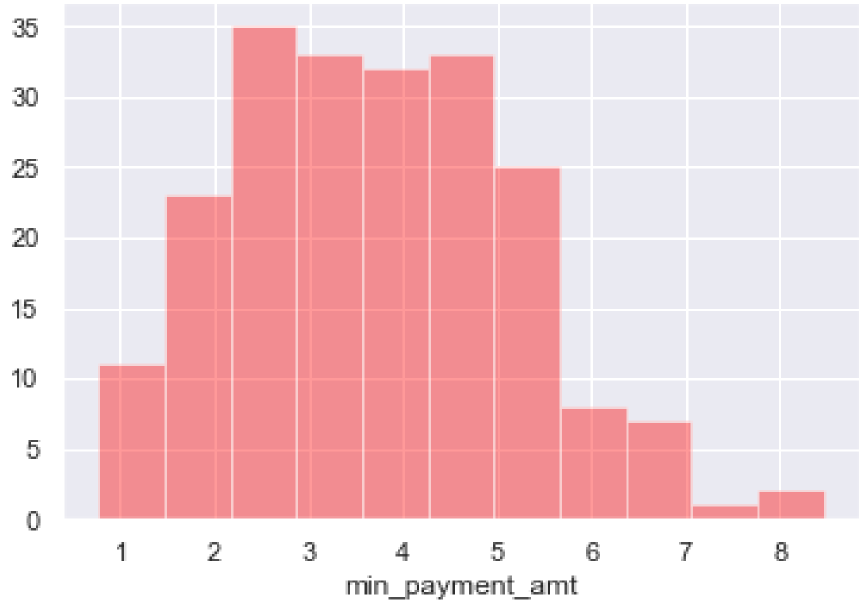


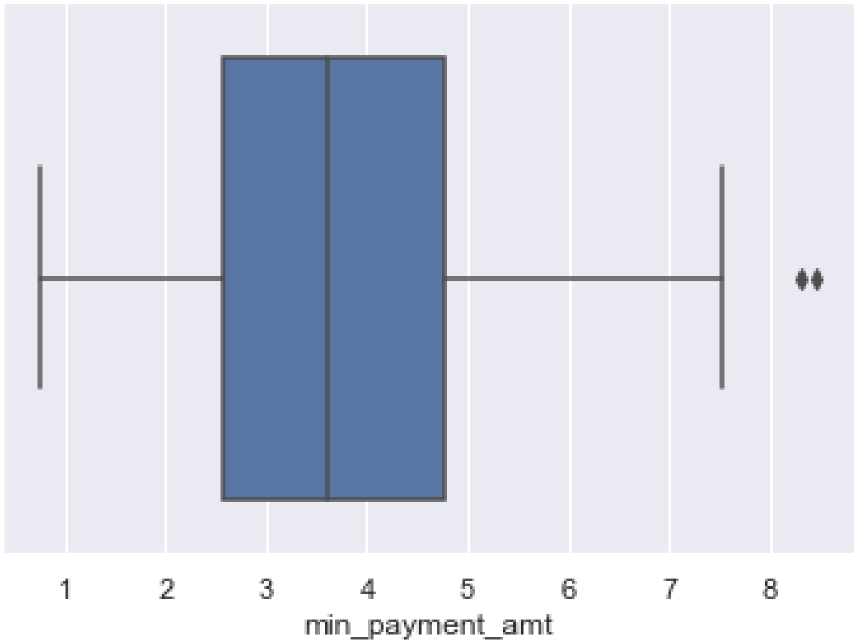


Back to Top

Figure 7. **Minimum payment amount data series: Description & graphical representation**

```
Description of min_payment_amt
----------------------------------------------------------------------
count    210.000000
mean       3.700201
std        1.503557
min        0.765100
25%        2.561500
50%        3.599000
75%        4.768750
max        8.456000
Name: min_payment_amt, dtype: float64 Distribution of min_payment_amt
----------------------------------------------------------------------
```

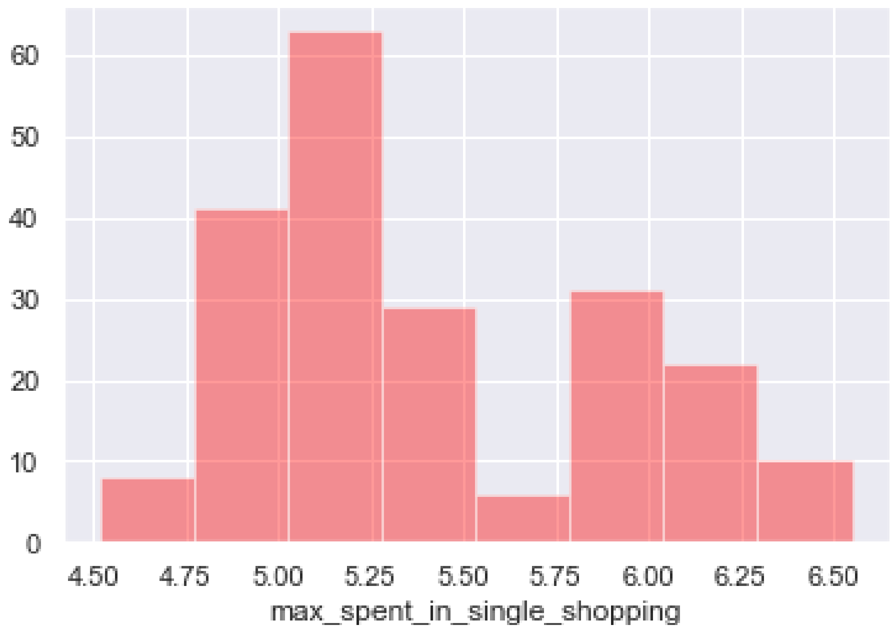




Back to Top

Figure 8. Maximum spent in single shopping data series: Description & graphical representation

```
Description of max_spent_in_single_shopping
---------------------------------------------------------------------------
count    210.000000
mean       5.408071
std        0.491480
min        4.519000
25%        5.045000
50%        5.223000
75%        5.877000
max        6.550000
Name: max_spent_in_single_shopping, dtype: float64 Distribution of max_spent_in_single_shopping
---------------------------------------------------------------------------
```



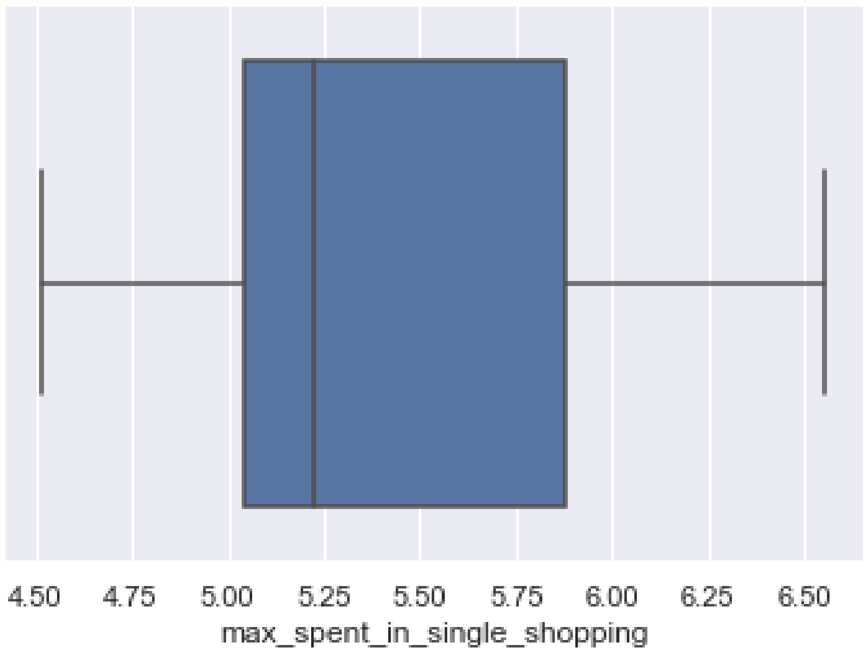
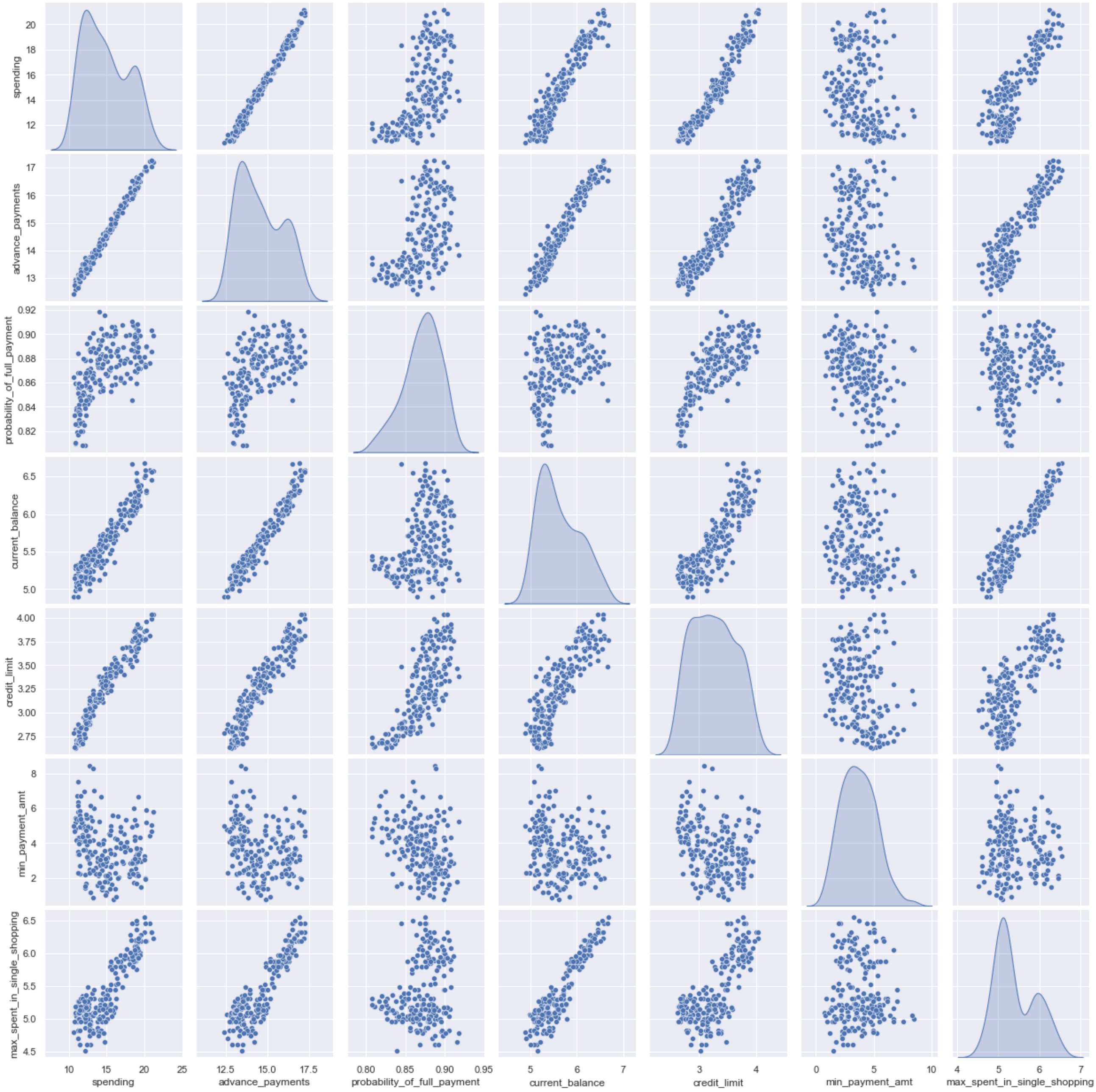


Figure 9. Skewness of the present seven variables

```
max_spent_in_single_shopping    0.561897
current_balance                 0.525482
min_payment_amt                 0.401667
spending                        0.399889
advance_payments                0.386573
credit_limit                    0.134378
probability_of_full_payment    -0.537954
dtype: float64
```

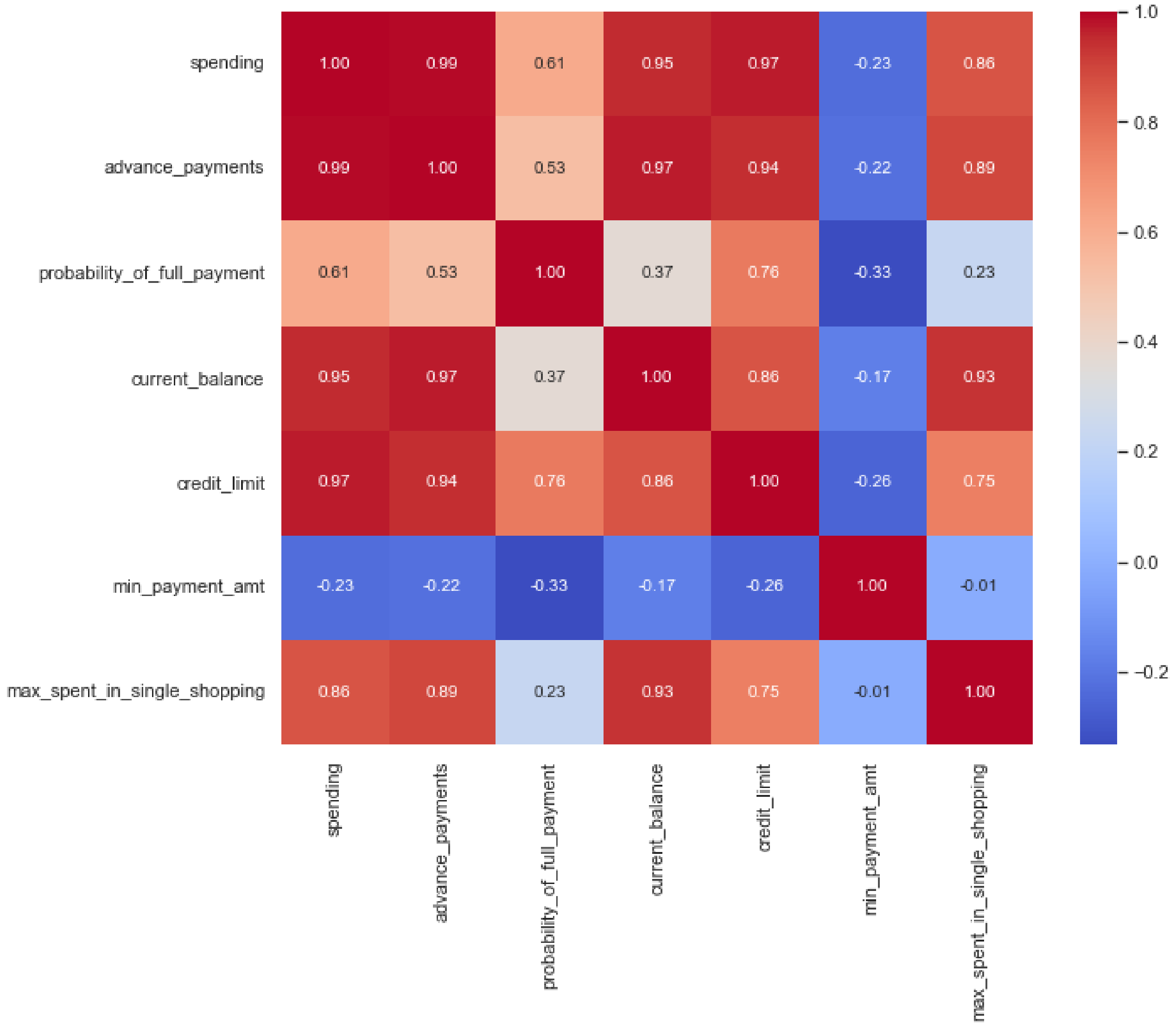
1.2.1.2 Bivariate Analysis

Figure 10. Pairplot



Back to Top

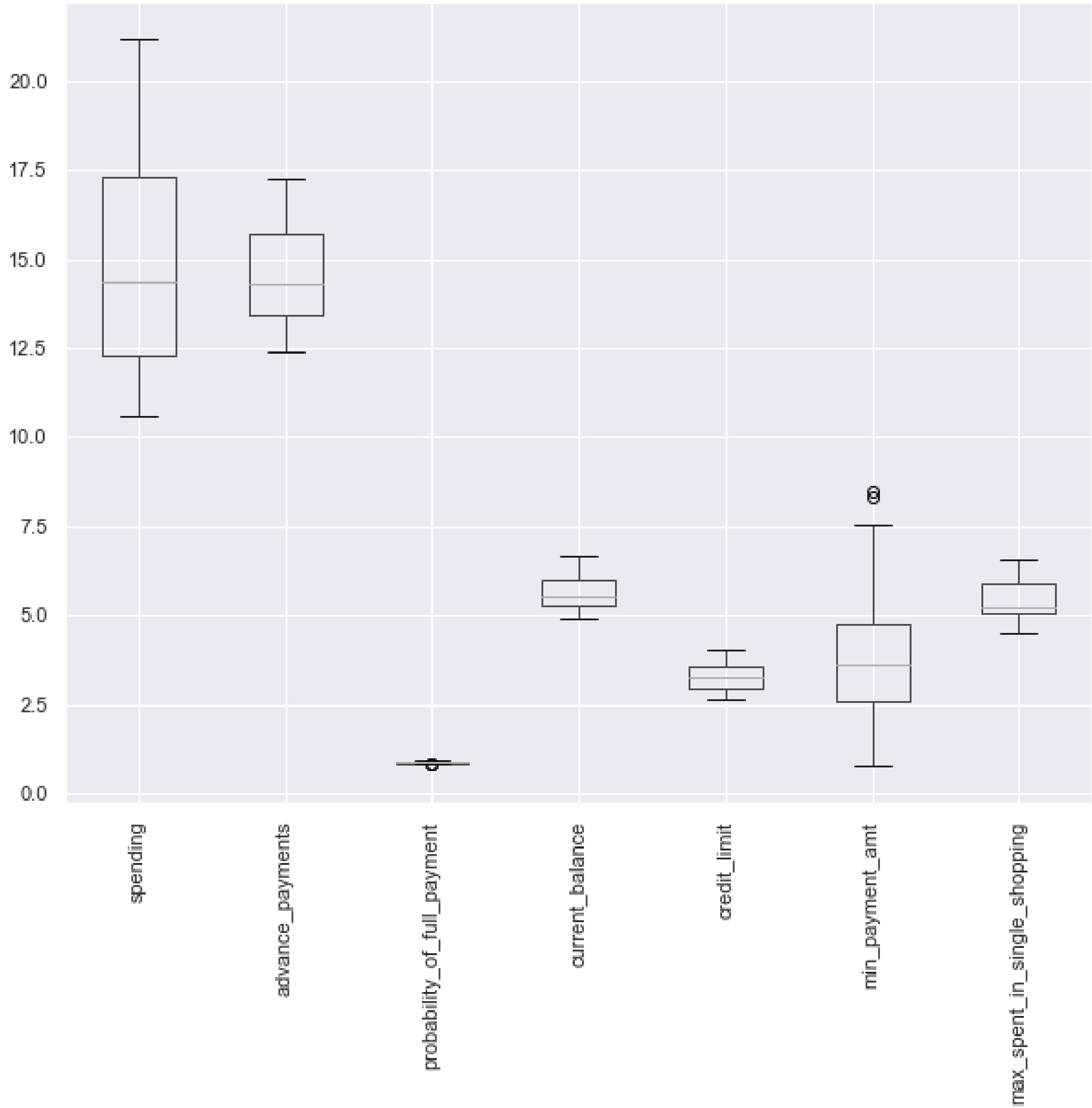
Figure 11. Heatmap



Analyzing the pairplot and heatmap, we can deduce that spending variable is directly related with advance payments, current balance, credit limit, and maximum spent in single shopping. Higher spending customers have higher current balance, advance payments, credit limit, and spending per single shopping. Furthermore, there is a strong correlation between current balance and maximum spent in single shopping & advance payments.

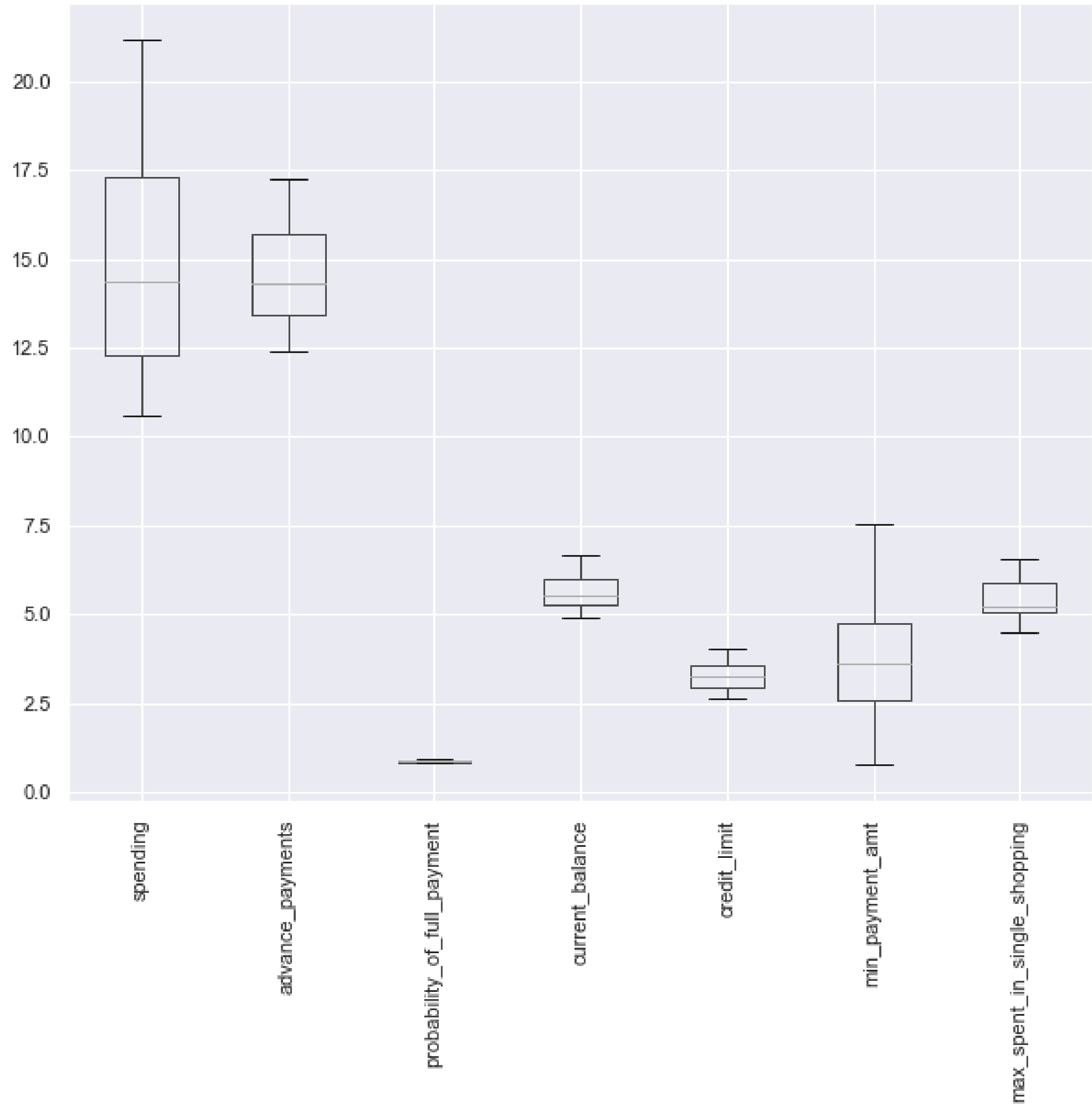
There is also a high correlation between credit limit and advance payments & spending. There is a weaker correlation between minimum payment amount and all the other variables, which can also be understood as there is higher correlation when it comes to spending and maximum spending in single shopping & current balance. There is also weaker correlation between probability of full payment and current balance.

Figure 12. Box plot with outliers



We can see of the seven variables included as part of the dataset, there are minimal outliers for two variables. However, we need to treat them for better accuracy in the clustering model.

Figure 13. Box plot post outlier treatment



We have taken 5, 25, 75 percentile of the column to treat the outliers. We have calculated IQR range and minimum threshold while calculating the lower bound and upper bound values to treat the outliers on the left of the lower whisker and right of the upper whisker respectively.

1.2.2 Do you think scaling is necessary for clustering in this case? Justify

Yes, scaling of the dataset is necessary as each column has been provided in different units i.e. in 100s, 1000s, and 10,000s. Spending, advance payment, and credit limit may get more weightage as compared to other variables. We need to bring them in the relative same range. So, we have used z score method to scale the data which is provided below:

Table 3 Scaled dataset: bank_scaled (with head function)

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	1.754355	1.811968	0.171955	2.367533	1.338579	-0.294861	2.328998
1	0.393582	0.253840	1.528129	-0.600744	0.858236	-0.236880	-0.538582
2	1.413300	1.428192	0.506652	1.401485	1.317348	-0.214791	1.509107
3	-1.384034	-1.227533	-1.970322	-0.793049	-1.639017	1.037338	-0.454961
4	1.082581	0.998364	1.215165	0.591544	1.155464	-1.112128	0.874813

1.2.3 Apply hierarchical clustering to scaled data. Identify the number of optimum clusters using Dendrogram and briefly describe them

We have considered two methods for clustering including wardlink and average. However, we will continue with average hierarchical clustering as wardlink method instead of measuring the distance directly, it analyzes the variance of clusters whereas in average clustering the distance between two clusters is defined as the average of distances between all pairs of objects, where each pair is made up of one object from each group.

Figure 14. Dendrogram with average linkage method (without truncating)

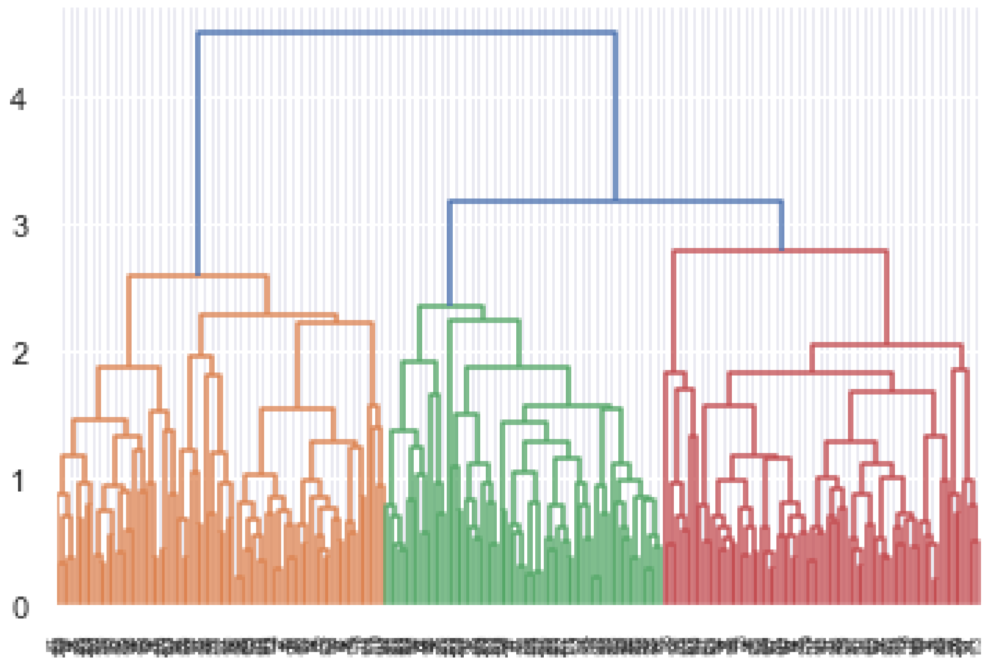
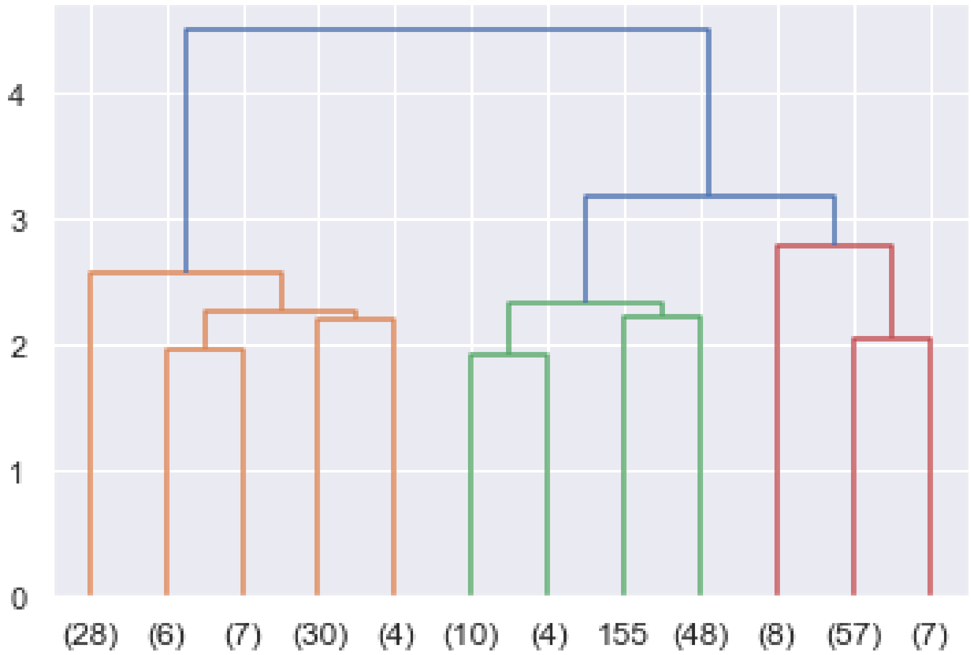


Figure 15. Dendrogram with average linkage method (post truncating)



We have clustered the scaled data set in 3 clusters using maxclust criterion. Three clusters seems to be most optimum number for clustering using average linkage method in a **new column called clusters** included as part of **new dataframe called cluster_bank.**

Table 4 Scaled dataset with clusters: bank_scaled (with head function)

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	clusters
0	19.94	16.92	0.875200	6.675	3.763	3.252	6.550	1
1	15.99	14.89	0.906400	5.363	3.582	3.336	5.144	2
2	18.95	16.42	0.882900	6.248	3.755	3.368	6.148	1
3	10.83	12.96	0.825915	5.278	2.641	5.182	5.185	3
4	17.99	15.86	0.899200	5.890	3.694	2.068	5.837	1

Figure 16. Value counts by clusters

```
1    75
2    63
3    72
Name: clusters, dtype: int64
```

We have also formed a new dataframe called final_bank, wherein we can analyze clustering on basis of seven columns given in the dataset as given below:

Table 5 Cluster dataset with seven variables: final_bank (Frequency table)

clusters	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Freq
1	18.129200	16.058000	0.881595	6.135747	3.648120	3.650200	5.987040	75
2	14.167302	14.186190	0.882776	5.451381	3.236794	2.377956	5.048698	63
3	12.024306	13.324583	0.850371	5.255194	2.871944	4.847925	5.119431	72

By analyzing the dataset by both the clustering methods, we can understand three clusters helps us examine the spending patterns of 210 customers included as part of the dataset. We have effectively divided the variables in three clusters to analyze the results further. The three group cluster solution gives a pattern based on high/ medium/ low spending with maximum spent in single shopping and probability of full payment.

1.2.4 Apply K-Means clustering on scaled data and determine optimum clusters. Apply elbow curve and silhouette score. Explain the results properly. Interpret and write inferences on the finalized clusters.

After we apply K Means clustering with:

One cluster: We get an output of 1469.99

Two clusters: We get an output of 659.13

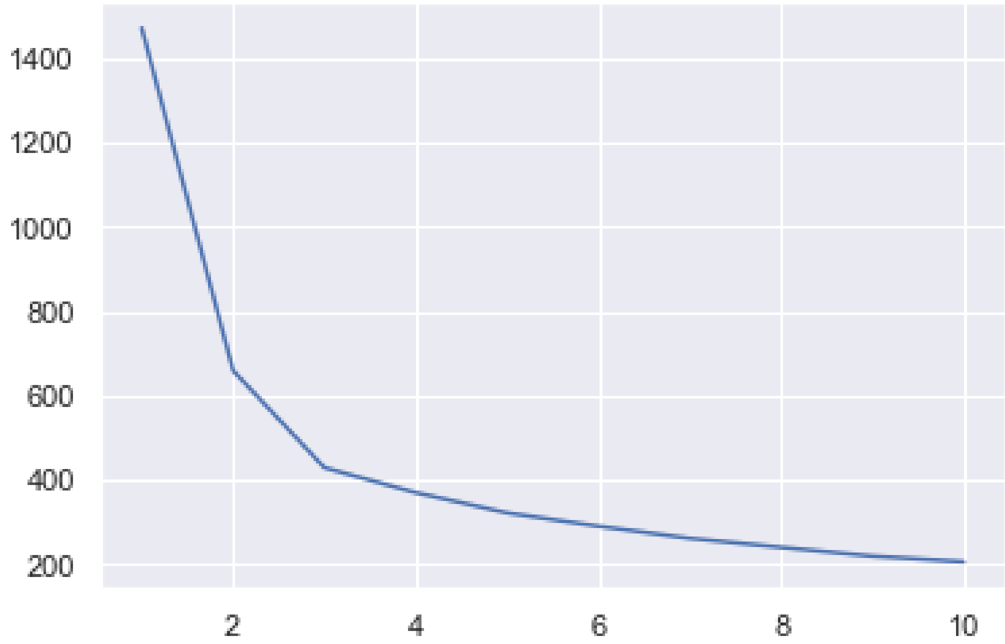
Three clusters: We get an output of 429.41

After applying K Means clustering for more than three clusters, there is a minimum drop, which makes the optimum number of cluster to be three. We have provided an output for 10 clusters below:

```
[1469.9999999999998,
 659.1308122335327,
 429.4139632109118,
 370.26944971303885,
 322.1752568424512,
 290.6666199436945,
 261.9694459859648,
 240.78498586728637,
 219.69480575660603,
 206.56204967535638]
```

It is evident that post three clusters, the drop in the means is minimal. We can also see it in below elbow graph, wherein we can see an elbow like shape forming after third point. In addition, the three clusters cover over 75% of the data as shown in the below graph:

Figure 17. Elbow graph



The silhouette score for the dataset is used for measuring the mean of the silhouette coefficient for each sample belonging to different clusters.which is 0.47906, wherein silhouette sample provides the Silhouette scores for each sample of different clusters which is integrated with the scaled dataset below:

Figure 18. Silhouette samples (with head function)

spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Clus_kmeans	sil_width
1.754355	1.811968	0.171955	2.367533	1.338579	-0.294861	2.328998	1	0.581663
0.393582	0.253840	1.528129	-0.600744	0.858236	-0.236880	-0.538582	0	0.383562
1.413300	1.428192	0.506652	1.401485	1.317348	-0.214791	1.509107	1	0.647782
-1.384034	-1.227533	-1.970322	-0.793049	-1.639017	1.037338	-0.454961	2	0.620992
1.082581	0.998364	1.215165	0.591544	1.155464	-1.112128	0.874813	1	0.393615

Back to Top

1.2.5 Describe cluster profiles for the clusters defined. Recommend different promotional strategies for different clusters.

clusters	1	2	3
spending	18.129200	14.167302	12.024306
advance_payments	16.058000	14.186190	13.324583
probability_of_full_payment	0.881595	0.882776	0.850371
current_balance	6.135747	5.451381	5.255194
credit_limit	3.648120	3.236794	2.871944
min_payment_amt	3.650200	2.377956	4.847925
max_spent_in_single_shopping	5.987040	5.048698	5.119431
Freq	75.000000	63.000000	72.000000

We can segment the customers Based on their spending into:

- High spenders
- Medium spenders
- Low spenders

As spending is directly related with the credit limit, advance payments, current balance, and maximum spent in single shopping,

The bank can provide the following **promotional strategies to** its customers**:**

- As high spenders have higher repayment capacity and are fairly regular, their **credit limit can be increased**

- Also, their spent per single shopping is high so they can be given **appropriate discounts at regular intervals** may be through **a points based system**

- **High** and **medium** spenders **can be offered loans**, based on their account types, they can be offered other facilities such as **premium credit cards**, current account holders can be given better **overdraft or other related services**

Back to Top

- The bank can provide **promotional offers that will basically increasing customer spending** which will in return benefit the bank itself

- The bank can provide **referral benefits** to the high and medium spenders which will **reduce the acquisition cost of a new customer**

- The bank can **tie-up with ecommerce platforms** from different **industries** such as **travel & tourism, food & groceries delivery,** popular **food chains & cafes**, **retail stores** such as clothing, luxury goods, electronics, etc. and **cab services**

- To ensure **timely payments**, bank can provide **reward points** to low spenders **for early payments** to improve their **spending habits** and **shift them into medium spenders** category

Chapter 2. CART-RF-ANN

2.1 Problem Statement

An Insurance firm providing tour insurance is facing higher claim frequency. The management decides to collect data from the past few years. You are assigned the task to make a model which predicts the claim status and provide recommendations to management. Use CART, RF & ANN and compare the models' performances in train and test sets.

2.2 Introduction

The dataset has 3000 rows and 10 colums. The columns of the dataset include age, agency code, type, claimed, commission, channel, duration, sales, product name, and destination. The dataset provides a list of insures several variables to understand the claim patterns for tour insurance.

2.2.1 Read the data, do the necessary initial steps, and exploratory data analysis (Univariate, Bi-variate, and Multivariate analysis).

Table 6 Dataframe: df (with head function)

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
0	48	C2B	Airlines	No	0.70	Online	7	2.51	Customised Plan	ASIA
1	36	EPX	Travel Agency	No	0.00	Online	34	20.00	Customised Plan	ASIA
2	39	CWT	Travel Agency	No	5.94	Online	3	9.90	Customised Plan	Americas
3	36	EPX	Travel Agency	No	0.00	Online	4	26.00	Cancellation Plan	ASIA
4	33	JZI	Airlines	No	6.30	Online	53	18.00	Bronze Plan	ASIA
5	45	JZI	Airlines	Yes	15.75	Online	8	45.00	Bronze Plan	ASIA
6	61	CWT	Travel Agency	No	35.64	Online	30	59.40	Customised Plan	Americas
7	36	EPX	Travel Agency	No	0.00	Online	16	80.00	Cancellation Plan	ASIA
8	36	EPX	Travel Agency	No	0.00	Online	19	14.00	Cancellation Plan	ASIA
9	36	EPX	Travel Agency	No	0.00	Online	42	43.00	Cancellation Plan	ASIA

Table 7 Dataframe: df (with describe with include all function)

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Age	3000.0	NaN	NaN	NaN	38.091	10.463518	8.0	32.0	36.0	42.0	84.0
Agency_Code	3000	4	EPX	1365	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Type	3000	2	Travel Agency	1837	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Claimed	3000	2	No	2076	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Commision	3000.0	NaN	NaN	NaN	14.529203	25.481455	0.0	0.0	4.63	17.235	210.21
Channel	3000	2	Online	2954	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Duration	3000.0	NaN	NaN	NaN	70.001333	134.053313	-1.0	11.0	26.5	63.0	4580.0
Sales	3000.0	NaN	NaN	NaN	60.249913	70.733954	0.0	20.0	33.0	69.0	539.0
Product Name	3000	5	Customised Plan	1136	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Destination	3000	3	ASIA	2465	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Figure 19. Dataset information

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3000 entries, 0 to 2999
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Age           3000 non-null   int64  
 1   Agency_Code   3000 non-null   object 
 2   Type          3000 non-null   object 
 3   Claimed       3000 non-null   object 
 4   Commision     3000 non-null   float64
 5   Channel       3000 non-null   object 
 6   Duration      3000 non-null   int64  
 7   Sales         3000 non-null   float64
 8   Product Name  3000 non-null   object 
 9   Destination   3000 non-null   object 
dtypes: float64(2), int64(2), object(6)
memory usage: 234.5+ KB
```

The dataset has no null values as the total number of rows are 3000 and the data types are in float, integer, or object form. We have also tried to identify the duplicates and **we have gotten 139 duplicate entries**. However, when we look at the columns, **we can assume** that as there are **no columns to uniquely identify the customers**, all 3000 entries are **unique entries and not duplicates**.

2.2.1.1Univariate Analysis

For continuous variables:

To analyze each of the relevant columns, we have given a value counts function with outputs below:

```
Agency_Code
 EPX     1365
C2B      924
CWT      472
JZI      239
Name: Agency_Code, dtype: int64
```

```
Type
 Travel Agency    1837
Airlines         1163
Name: Type, dtype: int64
```

```
Claimed
 No     2076
Yes     924
Name: Claimed, dtype: int64
```

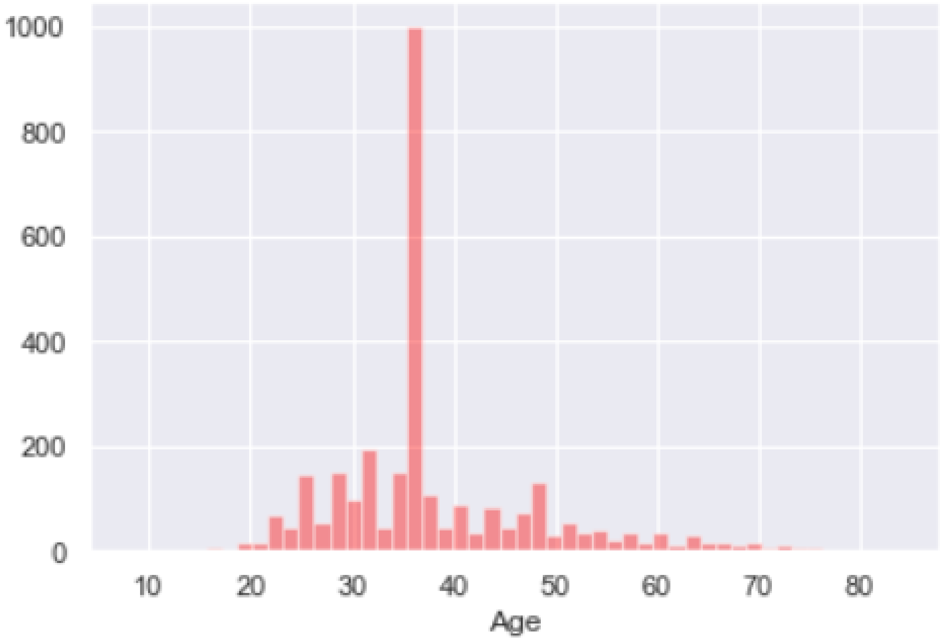
```
Channel
 Online     2954
Offline      46
Name: Channel, dtype: int64
```

```
Product Name
 Customised Plan      1136
Cancellation Plan     678
Bronze Plan           650
Silver Plan           427
Gold Plan             109
Name: Product Name, dtype: int64
```

```
Destination
 ASIA        2465
Americas     320
EUROPE       215
Name: Destination, dtype: int64
```

Figure 20. Age data series: Description & graphical representation

```
Description of Age
----------------------------------------------
count    3000.000000
mean       38.091000
std        10.463518
min         8.000000
25%        32.000000
50%        36.000000
75%        42.000000
max        84.000000
Name: Age, dtype: float64 Distribution of Age
----------------------------------------------
```


1000
800
600
400
200
0
10
20
30
40
50
60
70
80
Age

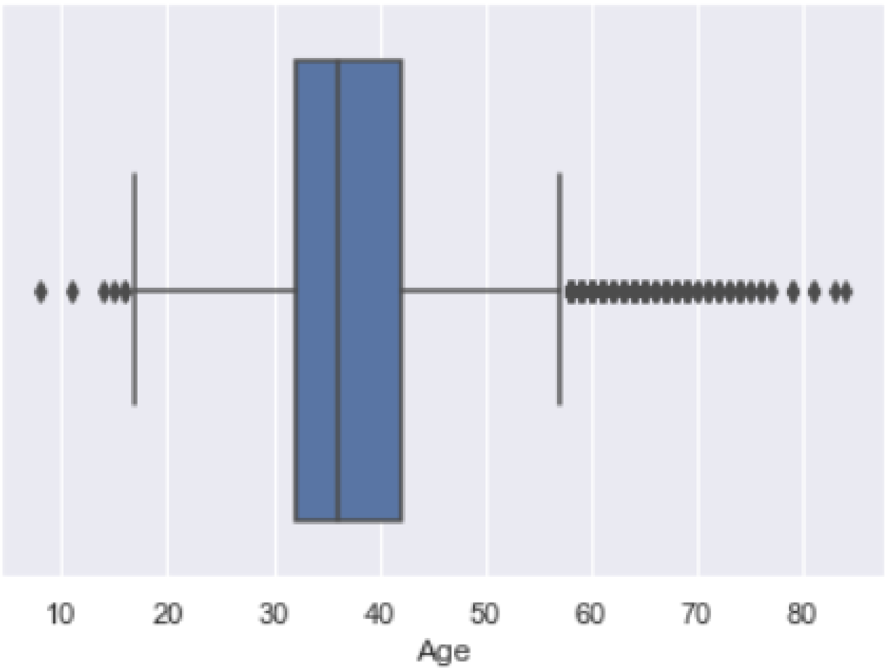
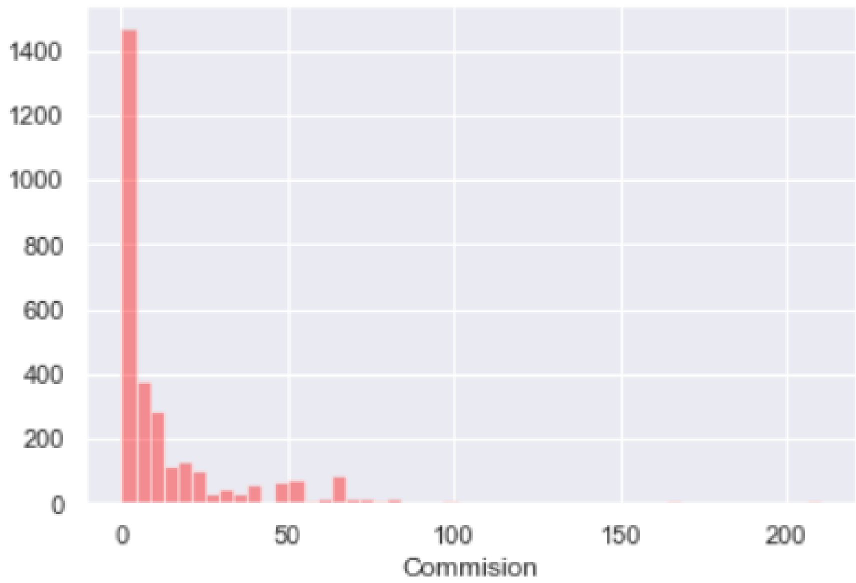
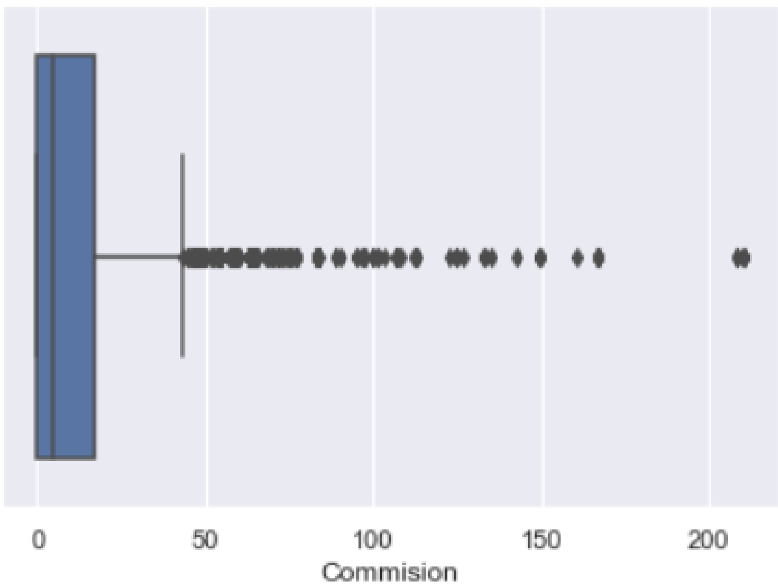

10
20
30
40
50
60
70
80
Age

Figure 21. **Commision data series: Description & graphical representation**

```
Description of Commision
-----------------------------------------------------------
count    3000.000000
mean       14.529203
std        25.481455
min         0.000000
25%         0.000000
50%         4.630000
75%        17.235000
max       210.210000
Name: Commision, dtype: float64 Distribution of Commision
-----------------------------------------------------------
```

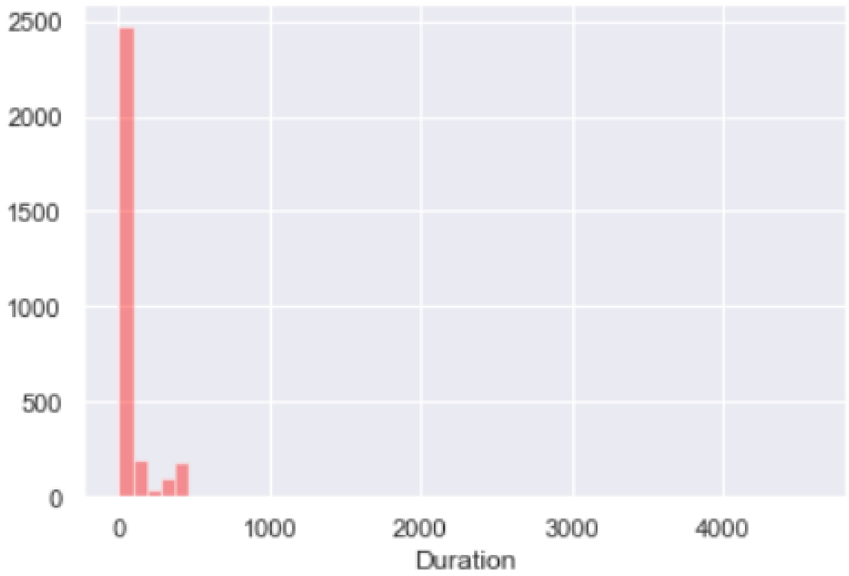




Back to Top

Figure 22. Duration data series: Description & graphical representation

```
Description of Duration
--------------------------------------------------------
count    3000.000000
mean       70.001333
std       134.053313
min        -1.000000
25%        11.000000
50%        26.500000
75%        63.000000
max      4580.000000
Name: Duration, dtype: float64 Distribution of Duration
--------------------------------------------------------
```



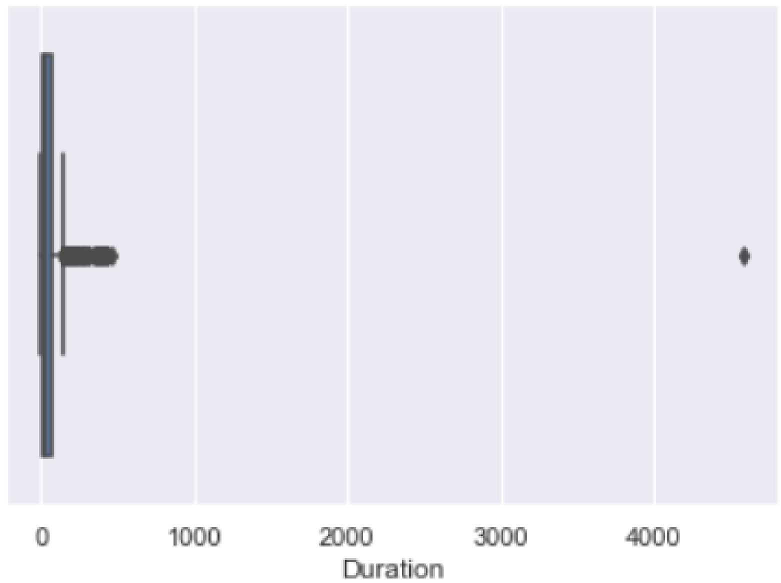
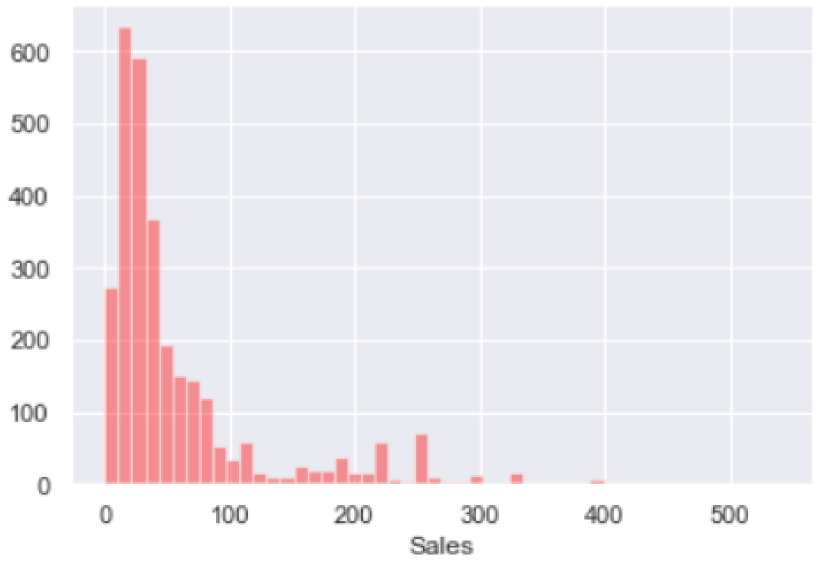
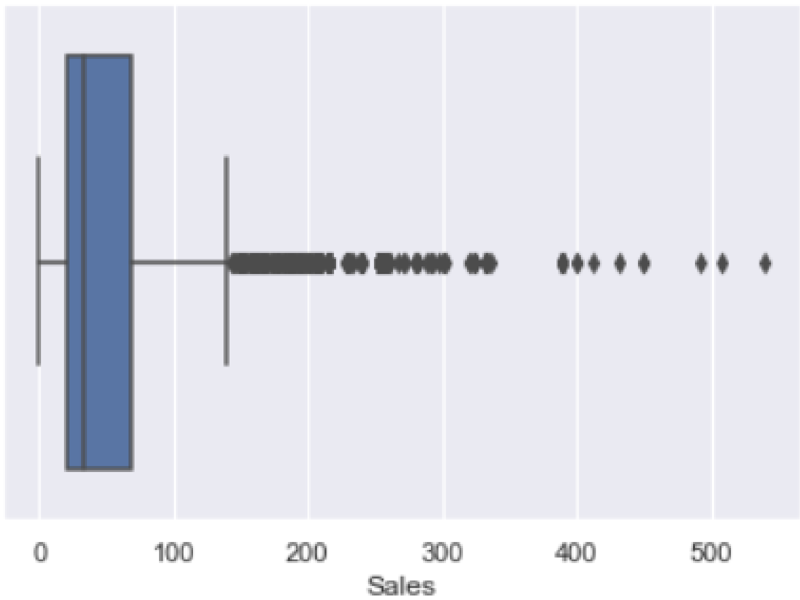


Figure 23. Sales data series: Description & graphical representation

```
Description of Sales
---------------------------------------------------
count    3000.000000
mean       60.249913
std        70.733954
min         0.000000
25%        20.000000
50%        33.000000
75%        69.000000
max       539.000000
Name: Sales, dtype: float64 Distribution of Sales
---------------------------------------------------
```





Back to Top

For object type variables:

Figure 24. Agency code data series: Description & graphical representation

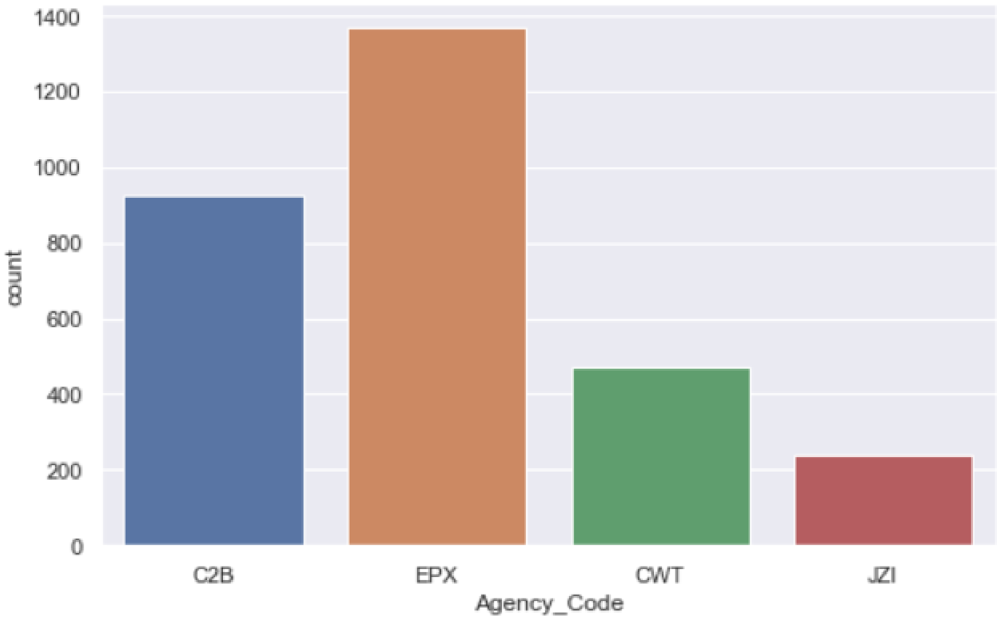
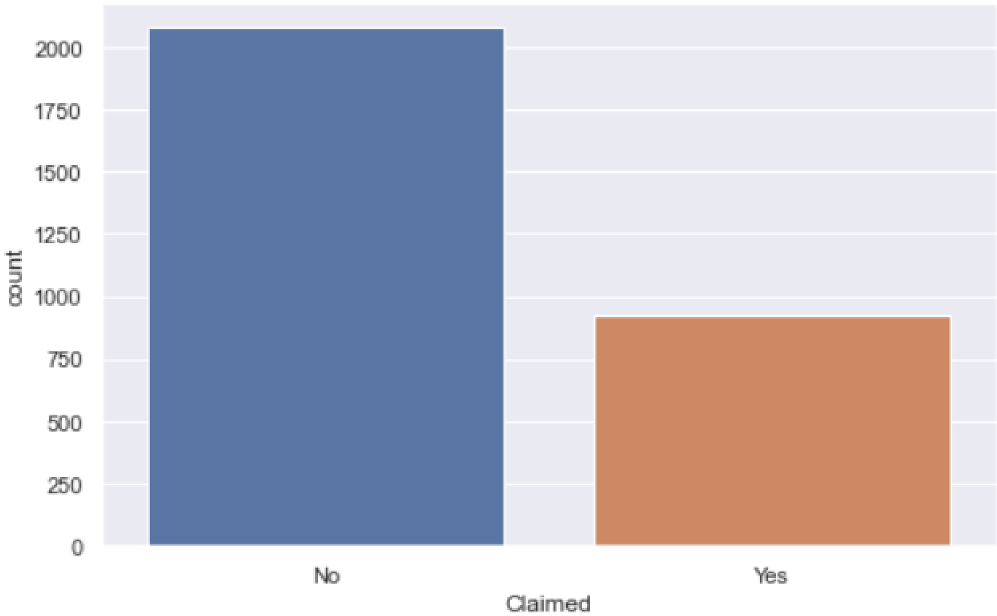


Figure 25. Claimed data series: Description & graphical representation



We can clearly see that EPX accounted for highest share with C2B and CWT acccouting for second and third largest share in terms of agency codes. However, we can see that more than 50% of the of the customers haven't made any claims.

Figure 26. Channel data series: Description & graphical representation

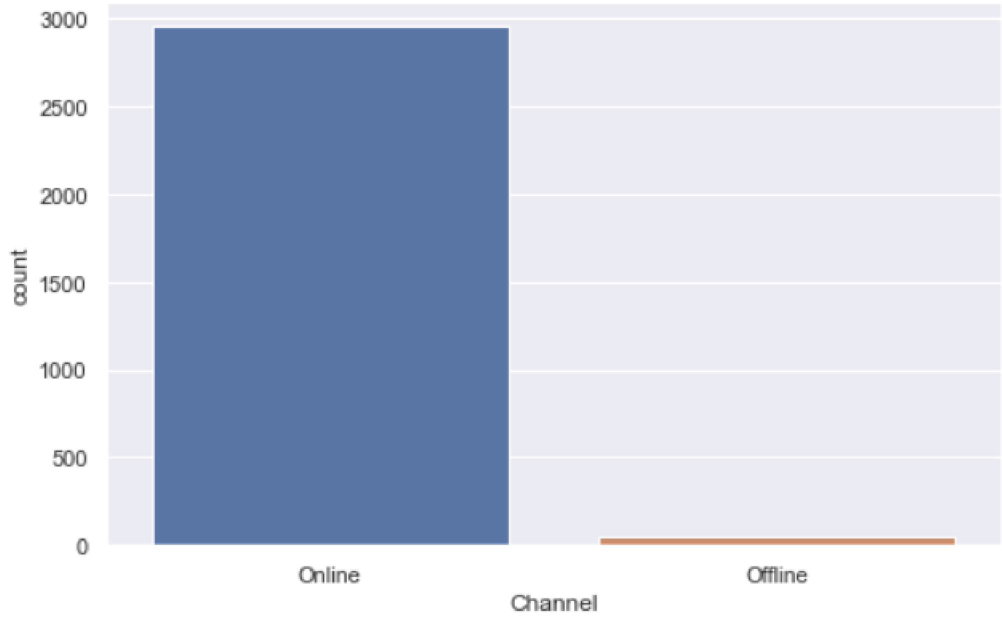
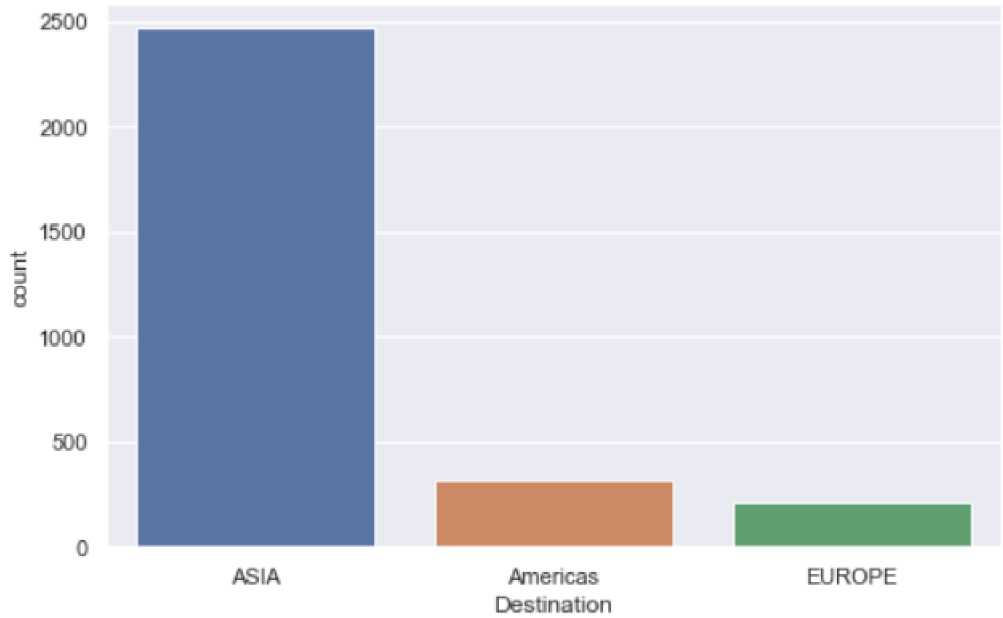


Figure 27. Destination data series: Description & graphical representation



Over 85% of the customers bought their tour insurance policies online and most of them were travelling to Asia as their final destination. Around 500 to 600 customers were travelling to either America or Europe.

Figure 28. Type data series: Description & graphical representation

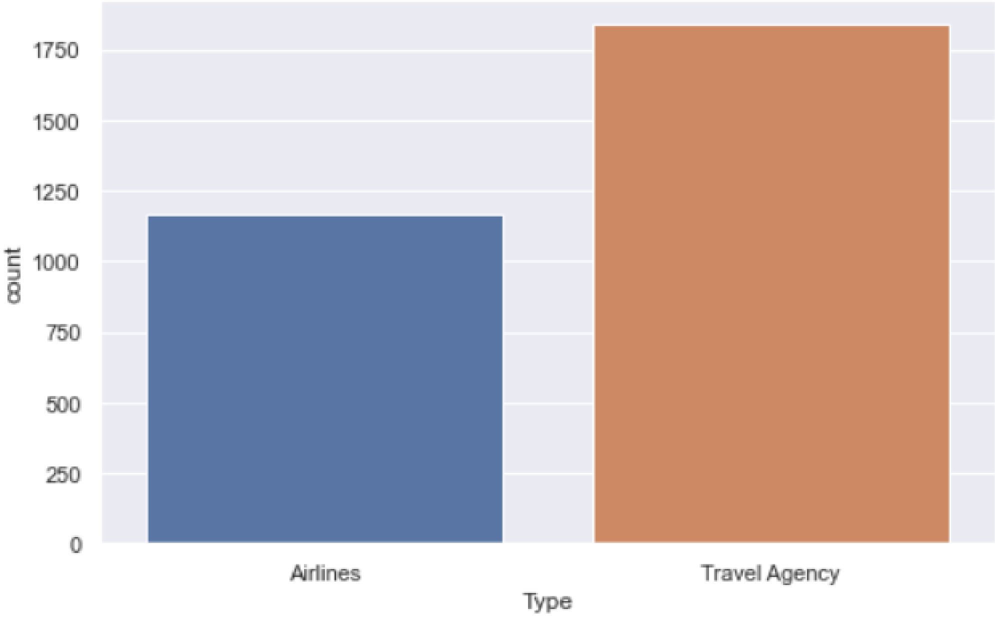
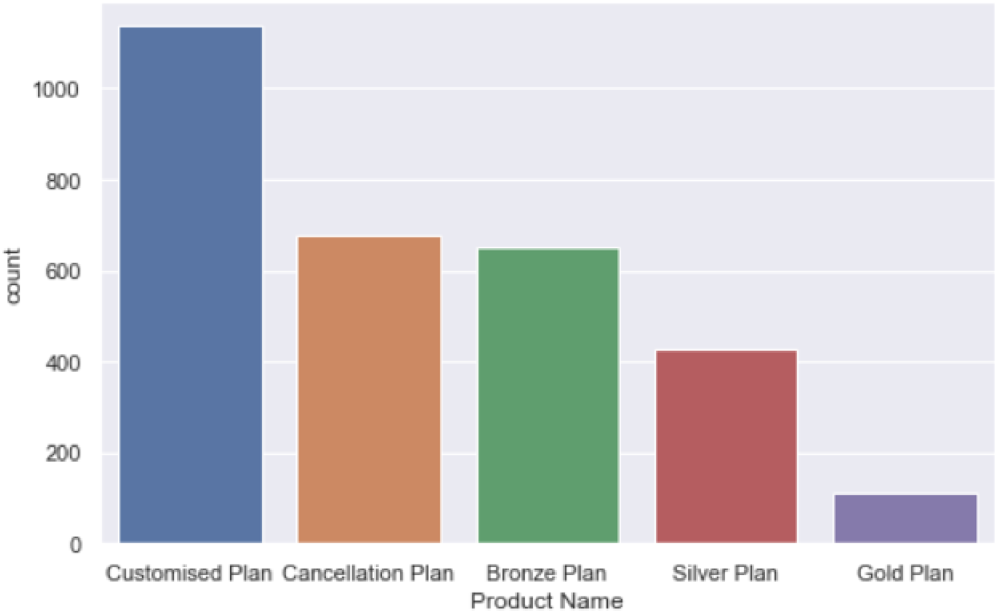


Figure 29. Product name data series: Description & graphical representation



Just over 61% of the travellers got their insurance policies from travel agency itself instead of airlines. Furthermore, a major chunk of the customers bought a customized plan and a small number of customers bought gold plan which we can assume must be mostly costly one.

2.2.1.2 Bivariate and Multivariate Analysis

Figure 30. Pairplot (Claimed variable as hue)

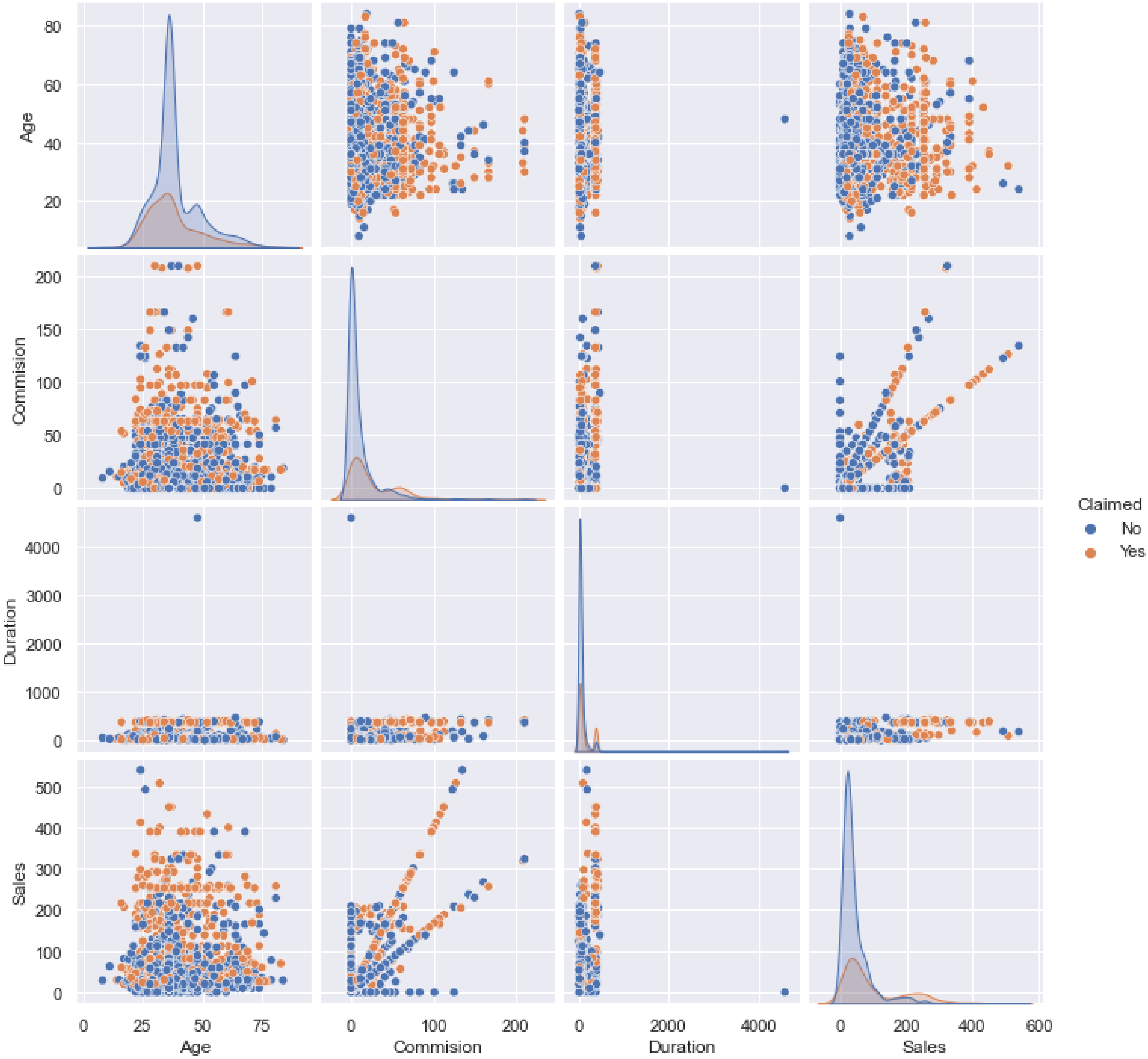


Figure 31. Agency code: Swarmplot

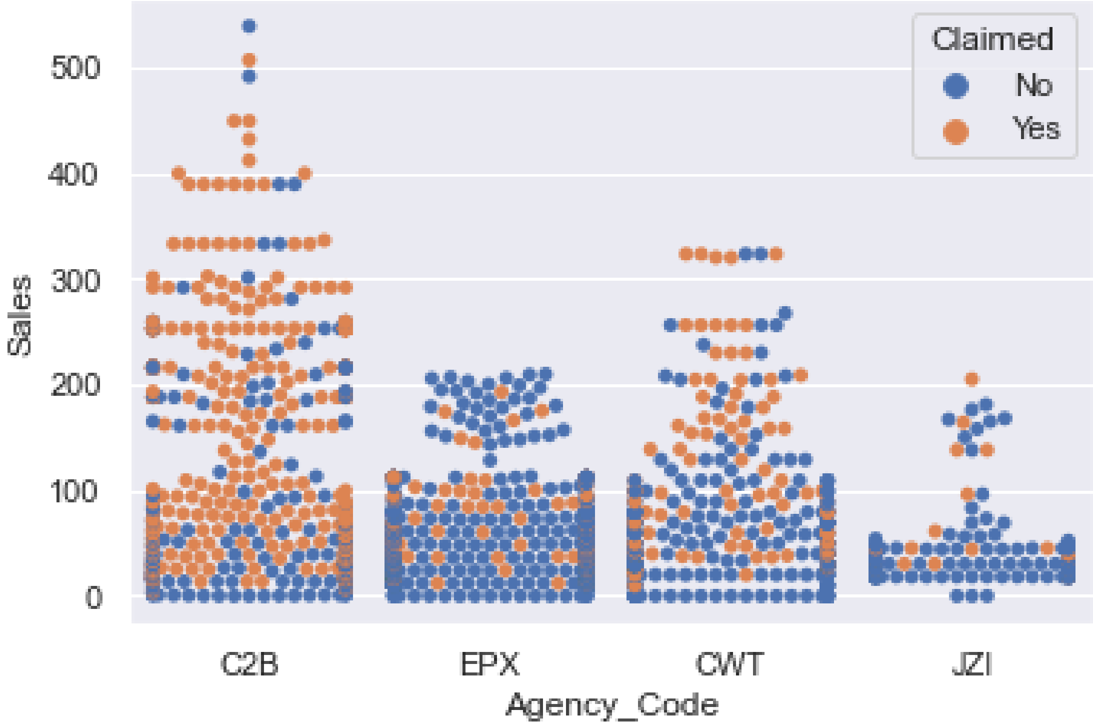


Figure 32. Channel: Swarmplot

Figure 33. **Product Name: Swarmplot**

Figure 34. **Type: Swarmplot**

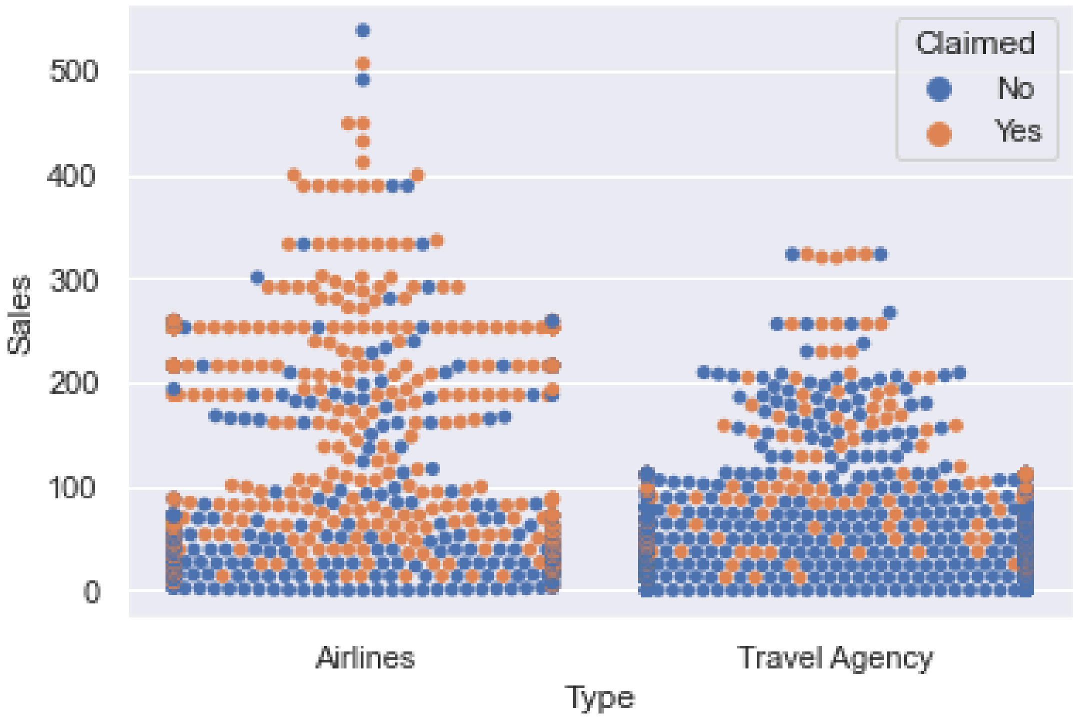
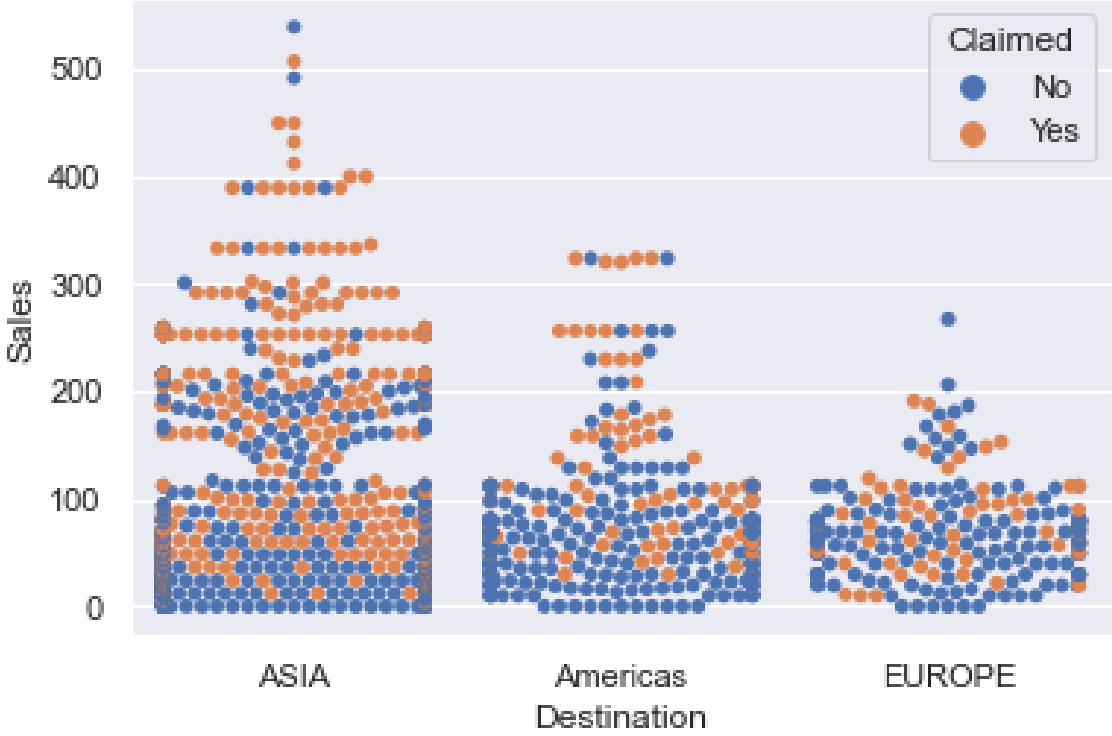


Figure 35. **Destination: Swarmplot**



Looking at the swarmplots, we can deduce the following:

- C2B agency code has highest number of sales and highest number of claims as we can see a large number of orange dots in the plot
- We can assume that more than 85% of the sales happened from online platform so the customers which claimed against the tour insurance policy are from online channel
- Even though majority of the sales were for customized plan, majority of the claims were from the customers who bought the sliver and gold plan
- We know that just over 62% customers bought policies from travel agency; however, customers insured through airlines claimed much more as compared to its counterpart
- Customers travelling to Asia claimed more as compared to the people travelling to Americas or Europe

2.2.2 Data Split: Split the data into test and train, build classification model CART, Random Forest, Artificial Neural Network

Decision Tree

We have creasted two datasets **y which comprises the target variable "Claimed"** and **x wherein we have saved the complete dataset except the target variable**. We have split the data into train and test with **test size** as **0.30 or 30%** and **random state as 1**. We have given the shape function to the train and test data for x and y:

```
X_train (2100, 9)
X_test (900, 9)
train_labels (2100,)
test_labels (900,)
```

To understand the best parameters for decision tree with **gini** creiterion and **cross validation index as 3**, while using GridSearchCV, we have given range of parameters such as max depth, min sample leaf, and min sample split for which we got the following output:

```
{'max_depth': 10, 'min_samples_leaf': 50, 'min_samples_split': 150}
```

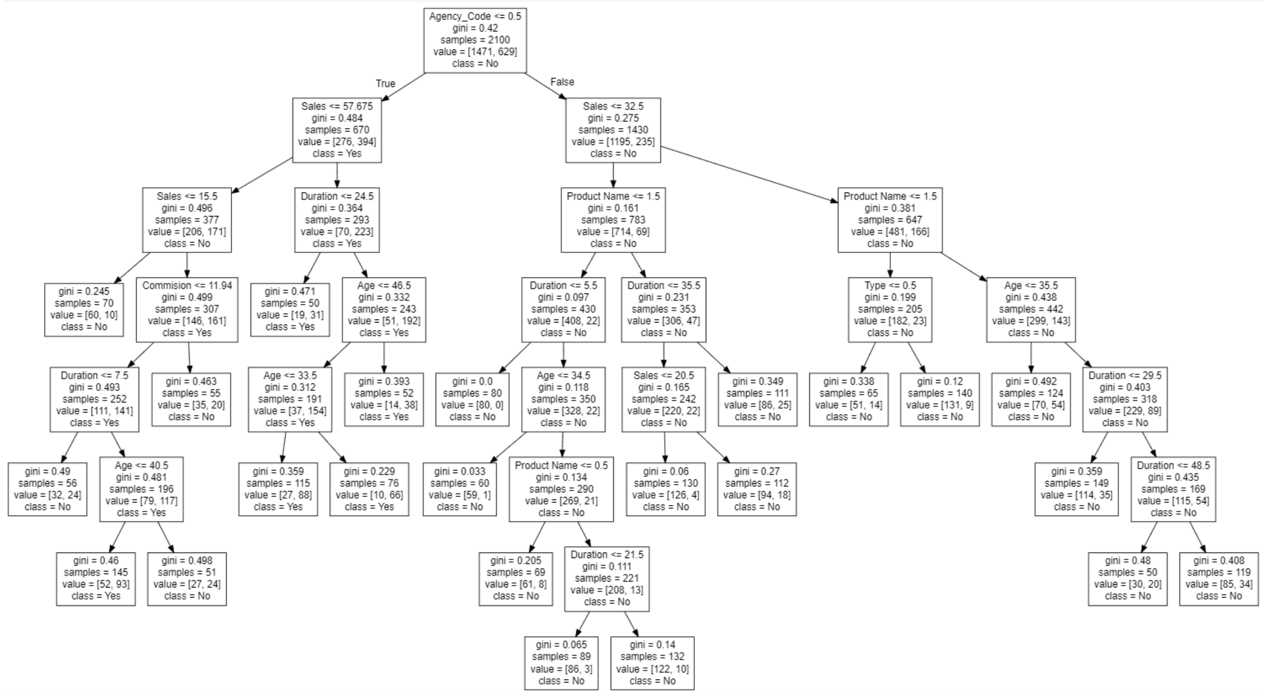
```
DecisionTreeClassifier
DecisionTreeClassifier(max_depth=10, min_samples_leaf=50, min_samples_split=150,
                       random_state=1)
```

Once we have used decision classifier for building a decision tree using the aforementioned parameters, we get the below decision tree. We then look into the **feature importance for x train** data wherein we get an output as **agency code to be highest as 0.599**.Once we have constructed the decision tree, we have calculated the AUC score for trains and test labels. We have also plotted ROC curve for train and test labels as to understand the area covered as shown below:

Figure 36. Importance Matrix

	Imp
Age	0.030261
Agency_Code	0.599363
Type	0.007416
Commision	0.012676
Channel	0.000000
Duration	0.037945
Sales	0.255785
Product Name	0.056555
Destination	0.000000

Figure 37. Decision Tree



Note: As we have a big decision tree, this is just a representation of how it looks and might not be legible.

Random Forest

For constructiong the randon forest, we have used random forest classifier with n_estimator equal to 500. Just like the decision tree, we have created a parameter grid to determine the best parameters for random forest using GridSearchCV as given below:

```
▸ GridSearchCV
▸ estimator: RandomForestClassifier
  ▸ RandomForestClassifier
```

```
{'max_depth': 5,
 'max_features': 5,
 'min_samples_leaf': 6,
 'min_samples_split': 45,
 'n_estimators': 450}
```

Artificial Neural Network (ANN)

We have used standard scalar and transformed the x train and x test data with following output:

X Train:

```
array([[-0.19192502,  0.72815922,  0.80520286, ..., -0.5730663 ,
         0.24642411, -0.43926017],
       [-0.19192502,  0.72815922,  0.80520286, ..., -0.26910565,
         0.24642411,  1.27851702],
       [-0.97188154, -1.28518425, -1.24192306, ...,  1.74601534,
         1.83381865, -0.43926017],
       ...,
       [-0.19192502,  0.72815922,  0.80520286, ...,  0.02103862,
         0.24642411, -0.43926017],
       [ 0.58803151,  1.73483096, -1.24192306, ..., -0.60069909,
        -1.34097044, -0.43926017],
       [-0.19192502, -1.28518425, -1.24192306, ..., -0.53852532,
         1.83381865, -0.43926017]])
```

X Test:

```
array([[-1.55684893, -0.27851251,  0.80520286, ...,  0.18683534,
        -1.34097044,  2.99629421],
       [ 1.66047173, -1.28518425, -1.24192306, ..., -0.48325974,
        -1.34097044, -0.43926017],
       [-0.87438698, -1.28518425, -1.24192306, ..., -0.62833187,
        -1.34097044, -0.43926017],
       ...,
       [-0.19192502, -1.28518425, -1.24192306, ..., -0.47635155,
        -1.34097044, -0.43926017],
       [ 1.07550434,  1.73483096, -1.24192306, ..., -0.43490237,
        -1.34097044, -0.43926017],
       [-0.28941958,  1.73483096, -1.24192306, ..., -0.49016794,
        -1.34097044, -0.43926017]])
```

For constructiong artificial neural network, we have used MLP classifier. Just like the decision tree and random forest, we have created a parameter grid to determine the best parameters for artificial neural network using GridSearchCV as given below:

```
MLPClassifier
MLPClassifier(hidden_layer_sizes=200, max_iter=1000, random_state=1, tol=0.01)
```

2.2.3 Performance Metrics: Comment and Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score, classification reports for each model.

2.2.3.1 Decision Tree

Train label AUC score: 0.836

Test label AUC score: 0.794

Figure 38. Train label ROC curve

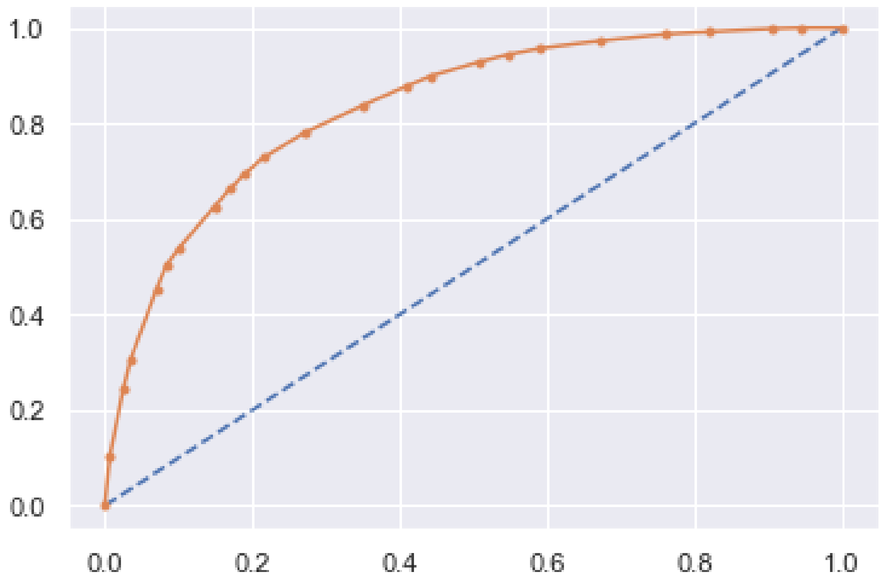
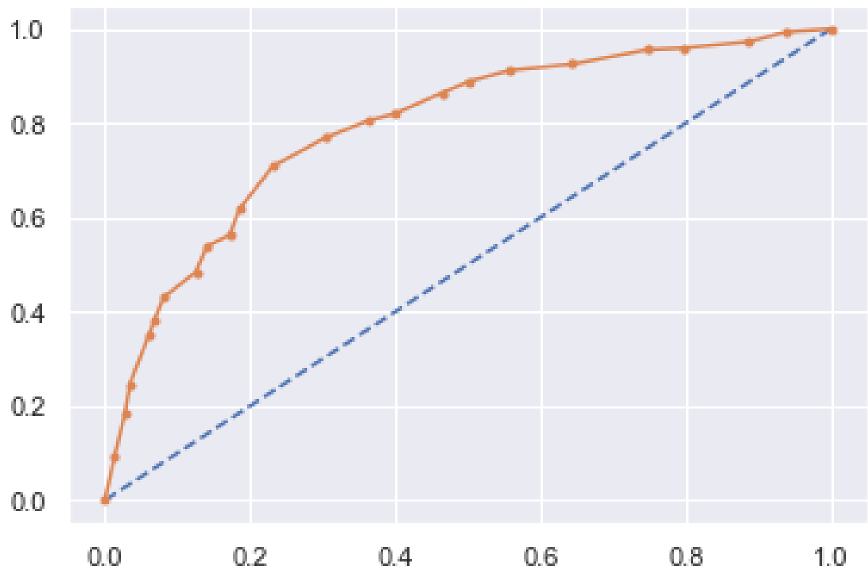


Figure 39. Test label ROC curve



Back to Top

2.2.3.1.1 Classification Report

Train Label:

	precision	recall	f1-score	support
0	0.81	0.92	0.86	1471
1	0.72	0.50	0.59	629
accuracy			0.79	2100
macro avg	0.77	0.71	0.73	2100
weighted avg	0.78	0.79	0.78	2100

Test Label:

	precision	recall	f1-score	support
0	0.76	0.93	0.83	605
1	0.73	0.38	0.50	295
accuracy			0.75	900
macro avg	0.74	0.66	0.67	900
weighted avg	0.75	0.75	0.72	900

2.2.3.1.2 Confusion Matrix

Train Label:

```
array([[1349,  122],
       [ 313,  316]], dtype=int64)
```

Test Label:

```
array([[564,  41],
       [183, 112]], dtype=int64)
```

2.2.3.2 Random Forest

Train label AUC score: 0.852

Test label AUC score: 0.820

Figure 40. Train label ROC curve

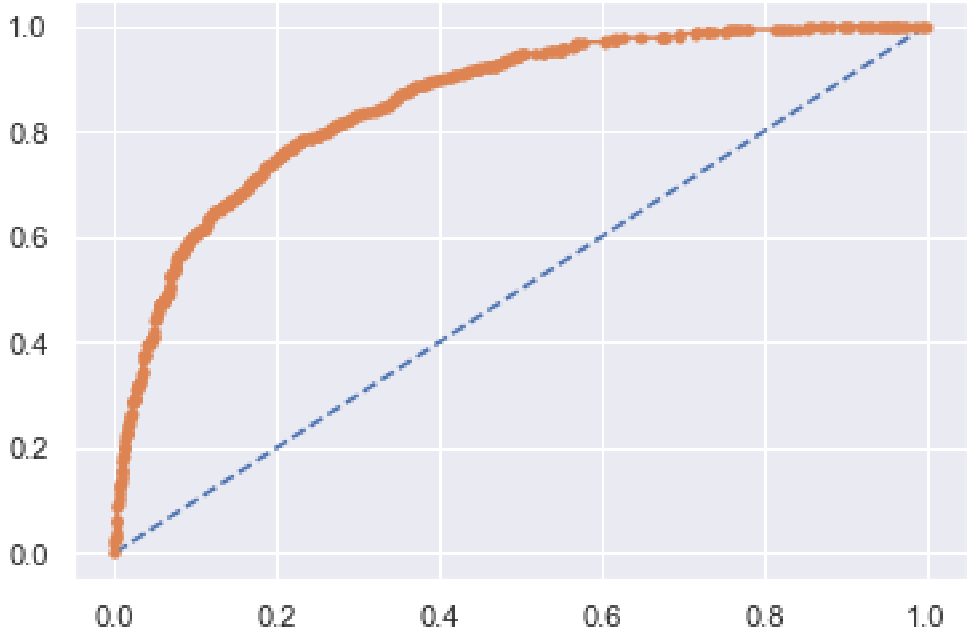
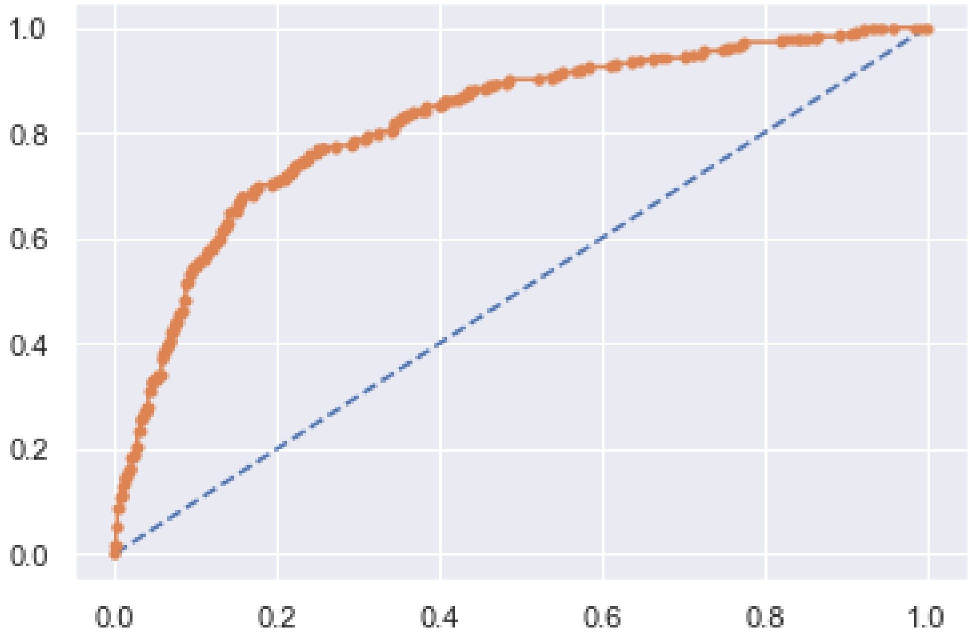


Figure 41. Test label ROC curve



Back to Top

2.2.3.2.1 Classification Report

Train Label:

	precision	recall	f1-score	support
0	0.84	0.89	0.87	1471
1	0.71	0.60	0.65	629
accuracy			0.81	2100
macro avg	0.77	0.75	0.76	2100
weighted avg	0.80	0.81	0.80	2100

Test Label:

	precision	recall	f1-score	support
0	0.79	0.92	0.85	605
1	0.74	0.49	0.59	295
accuracy			0.78	900
macro avg	0.76	0.71	0.72	900
weighted avg	0.77	0.78	0.76	900

2.2.3.2.2 Confusion Matrix

Train Label:

```
array([[1312,  159],
       [ 249,  380]], dtype=int64)
```

Test Label:

```
array([[554,  51],
       [149, 146]], dtype=int64)
```


Back to Top

2.2.3.3 Artificial Neural Network

Train label AUC score: 0.818

Test label AUC score: 0.804

Figure 42. Train label ROC curve

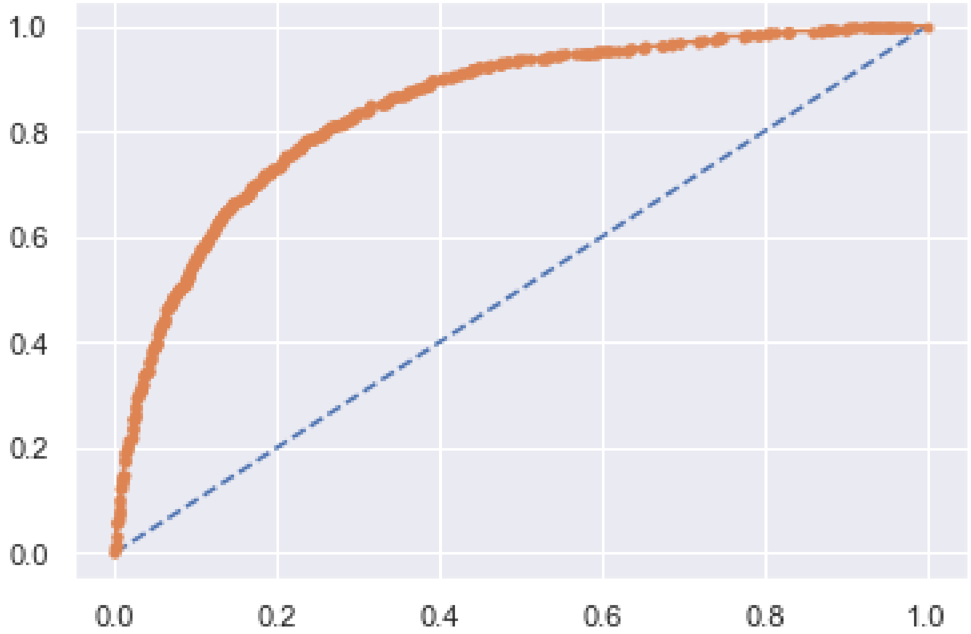
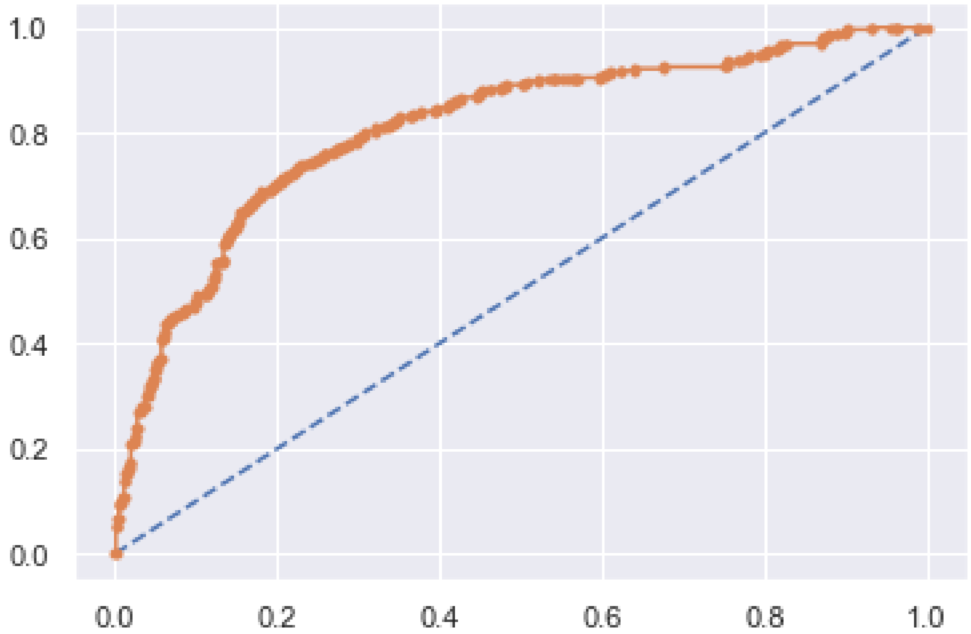


Figure 43. Test label ROC curve



Back to Top

2.2.3.3.1 Classification Report

Train Label:

	precision	recall	f1-score	support
0	0.81	0.89	0.85	1471
1	0.67	0.51	0.57	629
accuracy			0.78	2100
macro avg	0.74	0.70	0.71	2100
weighted avg	0.77	0.78	0.77	2100

Test Label:

	precision	recall	f1-score	support
0	0.77	0.92	0.84	605
1	0.72	0.43	0.54	295
accuracy			0.76	900
macro avg	0.75	0.68	0.69	900
weighted avg	0.75	0.76	0.74	900

2.2.3.3.2 Confusion Matrix

Train Label:

```
array([[1311,  160],
       [ 311,  318]], dtype=int64)
```

Test Label:

```
array([[556,  49],
       [167, 128]], dtype=int64)
```

2.2.4 Final Model: Compare all the models and write an inference which model is best/optimized.

Table 8 Comparative Analysis for Three Models

Scores	Dataset	Decision Tree		Random Forest		ANN	
		0	1	0	1	0	1
Accuracy	**Train**	0.79		0.81		0.78	
	Test	0.75		0.78		0.76	
Recall	**Train**	0.92	0.50	0.89	0.60	0.89	0.51
	Test	0.93	0.38	0.92	0.49	0.92	0.43
Precision	**Train**	0.81	0.72	0.84	0.71	0.81	0.67
	Test	0.76	0.73	0.79	0.74	0.77	0.72
F1 Score	**Train**	0.86	0.59	0.87	0.65	0.85	0.57
	Test	0.83	0.50	0.85	0.59	0.84	0.54
AUC Score	**Train**	0.836		0.852		0.818	
	Test	0.794		0.820		0.804	

Note: 0 = No
1 = Yes

Random forest will be the best model for the given dataset.

Explanation: Looking at the aforementioned table, we can see that the random forest model has better accuracy; and better recall, precision, and F1 score for test data. As recall is a ratio between true positives and false negatives, so recall value closer to 1 means depicts better model performance. In addition, as F1 score helps in classification of positives and negatives, higher F1 score means better model performance. As a result, we should consider random forest model for the given dataset.

2.2.5 Inference: Based on the whole Analysis, what are the business insights and recommendations

- We can see from the agency code, JZI has the least amount of sales. The company needs to find out a way to augment the sales from JZI by using promotional strategies and acquiring new customers.
- We can also see that majority of the sales are for customized plans but customers with silver and golden plans have higher number of claims. The company needs to find out the reason as to why the claims are so high with these customers and strategize to reduce the claim amount as silver and golden plans has higher coverage amount.
- We can see majority of the policies are sold through the travel agency; however, the amount of claims are high for policies sold by airlines. The company needs to analyze the patterns of claims and incidents to understand the reasons for the same.
- Around 10% of the policy sales are through offline channel; however, there is a higher percentage of claims from offline channel. The company needs to analyze the same and understand the pattern for the same.
- The company can see if they can reduce the cost of operations pertaining to the sale of policies which can improve the profit. In addition, they company can also reduce the claim cycle period which can reduce the number of claims again driving the profits.
- The company can also analyze the type of claims and improve the policy terms which can ultimately benefit the insurer and increase the overall profit

Back to Top