for technical audience

HOW TO PREVENT WASTEFUL SPENDING IN ONLINE RETAIL BUSINESS ?

Fatih Çağlar

5/23/2024

OUTLINE

Executive Summary

Introduction

Methodology

Results

Conclusion

Appendix

EXECUTIVE SUMMARY

- The organization records each sale through the online storefront in a database with various characteristics. We used this data to identify ways in which the business can cut down on wasteful spending in areas like product inventory, order fulfillment, and marketing.

- We have put the data through the ETL, analysis, and preprocessing phases. We have also created:
  - **A classification model** to predict whether a customer will be susceptible to churn.
  - **A regression model** to predict how much money a customer will spend at the online store.
  - **A clustering model** that divides customers into groups based on their shopping behavior.

EXECUTIVE SUMMARY

- After training and evaluating our models, we observed strong performance in both classification and regression tasks. Our classification model accurately predicts customer churn with an overall accuracy of 93%, correctly identifying 89% of customers who churn and 95% of customers who do not.
- For the monetary value prediction, our regression model explains 99.97% of the variability in customer spending. With an average customer spending of £152, the model achieves a prediction error of just £1.48, which is less than 1% of the mean spending.
- Our clustering model provides valuable insights into customer segmentation by grouping customers based on purchase frequency, monetary value, and recency. These insights enable targeted actions to retain the most valuable customers and increase spending among consistent buyers. Additionally, we identified the least spending customer segment and provided recommendations for either increasing their spending through segment-specific initiatives or reducing investment in this group to lower overall company costs.

INTRODUCTION

- We used stock descriptions and online retail history data for the ETL process and aggregation. After initial analyses, we cleaned the data, performed feature extraction, and conducted feature engineering to prepare it for training machine learning models. During this stage, we also obtained customer churn and tenure data.
- We prepared training and testing sets, trained our models, and evaluated them. Based on the evaluation scores, we made iterative improvements to enhance model performance.

METHODOLOGY

- **During the ETL process**, we extracted data from the organization's database, aggregating stock descriptions and online retail history tables. We identified and fixed corrupted or unusable data and performed necessary data type conversions.
- **In the analysis phase**, we first focused on price data. Next, we examined correlations between all numeric variables and their distributions and analyzed time-series data. At the end of the analysis, we corrected any remaining data issues like outliers.
- **To prepare the dataset for the machine learning process**, we generated new features from the existing data and indexed them by customer IDs. We also integrated customer churn and tenure data obtained from the organization's database.

METHODOLOGY

- To create an optimal classification model for predicting customer churn, we set up a pipeline incorporating scaling, dimensionality reduction, and model selection. Using the halving grid search algorithm, we efficiently identified the best pipeline configuration, minimizing computation time. The model was then evaluated using several metrics, including the ROC curve, classification report, and learning curve, to ensure robust performance.
- To create the regression model for monetary value prediction we first bootstrapped to create multiple samples and removed the impactful outliers of data. Standard scaling is applied to the feature sets to standardize the data. Several regression models, including Dummy Regressor, Linear Regression, Random Forest Regressor, and XGBoost Regressor, are trained and evaluated using standard evaluation metrics. As before, we evaluated the model using different metrics. Hyperparameter tuning is conducted using both GridSearchCV and RandomizedSearchCV to optimize the model's performance further. The tuned models' predictions are evaluated and compared using the same metrics.

METHODOLOGY

- To cluster customer data, we followed a systematic process. First, we standardized the data using the StandardScaler to ensure uniformity in feature scales. Then, we applied the Elbow Method to determine the optimal number of clusters, settling on eight clusters. Next, we employed K-means clustering with the chosen number of clusters and evaluated the model using silhouette analysis. We assessed the clustering results by examining the distribution of customers across clusters and their summary statistics, particularly focusing on key metrics like recency, frequency, and monetary value. Lastly, we visualized the clusters in three dimensions using PCA, providing a comprehensive understanding of customer segmentation.

RESULTS

ETL

- Online retail history data – 15321 observations for 8 features:

	Invoice	StockCode	Quantity	InvoiceDate	Price	CustomerID	Country	TotalAmount
0	536365	85123A	6	2010-12-01 08:26:00	2.55	u1785	United Kingdom	15.30
1	536367	84879	32	2010-12-01 08:34:00	1.69	u13047	United Kingdom	54.08
2	536373	85123A	6	2010-12-01 09:02:00	2.55	u1785	United Kingdom	15.30
3	536375	85123A	6	2010-12-01 09:32:00	2.55	u1785	United Kingdom	15.30
4	536378	20725	10	2010-12-01 09:37:00	1.65	u14688	United Kingdom	16.50

- Stock Description data – 3952 observations for 2 features:

	StockCode	Description
0	10002	INFLATABLE POLITICAL GLOBE
1	10080	GROOVY CACTUS INFLATABLE
2	10120	DOGGY RUBBER
3	10123C	HEARTS WRAPPING TAPE
4	10124A	SPOTS ON RED BOOKCOVER TAPE

ETL

- Aggregated data by left-joining stocks to retails – 17032 observations for 9 features:

	Invoice	StockCode	Quantity	InvoiceDate	Price	CustomerID	Country	TotalAmount	Description
0	536365	85123A	6	2010-12-01 08:26:00	2.55	u1785	United Kingdom	15.30	CREAM HANGING HEART T-LIGHT HOLDER
1	536367	84879	32	2010-12-01 08:34:00	1.69	u13047	United Kingdom	54.08	ASSORTED COLOUR BIRD ORNAMENT
2	536373	85123A	6	2010-12-01 09:02:00	2.55	u1785	United Kingdom	15.30	CREAM HANGING HEART T-LIGHT HOLDER
3	536375	85123A	6	2010-12-01 09:32:00	2.55	u1785	United Kingdom	15.30	CREAM HANGING HEART T-LIGHT HOLDER
4	536378	20725	10	2010-12-01 09:37:00	1.65	u14688	United Kingdom	16.50	LUNCH BAG RED RETROSPOT

ETL

- 1711 records of unknown description and 115 duplicated records were removed.
- Date formats were corrected.
- We ended up with 15206 records.

ANALYSIS – INITIAL LOOKS

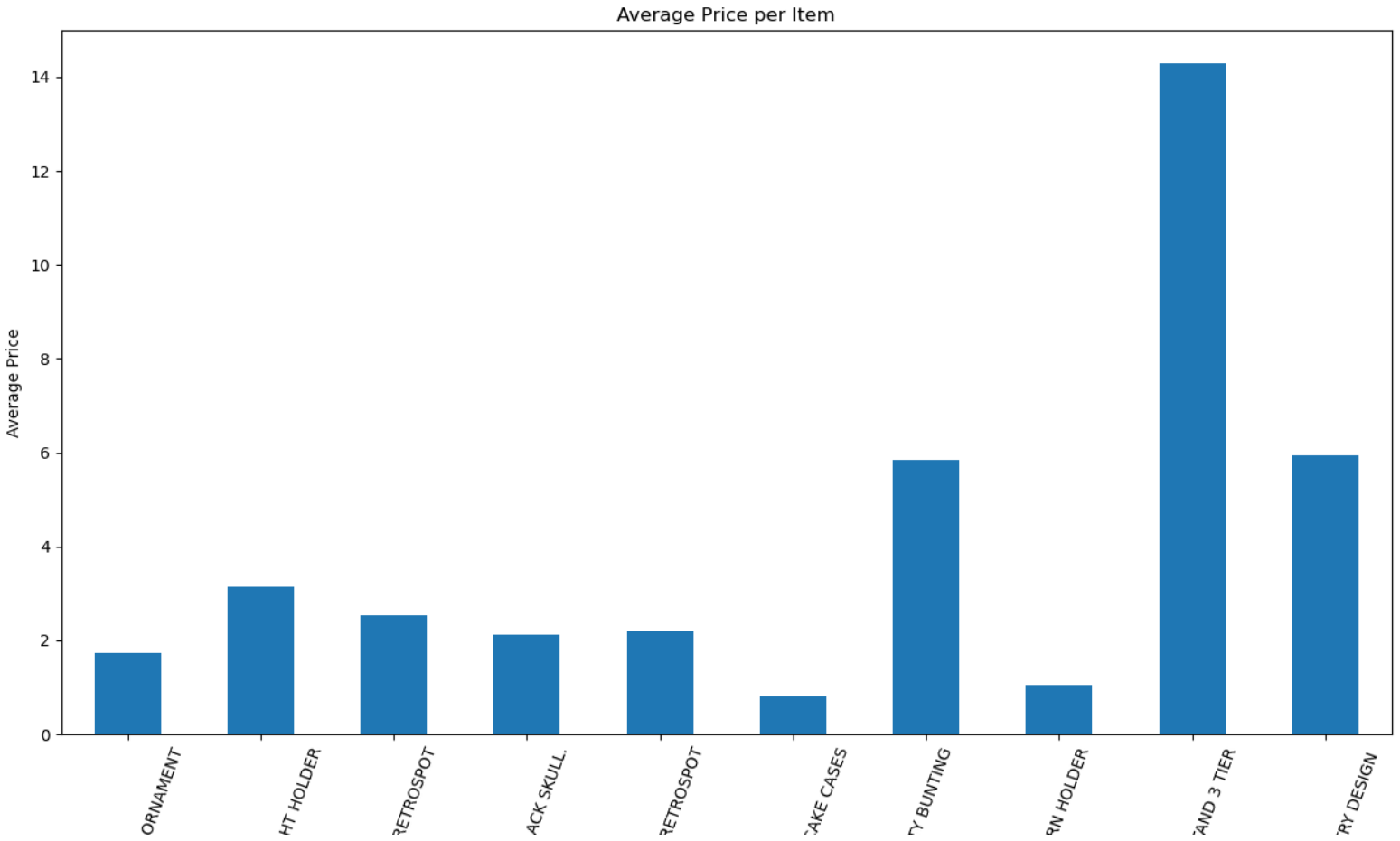
	Invoice	StockCode	InvoiceDate	CustomerID	Country	Description
count	15206	15206	15206	12435	15206	15206
unique	8315	10	305	2473	1	10
top	536876	85123A	2011-04-18	u17841	United Kingdom	CREAM HANGING HEART T-LIGHT HOLDER
freq	10	2163	100	171	15206	2163

- ✓ There are 10 different products sold between Dec 1, 2010 and Dec 9, 2011.
- ✓ There are 2473 different customers making purchases.
- ✓ Most purchased product is Cream Hanging Heart T- Light Hodler purchased 2163 times.
- ✓ There are 68 days at no one purchased anything.

ANALYSIS – INITIAL LOOKS

	Quantity	Price	TotalAmount
count	15206.000000	15194.000000	15194.000000
mean	16.775483	4.164267	40.705153
std	79.496270	4.377605	132.142503
min	1.000000	0.400000	0.550000
25%	2.000000	1.650000	8.850000
50%	6.000000	2.550000	16.500000
75%	12.000000	4.950000	30.360000
max	4300.000000	32.040000	4921.500000

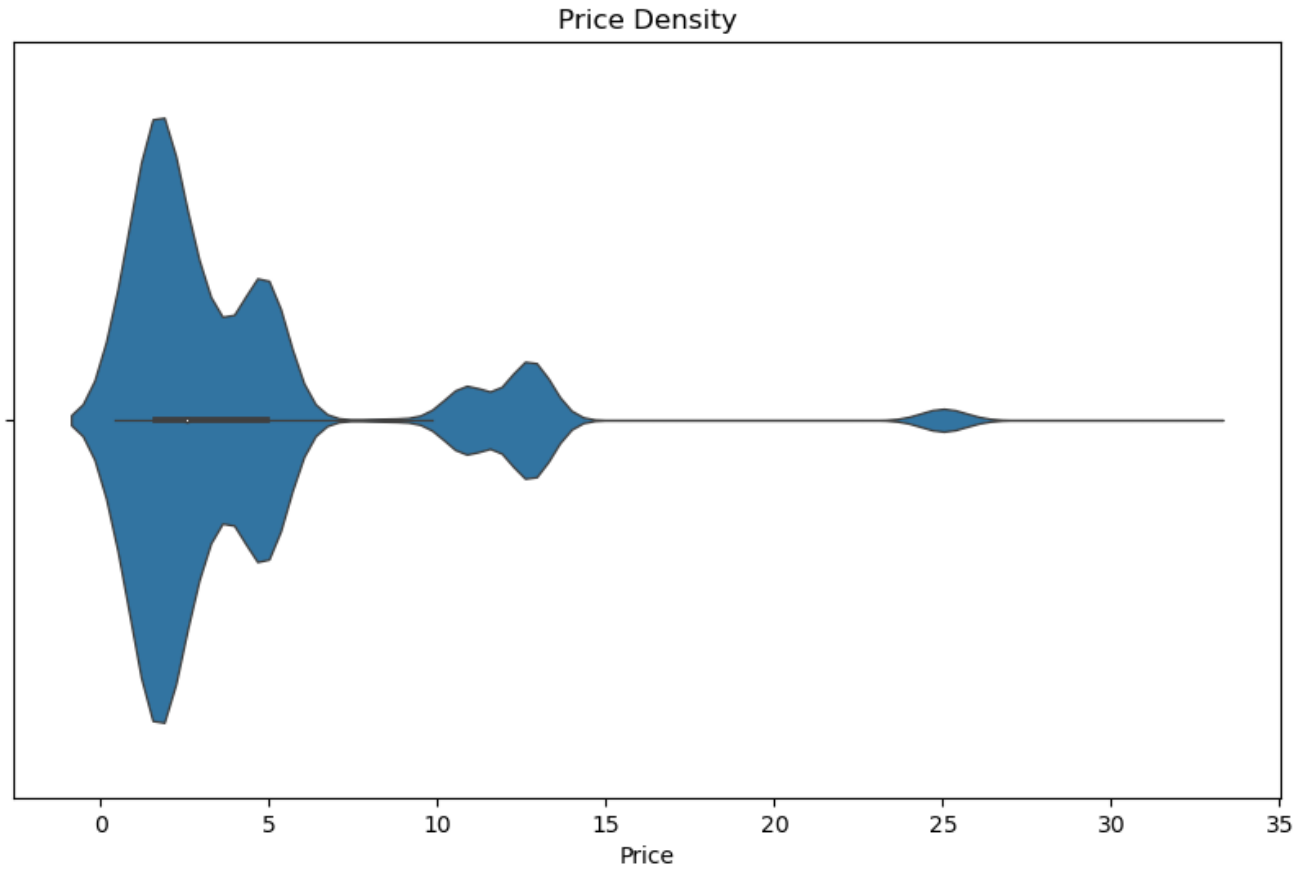
- The distributions for quantity, price and total amount are all right-skewed. The skewness decreases through quantity, total amount and price respectively.
- The standard deviations shows that there are many outliers for total amount and quantity.



- ✓ There are three items significantly more expensive than others.

ANALYSIS - INSIGHTS

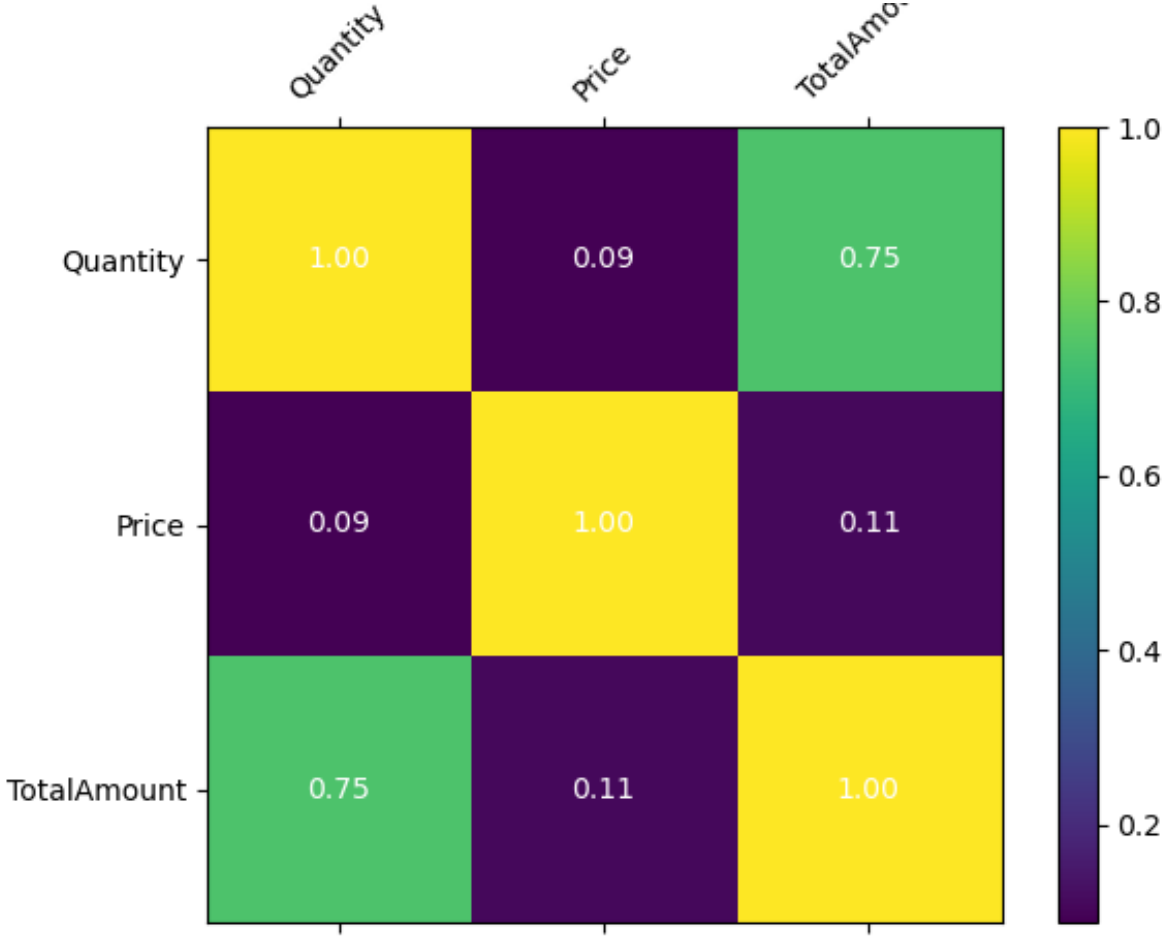
AVERAGE PRICE PER ITEM



- ✓ Price concentrates mainly on the range up to £7.5. The rest is around £10-£15 and £25.

ANALYSIS - INSIGHTS

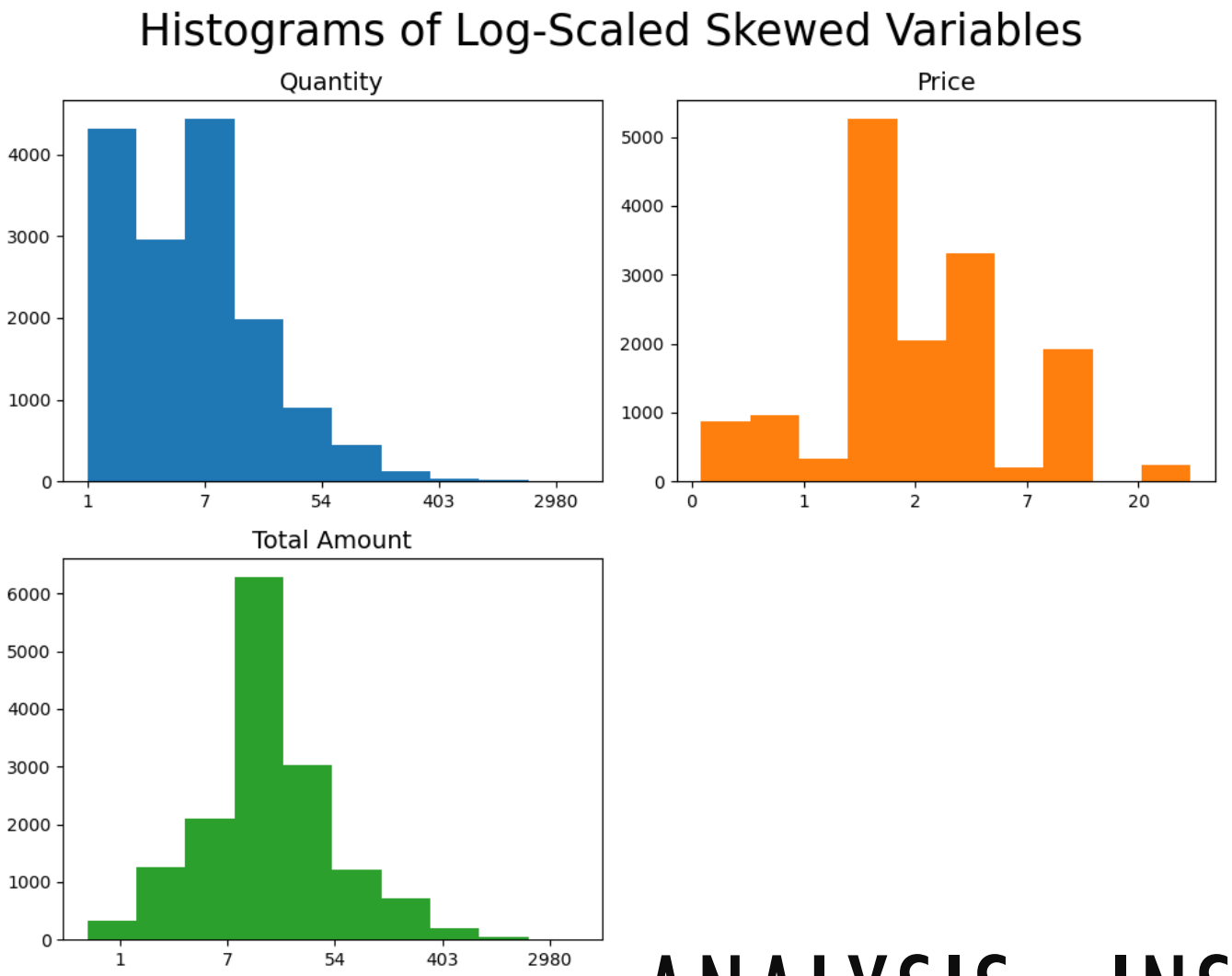
PRICE DENSITY



- ✓ There is a strong positive relationship between total amount and quantity. There is no any other correlation.

ANALYSIS - INSIGHTS

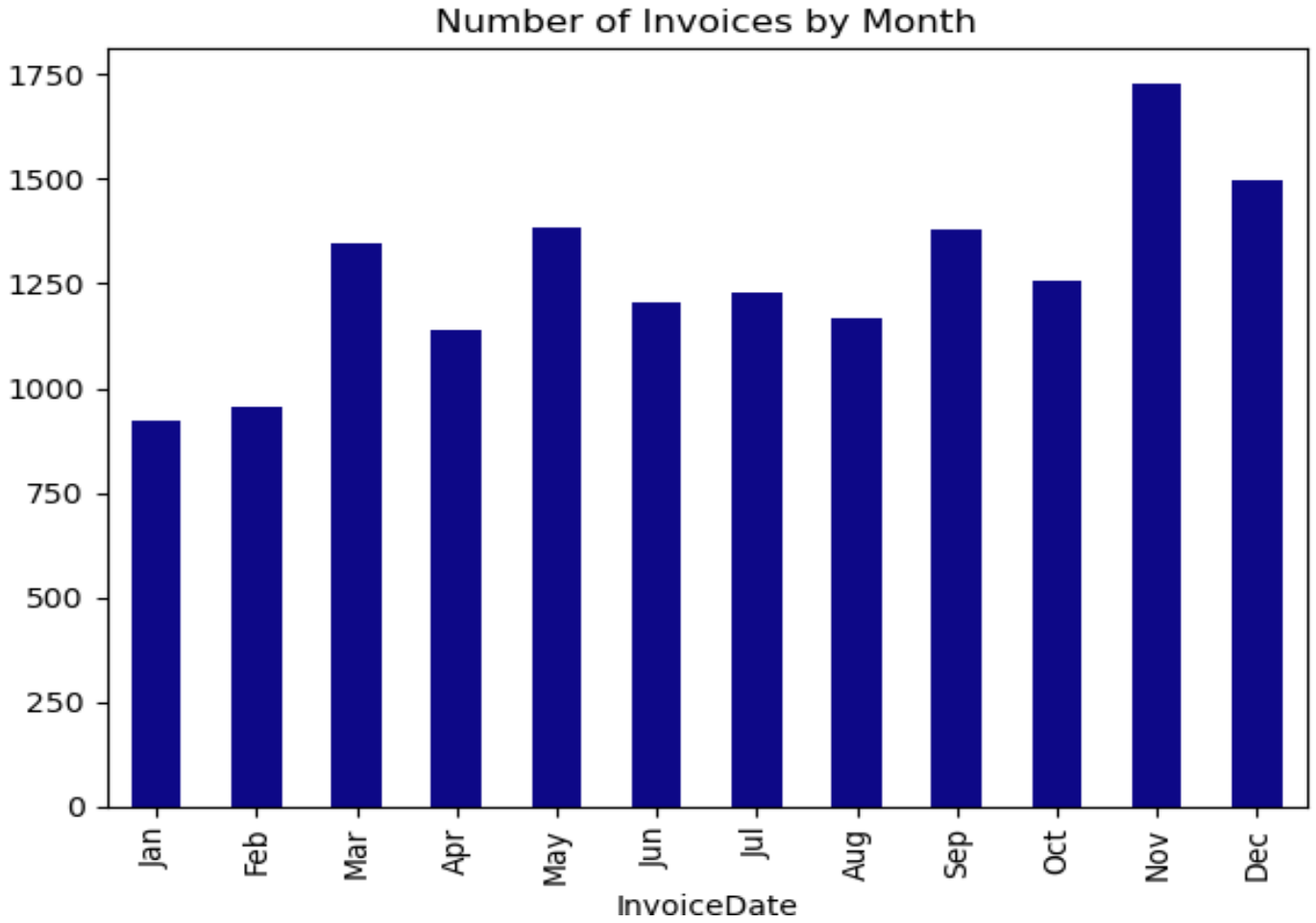
CORRELATION MATRIX



- ✓ Getting log-scaled, quantity is still highly right-skewed. Price is distributed moderately normal and total amount is distributed totally normal.

ANALYSIS - INSIGHTS

Using Log-Scale to Handle Skewness



- ✓ Highest numbers of invoices take place in November and December whereas the least takes place in January.

ANALYSIS - INSIGHTS

MONTHLY INVOICES

```
Percentages of Null Values
Invoice          0.0000
StockCode        0.0000
Quantity         0.0000
InvoiceDate      0.0000
Price            0.0789
CustomerID      18.2231
Country          0.0000
TotalAmount      0.0789
Description      0.0000
dtype: float64
```

- ✓ The columns with missing values are price, customer id and total amount and the percentages are 0.08%, 18% and 0.08%.
- ✓ The only clever way to impute customer ids is utilizing invoice numbers but there is no customer id info in remaining rows with same invoice number of rows with missing customer ids. Therefore, we dropped related rows.
- ✓ As the impact of the rows with missing price or total amount on analysis is very low, we filled them with 0.
- ✓ 12435 invoices remained.

```
[270]: missing_customer_invoices = online_history[online_history.CustomerID.isna()].Invoice.unique()
       online_history[(online_history.Invoice.isin(missing_customer_invoices)) & (~online_history.CustomerID.isna())]
```

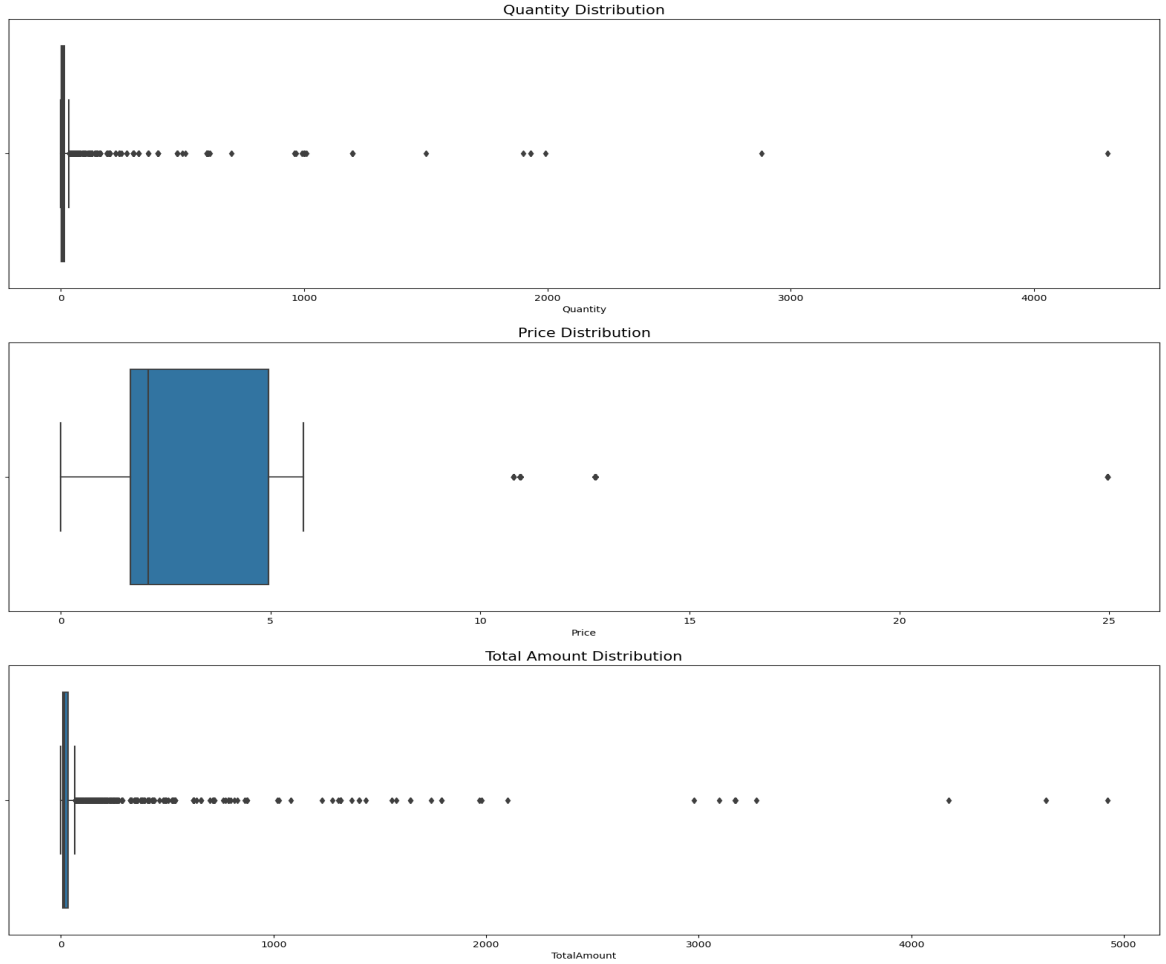
[270]:

Invoice	StockCode	Quantity	InvoiceDate	Price	CustomerID	Country	TotalAmount	Description

ANALYSIS - PREPROCESSING

HANDLING MISSING VALUES

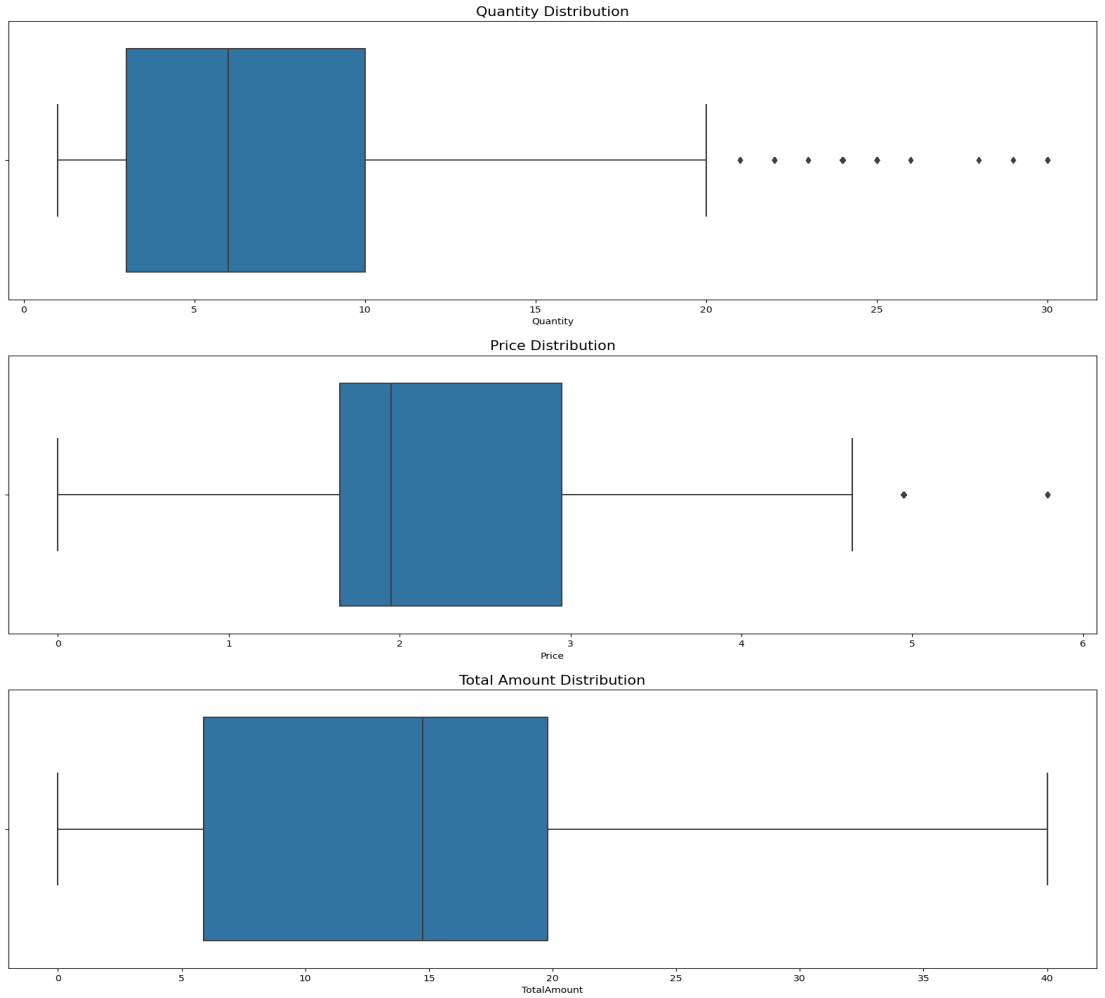
DISTRIBUTION OF Quantity, Price & Total Amount When Outliers Kept



ANALYSIS - PREPROCESSING | **REMOVING OUTLIERS**

DISTRIBUTION OF Quantity, Price & Total Amount When Outliers Removed

- ✓ 9133 rows remained.



- `frequency:` How often a customer makes purchase
- `recency:` The time elapsed since the customer's last purchase
- `tenure:` The length of time a customer has been with the company
- `monetary_value:` Total amount of money a customer has spent with the business
- `num_unique_items:` The diversity of the products a customer has purchased
- `churned:` Whether a customer has stopped doing business with the company

- ✓ To work with more meaningful features when training our models, we extracted new features from our data.
- ✓ We have taken the tenure and churn data from organization database.

```
freq_func = "count"
recency_func = lambda x: (dt.date(2011,12,10) - x.max()).days
#tenure data will be taken from organization database
monetary_value_func = "sum"
num_unique_items_func = "nunique"
#churn data will be taken from organization database
agg_funcs = {"Invoice":freq_func,"InvoiceDate":recency_func,"TotalAmount":monetary_value_func,"StockCode":num_unique_items_func}
```

```
tenure_churn = pd.read_pickle("../data/customer_data.pickle")[["tenure","churned"]]
```

```
customer_data = online_history.groupby("CustomerID")[["Invoice","InvoiceDate","TotalAmount","StockCode"]].agg(agg_funcs)
```

ANALYSIS - PREPROCESSING

FEATURE EXTRACTION

```
customer_data_preprocessed = customer_data_preprocessed.join(tenure_churn)
customer_data_preprocessed.columns = ["frequency","recency","monetary_value","num_unique_items","tenure","churned"]
```

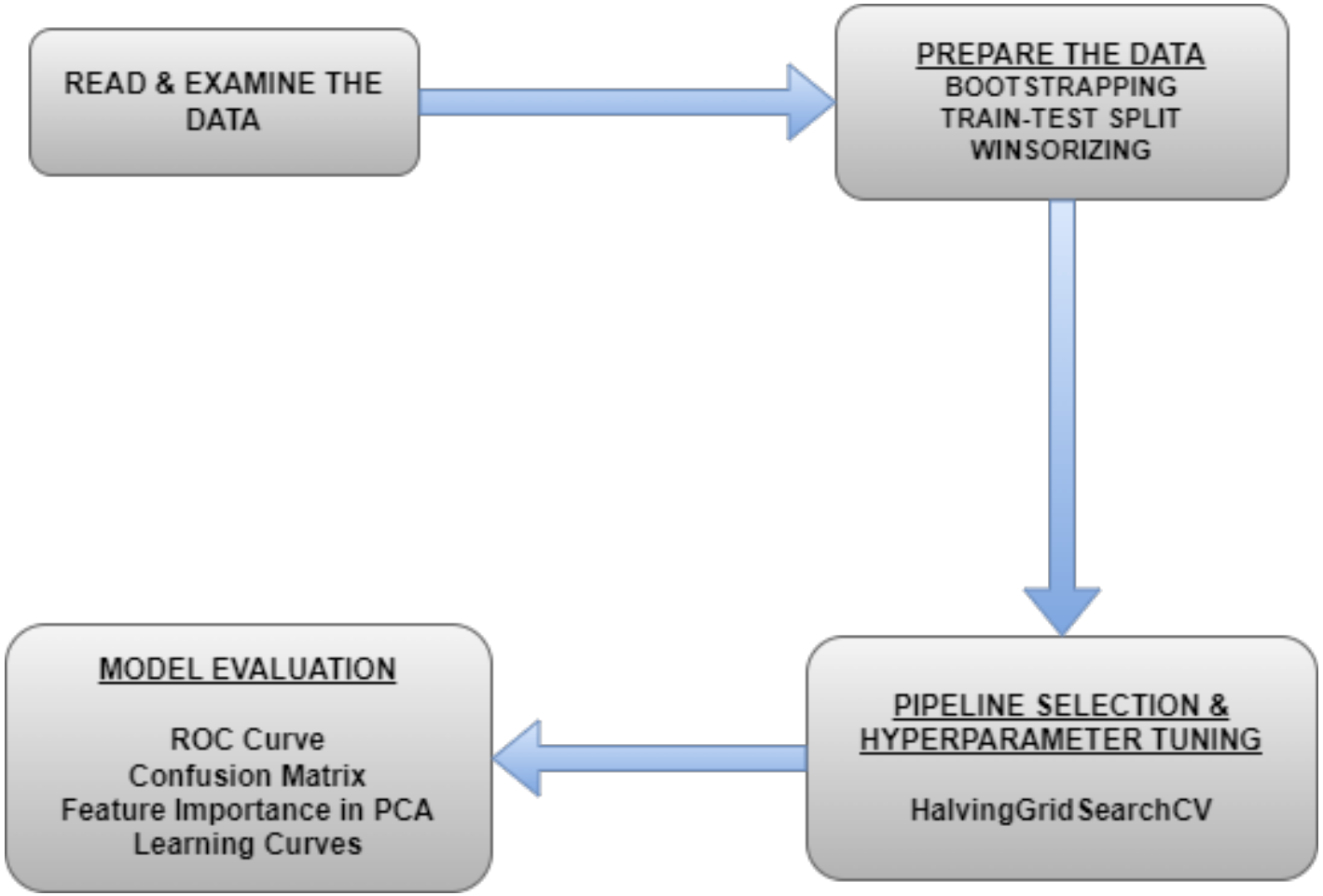
CustomerID	frequency	recency	monetary_value	num_unique_items	tenure	churned
u12747	11	3	302.76	3	369.0	1.0
u12748	95	2	783.19	10	369.0	0.0
u12749	6	4	251.42	3	130.0	1.0
u1282	1	327	17.70	1	326.0	0.0
u12822	1	88	16.50	1	87.0	1.0

```
<class 'pandas.core.frame.DataFrame'>
Index: 2130 entries, u12747 to u18283
Data columns (total 6 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   frequency         2130 non-null   int64
 1   recency           2130 non-null   int64
 2   monetary_value    2130 non-null   float64
 3   num_unique_items  2130 non-null   int64
 4   tenure            2130 non-null   float64
 5   churned           2130 non-null   float64
dtypes: float64(3), int64(3)
memory usage: 116.5+ KB
```

ANALYSIS - PREPROCESSING

FEATURE EXTRACTION

CLASSIFICATION – FLOWCHART



	frequency	recency	monetary_value	num_unique_items	tenure	churned
count	2130.000000	2130.000000	2130.000000	2130.000000	2130.000000	2130.000000
mean	5.465258	107.795305	152.446244	2.532394	214.496244	0.352113
std	8.530990	100.039421	441.992190	1.645733	112.923328	0.477741
min	1.000000	1.000000	0.550000	1.000000	1.000000	0.000000
25%	1.000000	25.250000	18.515000	1.000000	110.000000	0.000000
50%	3.000000	72.000000	47.860000	2.000000	240.000000	0.000000
75%	6.000000	171.000000	130.440000	3.000000	311.000000	1.000000
max	171.000000	374.000000	10281.200000	10.000000	373.000000	1.000000

- ✓ Customer data has 2130 rows. Monetary value and frequency have so many outliers.
- ✓ The target data is imbalanced.

CLASSIFICATION

READ & EXAMINE THE DATA

```
customer_data = bootstrapped_df
customer_data.describe()
```

	frequency	recency	monetary_value	num_unique_items	tenure	churned
count	213000.000000	213000.000000	213000.000000	213000.000000	213000.000000	213000.000000
mean	5.504216	107.359995	154.119585	2.533653	214.481944	0.351742
std	8.603438	99.794488	450.426806	1.649888	112.950909	0.477515
min	1.000000	1.000000	0.550000	1.000000	1.000000	0.000000
25%	1.000000	25.000000	18.750000	1.000000	110.000000	0.000000
50%	3.000000	72.000000	48.600000	2.000000	239.000000	0.000000
75%	6.000000	170.000000	131.900000	3.000000	311.000000	1.000000
max	171.000000	374.000000	10281.200000	10.000000	373.000000	1.000000

- ✓ Bootsrapping increased the number of rows to 213000.

CLASSIFICATION | BOOTSTRAPPING

```
X_train, X_test, y_train, y_test = \
train_test_split(features, \
target, test_size=0.35, random_state=42)
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

```
((138450, 5), (74550, 5), (138450,), (74550,))
```

```
counter_before = Counter(y_train)
counter_before
```

```
Counter({0.0: 89743, 1.0: 48707})
```

```
round(counter_before[1]/counter_before[0],2)
```

```
0.54
```

- ✓ We used 0.35 test size.
- ✓ The imbalance ratio was 0.5 before oversampling. We slightly raised it to 0.7.
- ✓ We winsorized the frequency and monetary value columns whose outliers can have impact.

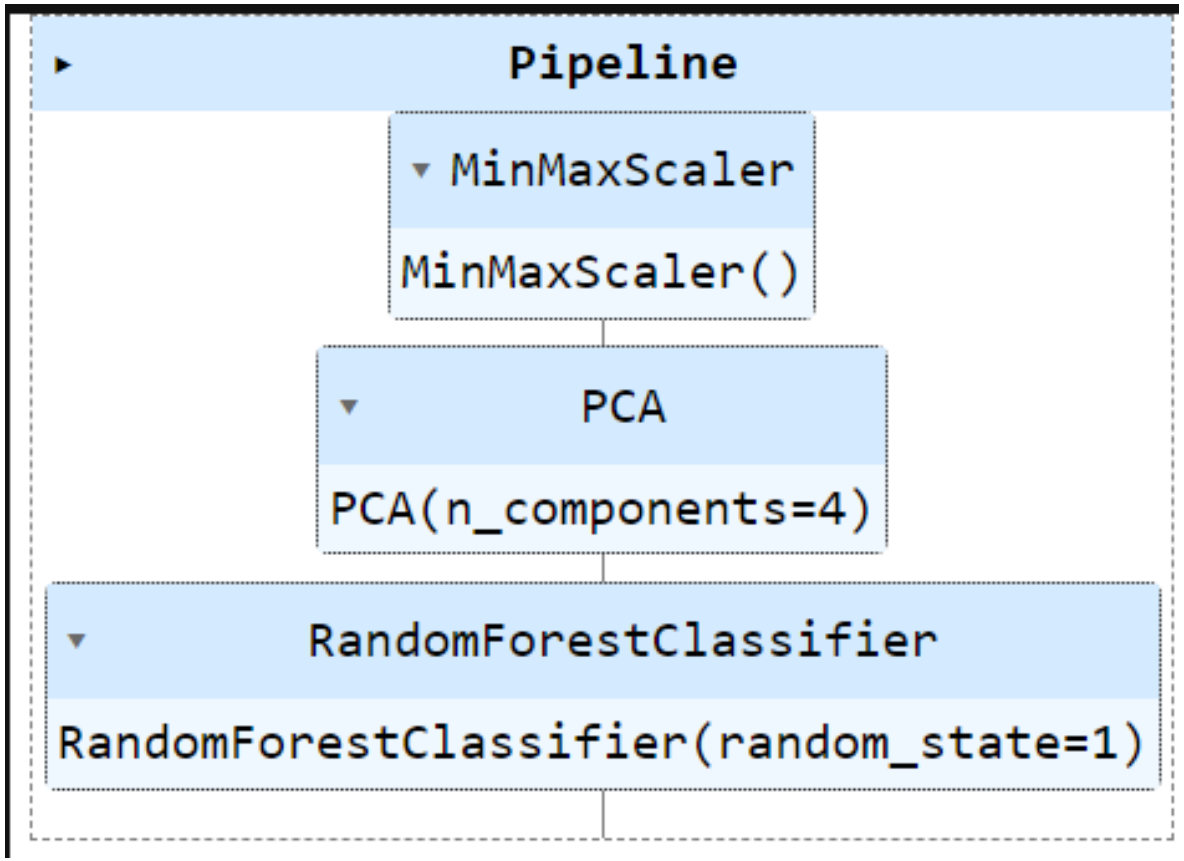
```
# Define oversampling strategy.
smote = SMOTE(sampling_strategy=0.7)
# Fit and apply the transform.
X_train,y_train = smote.fit_resample(X_train,y_train)
X_train = pd.DataFrame(X_train,columns=X_train.columns)
Counter(y_train)
```

```
Counter({0.0: 89743, 1.0: 62820})
```

```
customer_data["frequency"] = \
winsorize(customer_data['frequency'], limits=[0, 0.05])
customer_data["monetary_value"] = \
winsorize(customer_data['monetary_value'], limits=[0, 0.15])
```

CLASSIFICATION

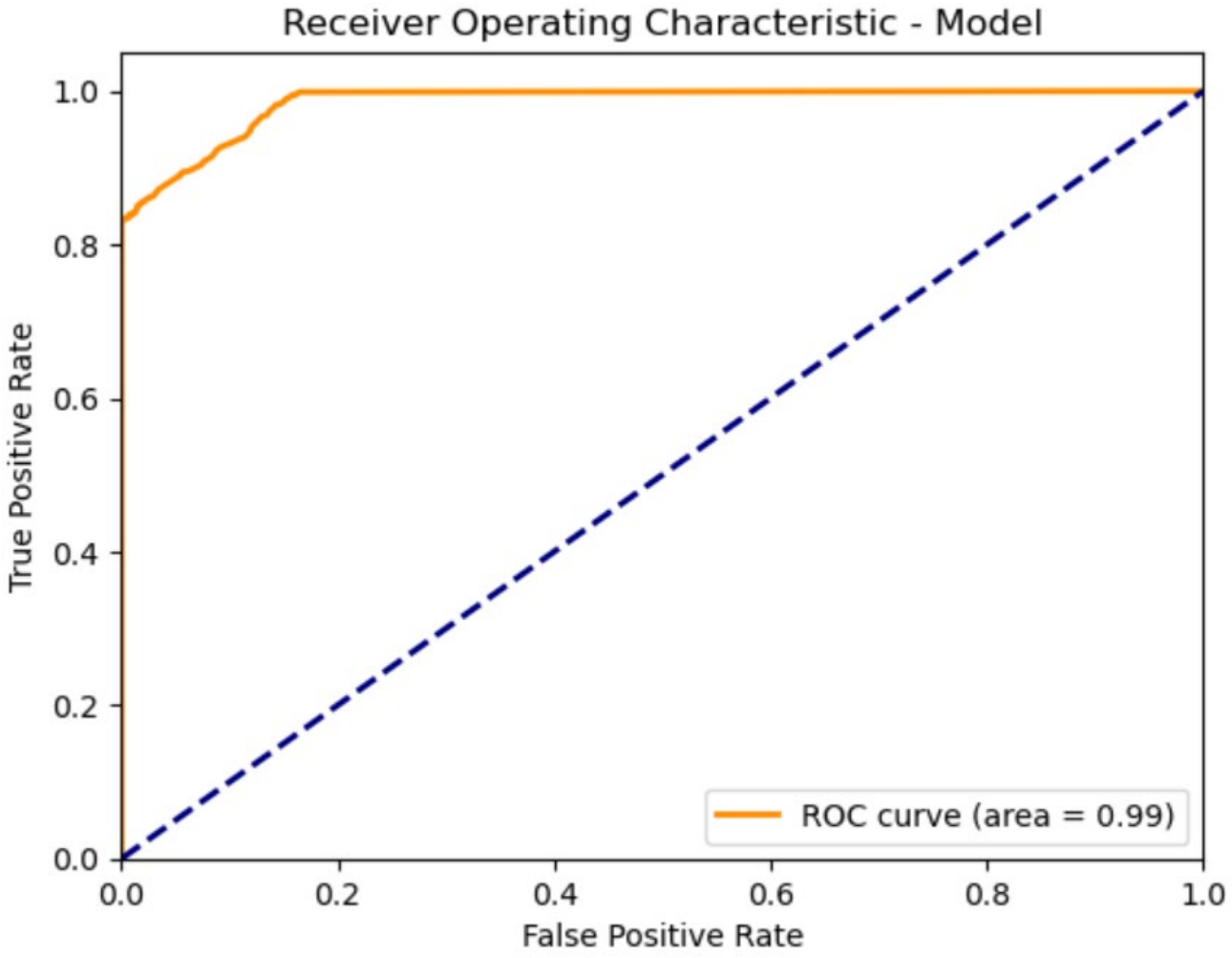
TRAIN-TEST SPLIT, OVERSAMPLING & WINSORIZING



- ✓ We conducted grid search with 2 different scalers, PCA and 3 different models, namely, XGB, DecisionTree and RandomForest. Passed the pipeline to the search with different hyperparameters.
- ✓ The picture shows the best model. Its r2 is 0.993.

CLASSIFICATION

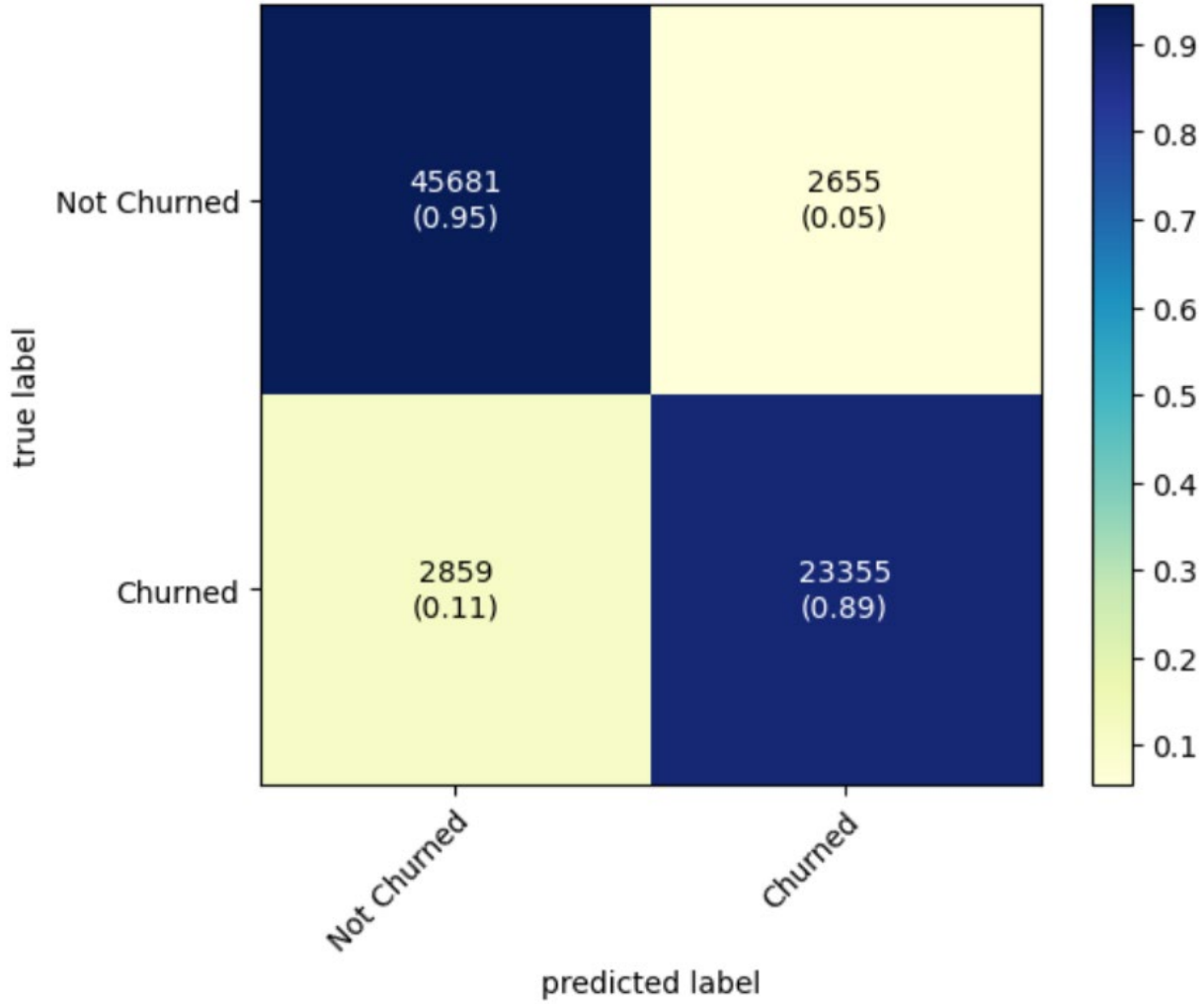
MODEL SEARCH & HYPERPARAMETER TUNING USING HALVINGGRIDSEARCHCV



- ✓ Our model performs near-perfect. It's AUC score is 0.99 and a classification threshold with a 0.1 FPR can be chosen.

CLASSIFICATION – EVALUATING THE MODEL

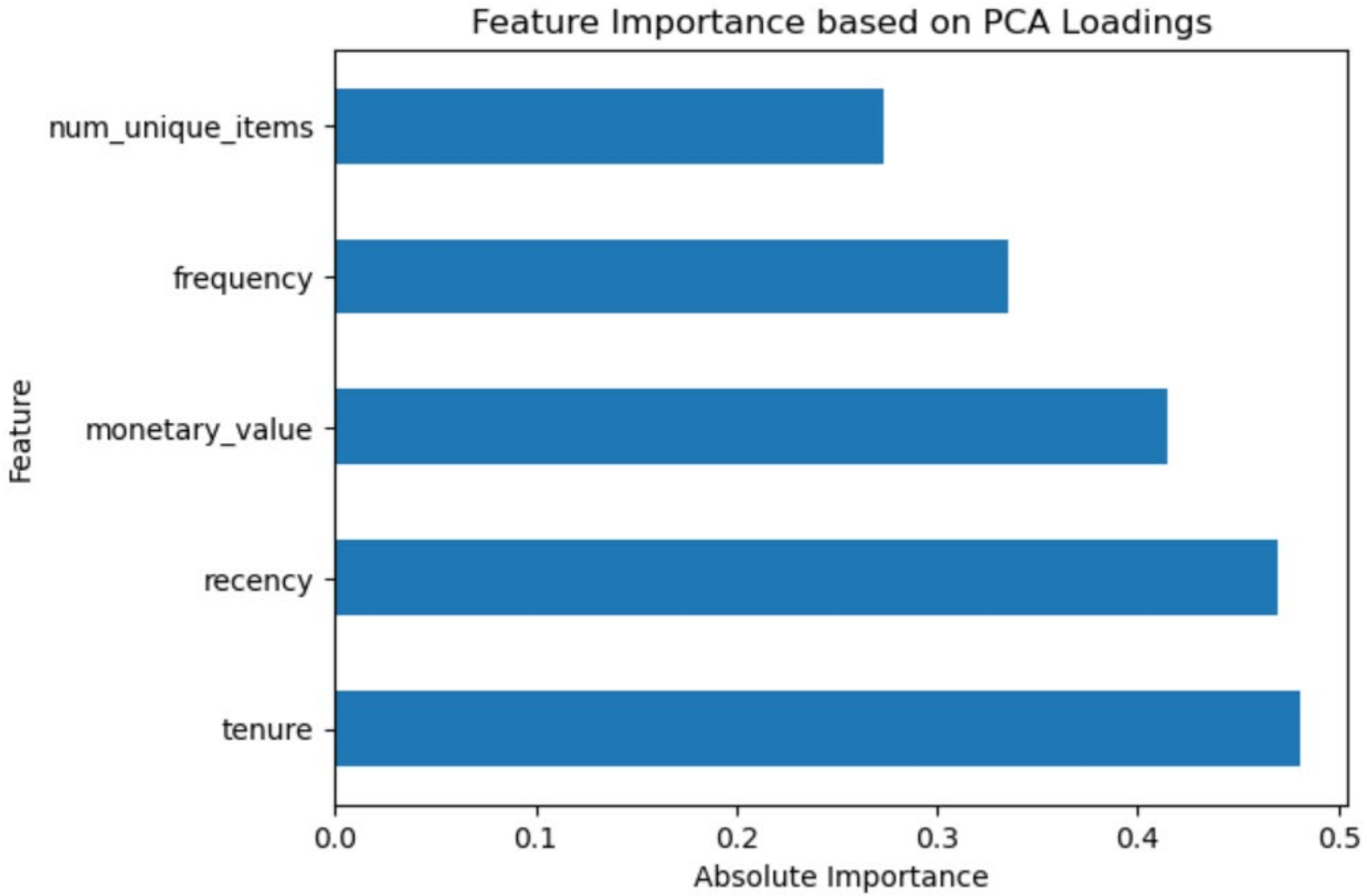
ROC CURVE



- ✓ Our model performs quite well on both cases. It correctly predicts 89% of customers churned and 95% of customers not churned.

CLASSIFICATION – EVALUATING THE MODEL

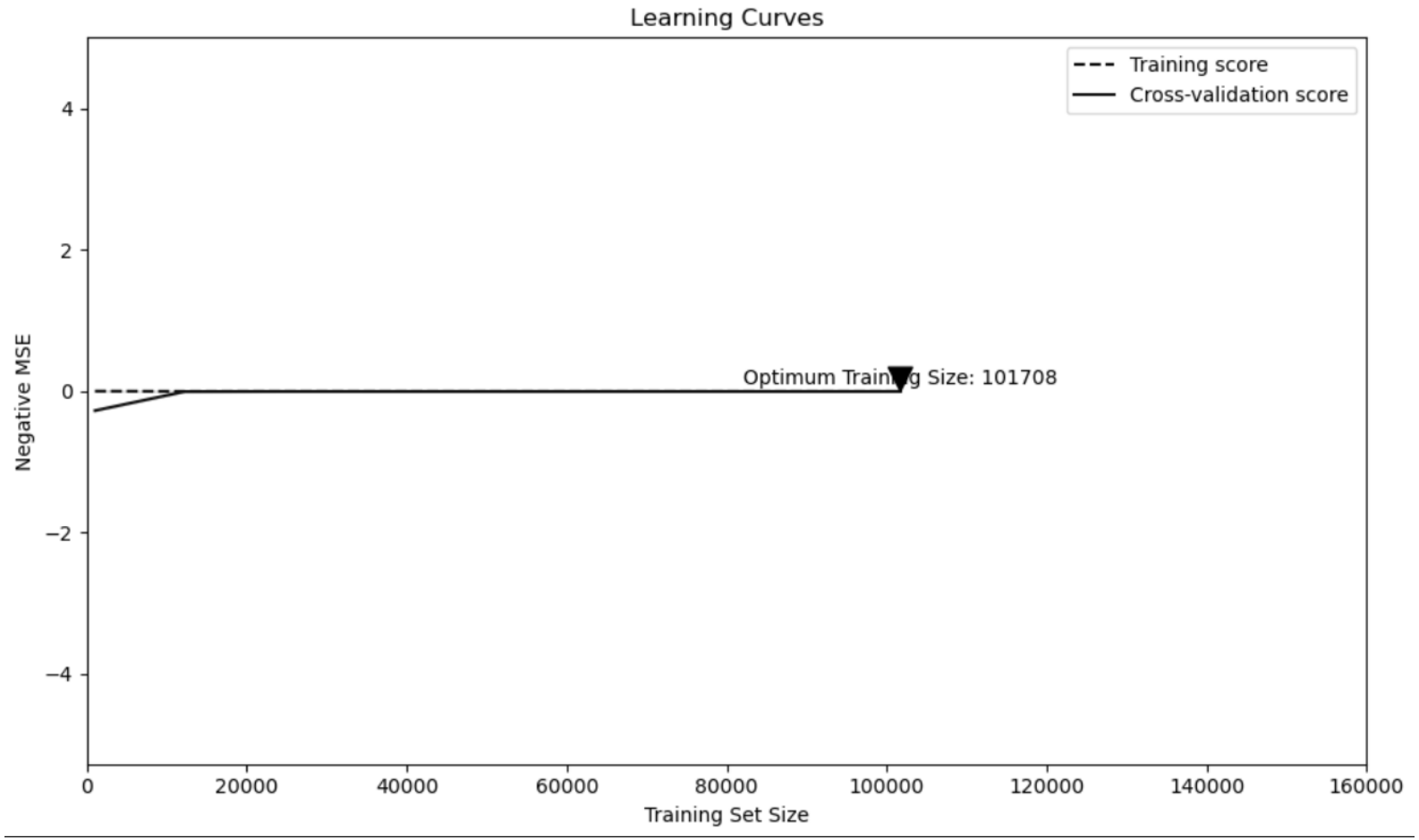
CONFUSION MATRIX



- ✓ The feature most contributing to PCA is tenure.

CLASSIFICATION – EVALUATING THE MODEL

PCA FEATURE IMPORTANCE



- ✓ The learning curves show that our model is a good fit and does have a suitable capacity for the complexity of the dataset.

CLASSIFICATION – EVALUATING THE MODEL

LEARNING CURVES

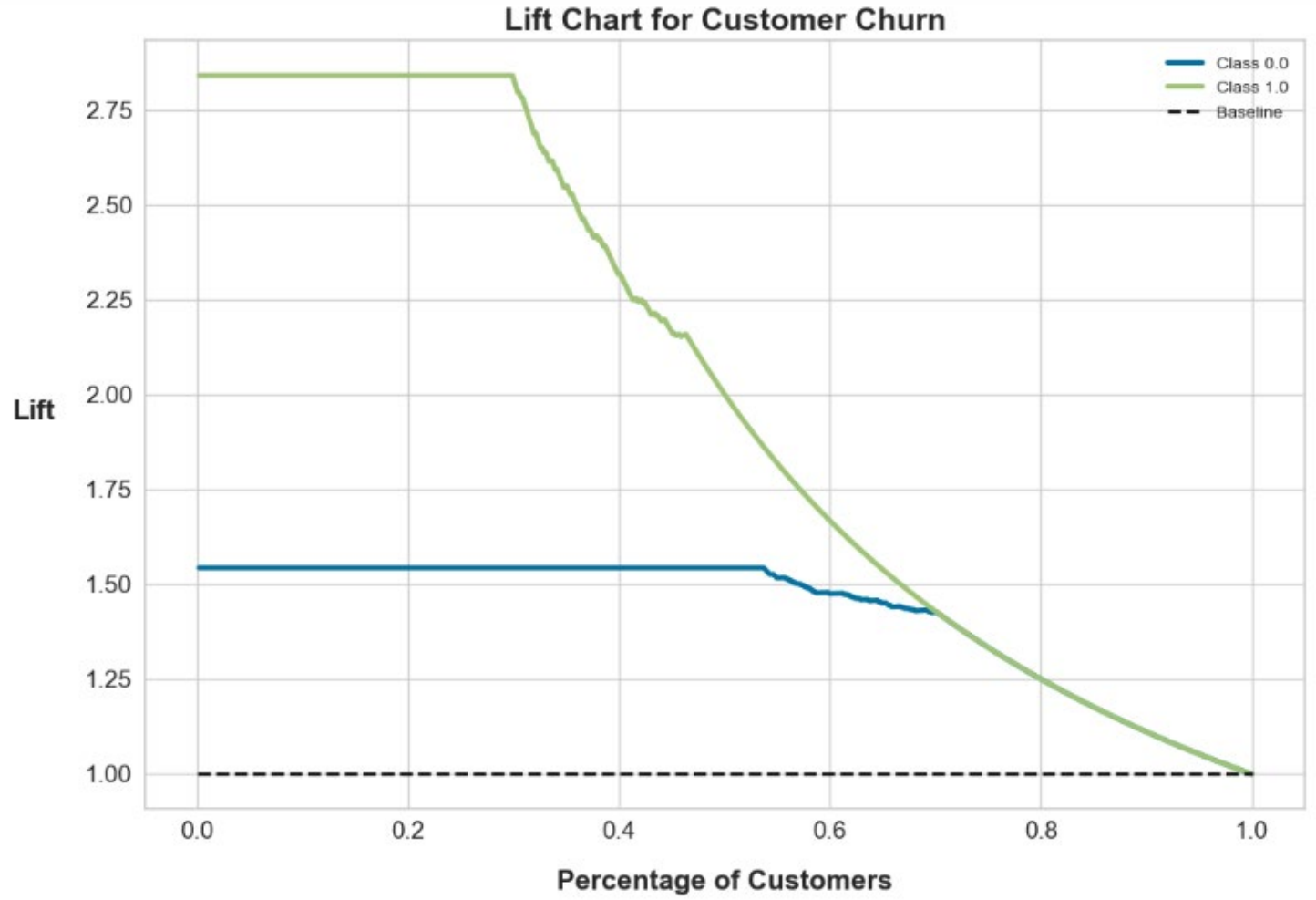
Accuracy is 0.9260362173038229

	Not churned	Churned	macro avg	weighted avg
precision	0.941100	0.897924	0.919512	0.925918
recall	0.945072	0.890936	0.918004	0.926036
f1-score	0.943082	0.894416	0.918749	0.925970
support	48336.000000	26214.000000	74550.000000	74550.000000

- ✓ Our model strongly performs at predicting churn.
- ✓ It is slightly better at predicting customers who will not churn.
- ✓ There is no precision-recall bias.
- ✓ The model can be used to develop new retention strategies.

CLASSIFICATION – EVALUATING THE MODEL

CLASSIFICATION REPORT

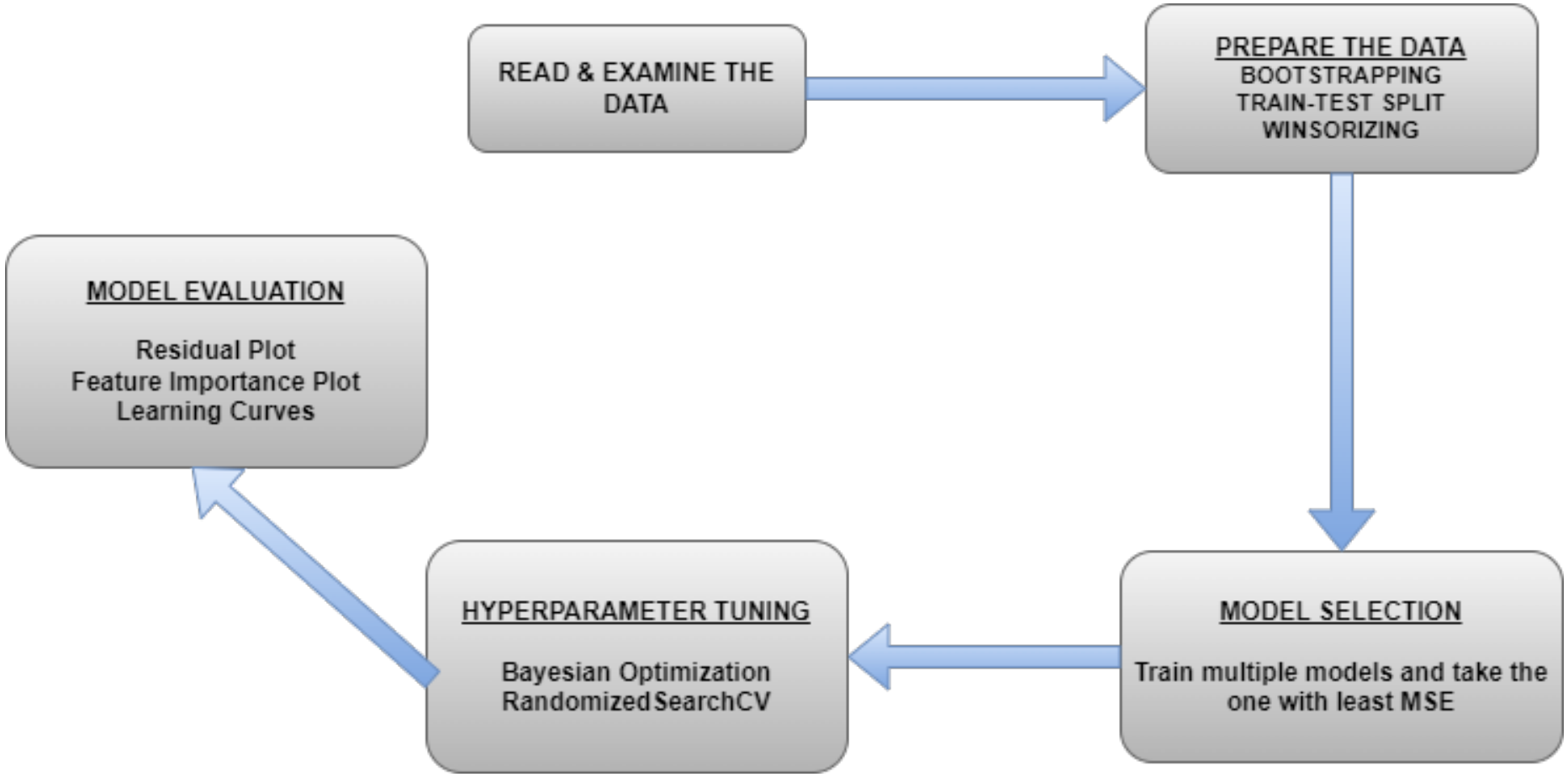


- ✓ Our model is almost 3 times better than random guessing at predicting positive cases ie. churn.

CLASSIFICATION – EVALUATING THE MODEL

LIFT CHART

REGRESSION – FLOWCHART



REGRESSION – FIRST STEPS

- ✓ We used the same data we used in the classification process and to prepare the data for the model, we did same bootstrapping. Since our target is **monetary_value** we only winsorized **frequency**. Did the split with test size 0.3. We scaled the data using StandardScaler for Logistic Regression.

	Model	R2	Mean Absolute Error	Mean Squared Error
2	Random Forest	0.9997	1.4800	4.945570e+01
3	XGBoost	0.9976	14.0884	4.607208e+02
0	Dummy Regressor	-0.0000	169.0752	1.896128e+05
1	Linear Regression	-593.1628	8084.2800	1.126565e+08

- ✓ We trained four different regression models and took the one with least MSE as the best.

REGRESSION

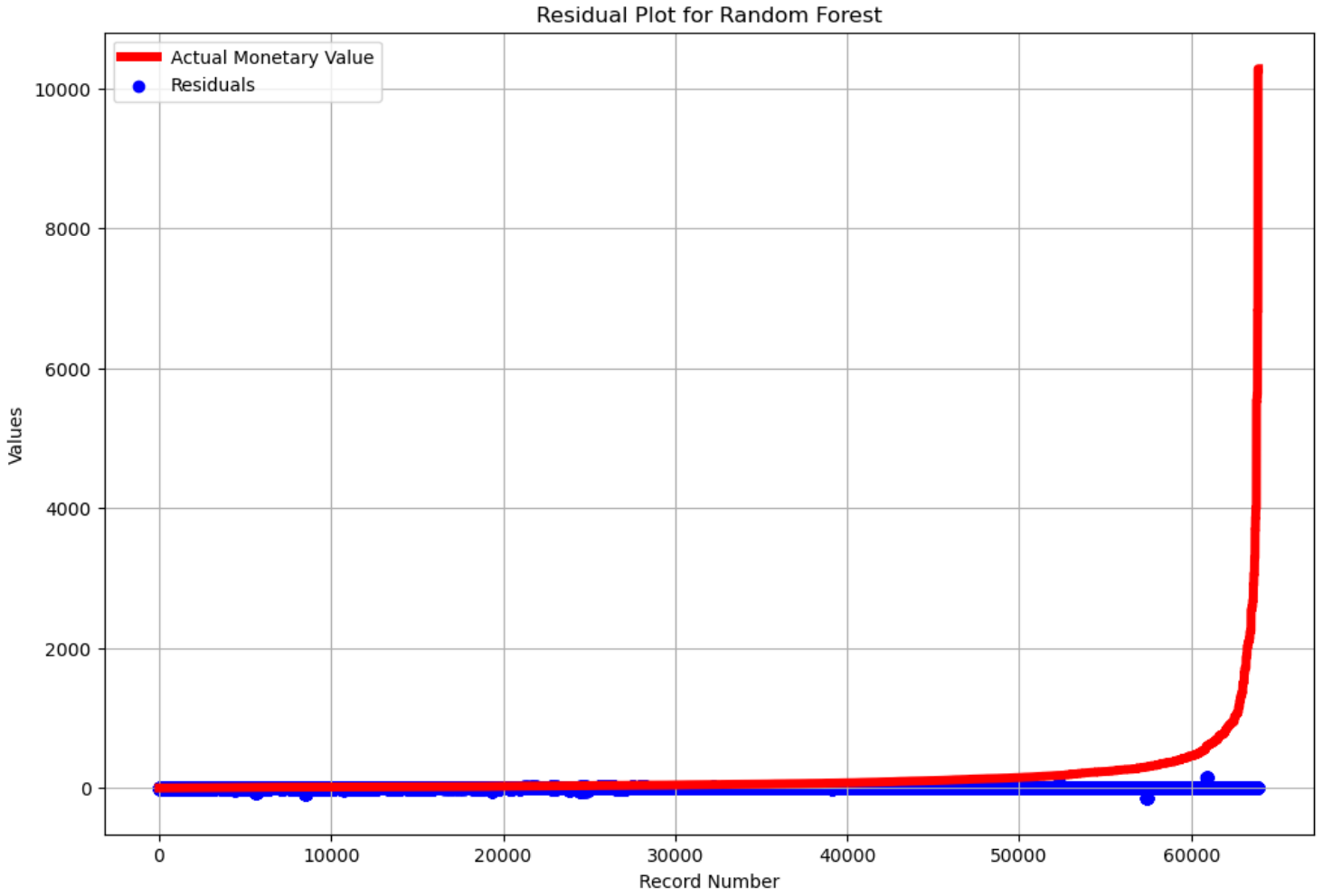
MODEL SELECTION

	Model	R2	Mean Absolute Error	Mean Squared Error
2	Random Forest	0.9997	1.4800	4.945570e+01
4	Random Forest with Optimum Test Size	0.9997	1.5001	5.350380e+01
5	Tuned Random Forest	0.9997	2.1958	6.166880e+01
6	Tuned XGBoost	0.9993	7.7199	1.489440e+02
3	XGBoost	0.9976	14.0884	4.607208e+02
0	Dummy Regressor	-0.0000	169.0752	1.896128e+05
1	Linear Regression	-593.1628	8084.2800	1.126565e+08

- ✓ We tuned the hyperparameters of RF with Bayesian Optimization using hyperopt and tuned those of XGB with RandomizedSearchCV but both didn't give any better results than RF. Therefore, we selected the RF as best.

REGRESSION

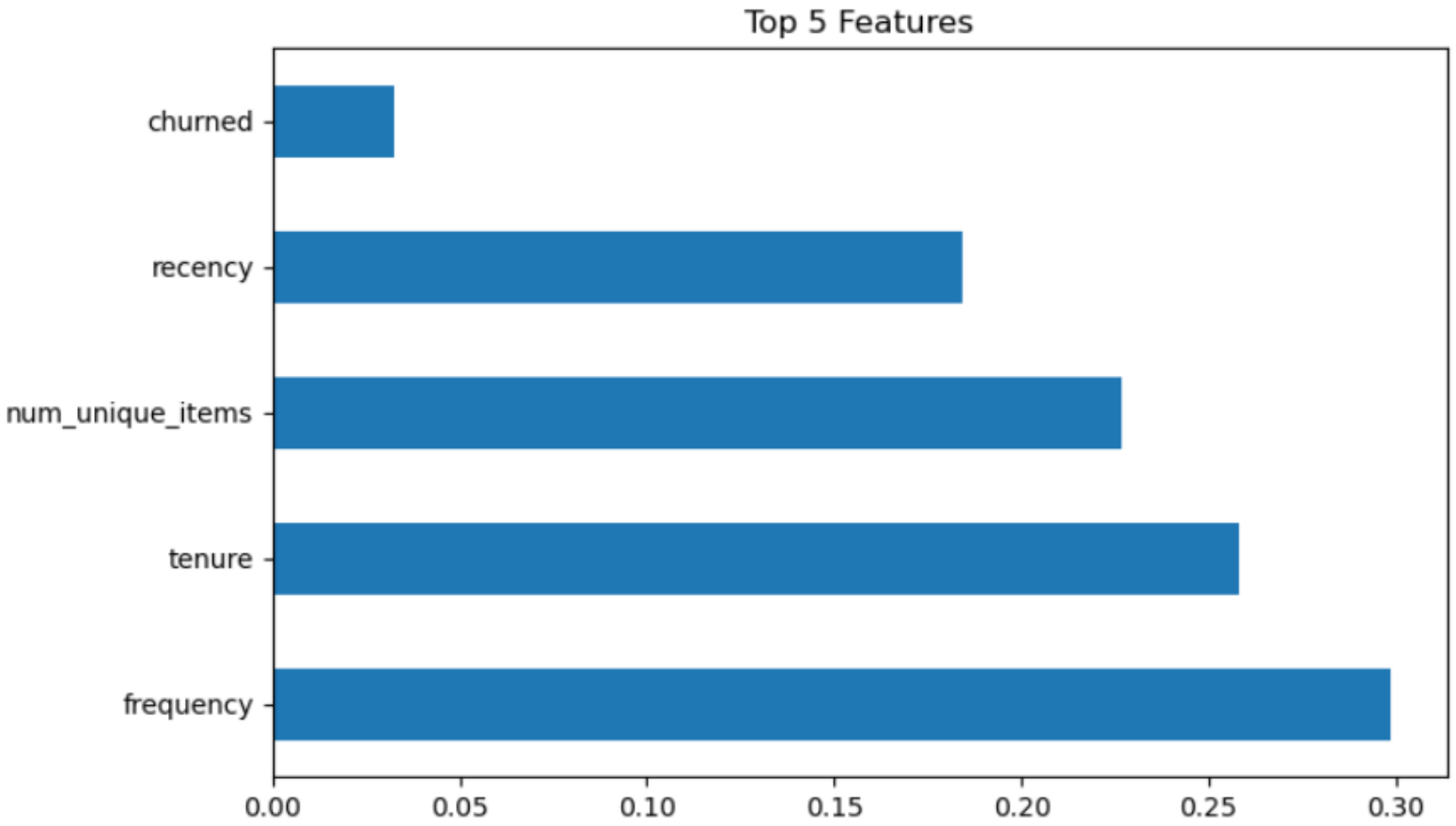
HYPERPARAMETER TUNING



- ✓ The random distribution of residuals along base line shows that our predictions fits well on test data with low MSE.

REGRESSION – EVALUATING BEST MODEL

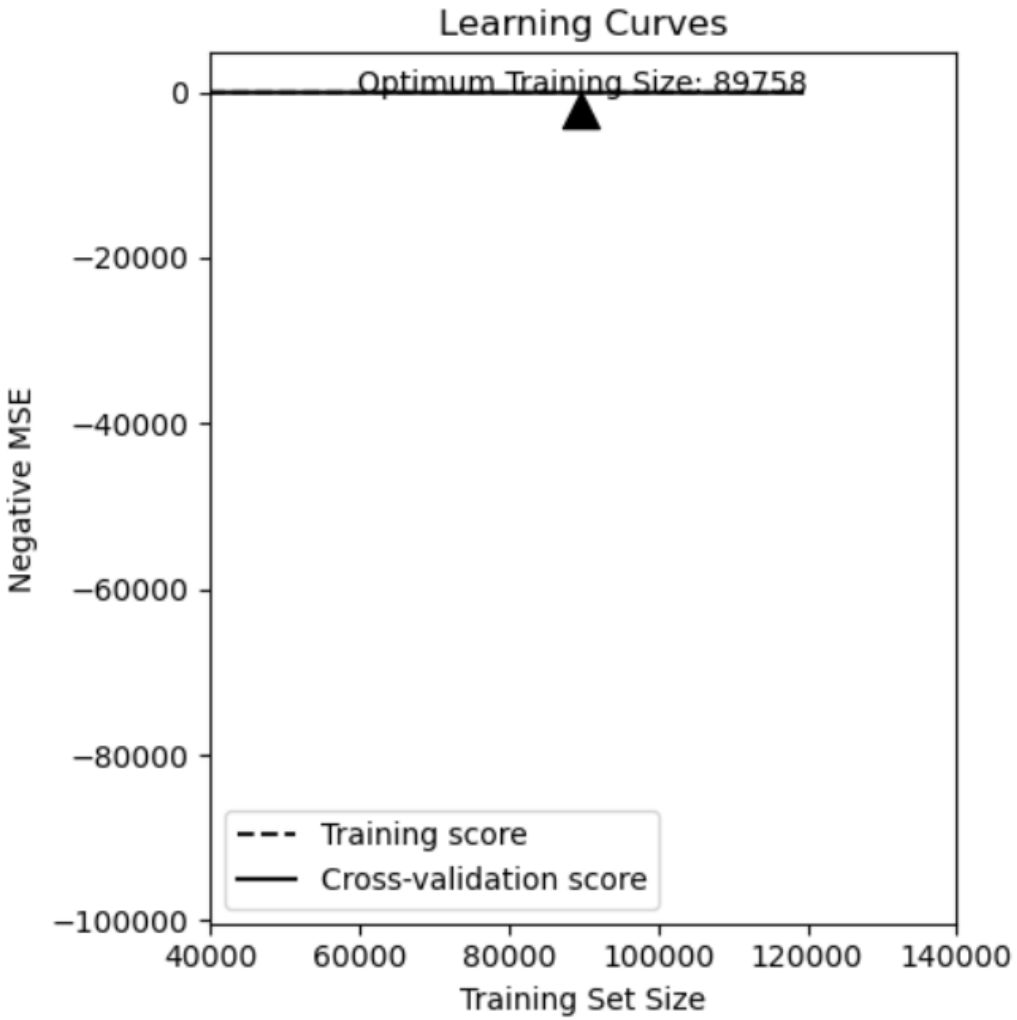
RESIDUAL PLOT



- ✓ Most important feature for regression task is frequency.

REGRESSION – EVALUATING BEST MODEL

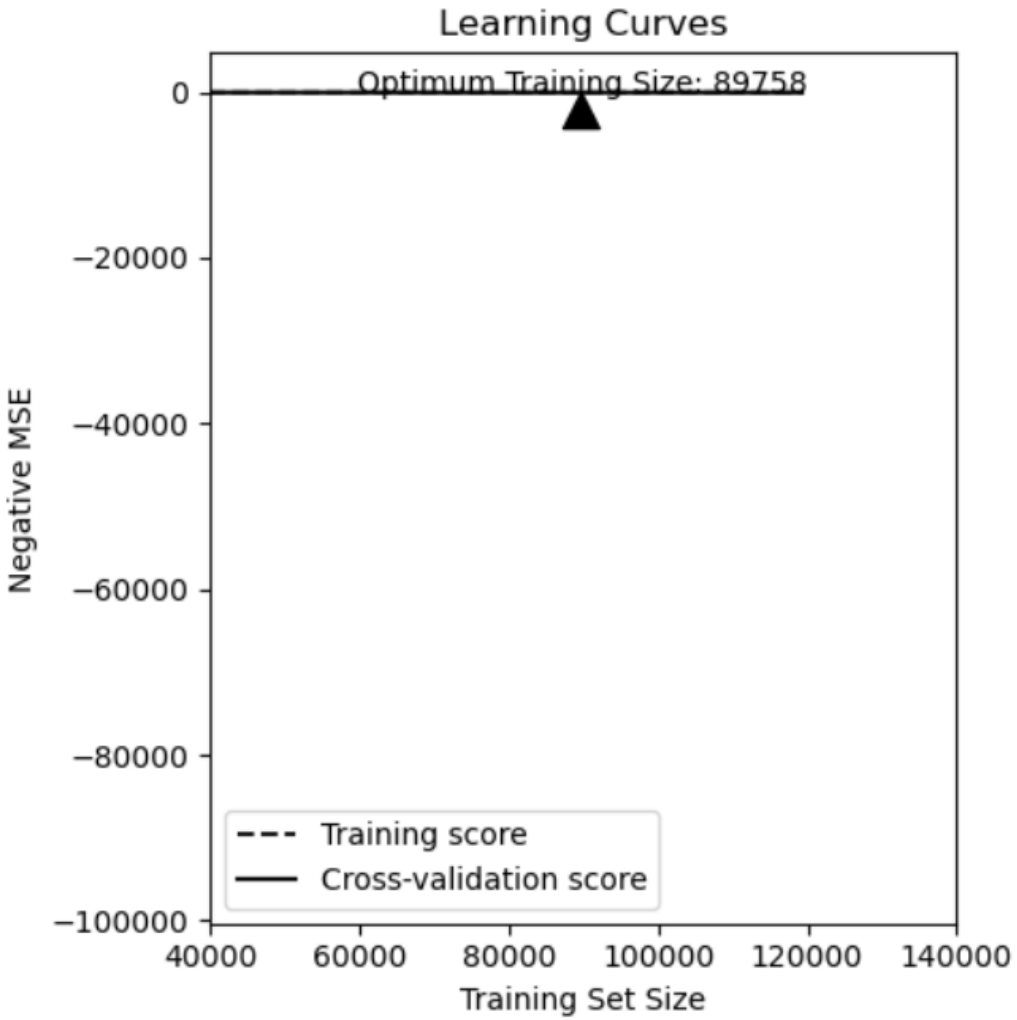
FEATURE IMPORTANCE PLOT



- ✓ Learning curves show that our training and cv scores fit well after some size of training data.

REGRESSION – EVALUATING BEST MODEL

LEARNING CURVES



- ✓ Learning curves show that our training and cv scores fit well after some size of training data.

REGRESSION – EVALUATING BEST MODEL

LEARNING CURVES

Model	R2	Mean Absolute Error	Mean Squared Error
Random Forest	0.9997	1.4800	4.945570e+01

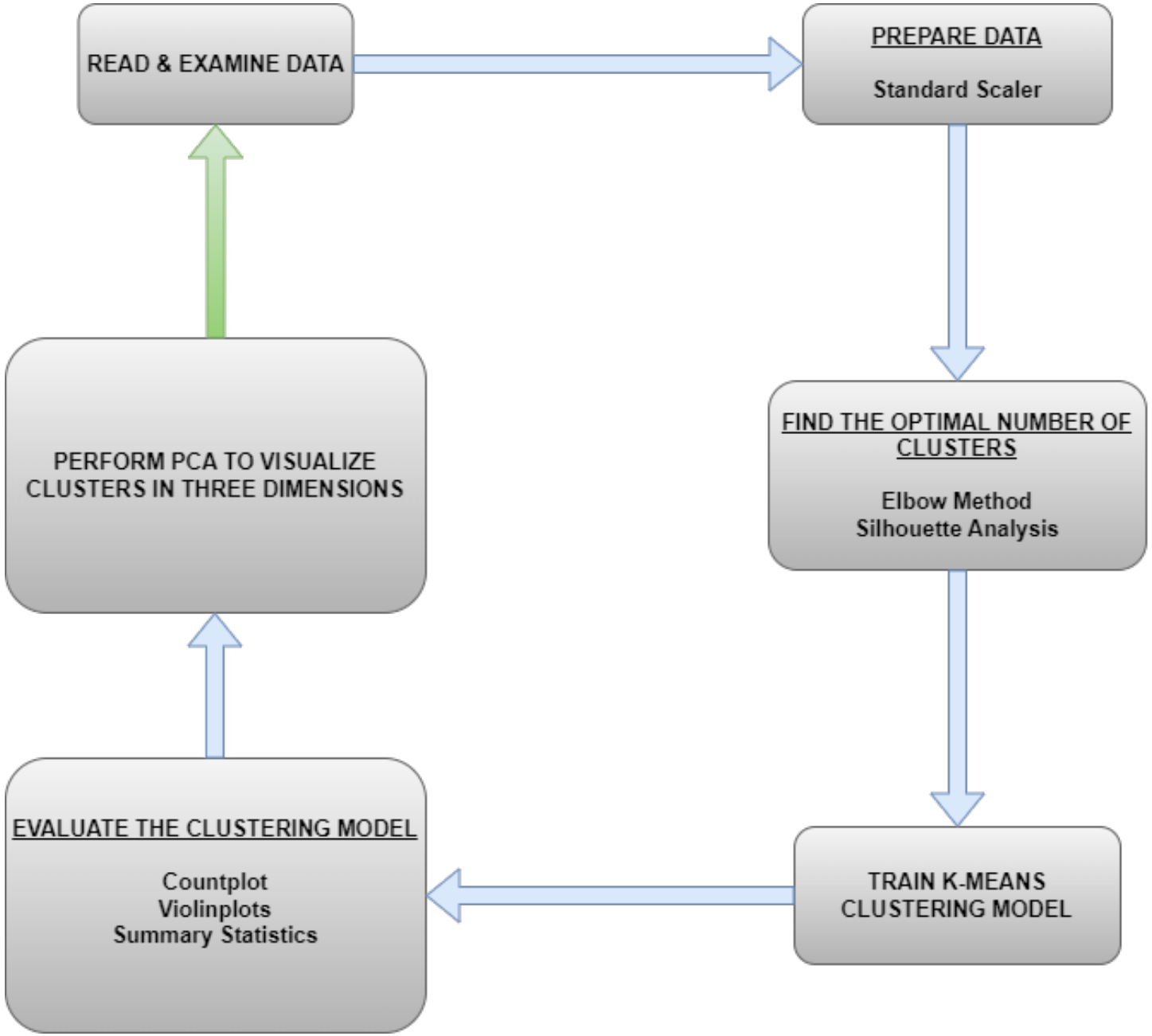
```
y_test.describe()

count    63900.000000
mean       152.216800
std        435.440385
min          0.550000
25%         18.750000
50%         48.600000
75%        131.900000
max      10281.200000
Name: monetary_value, dtype: float64
```

REGRESSION – EVALUATING BEST MODEL

INTERPRETING THE SCORES

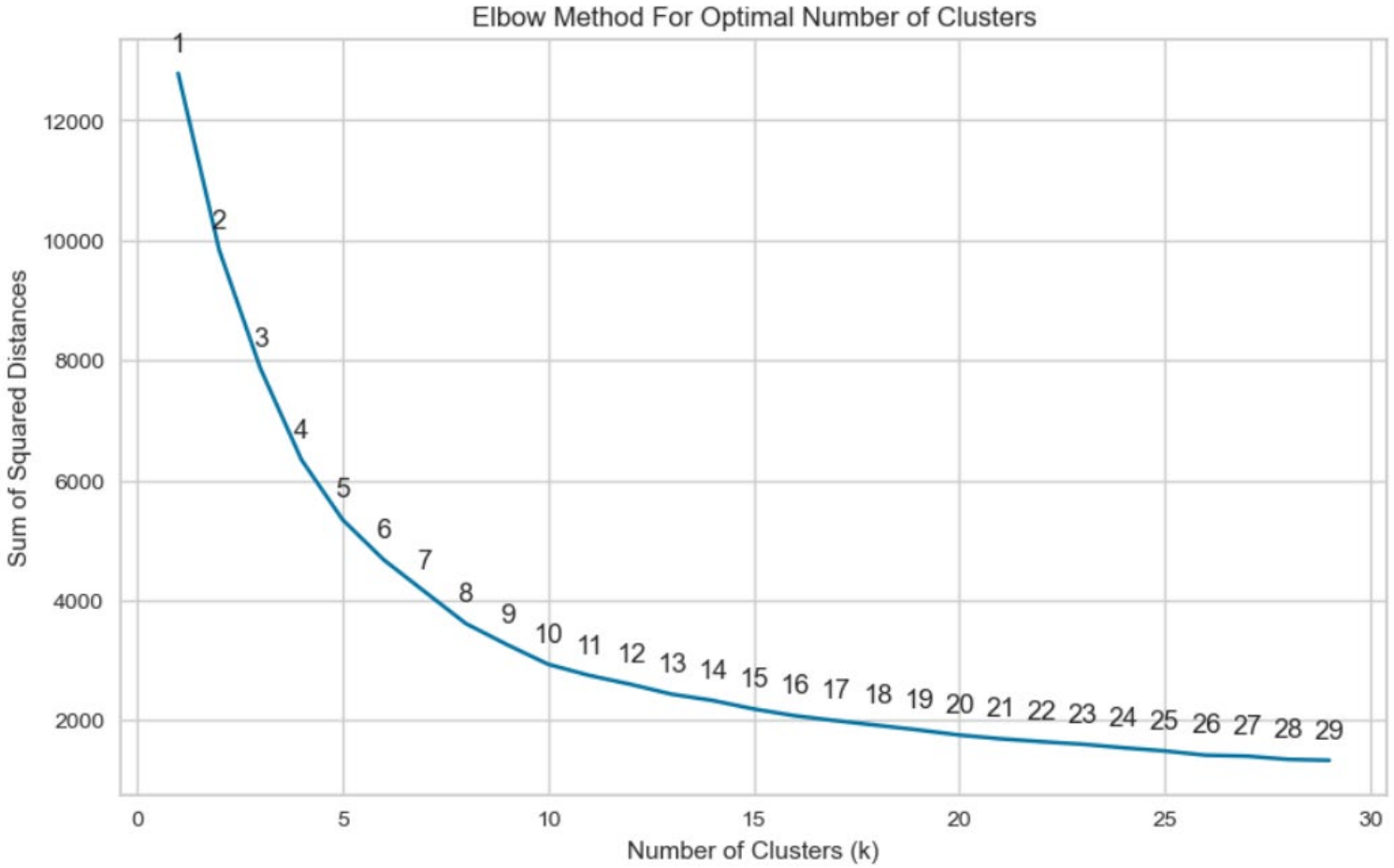
- ✓ R-2 of our model is near to perfect.
- ✓ MAE is less than 1% of the mean of the test set. This is a low error rate.
- ✓ MSE is 49.46 which says RMSE is close to 7.2. This is also a good score given the range of data.

CLUSTERING - FLOWCHART



CLUSTERING – INITIAL STEPS

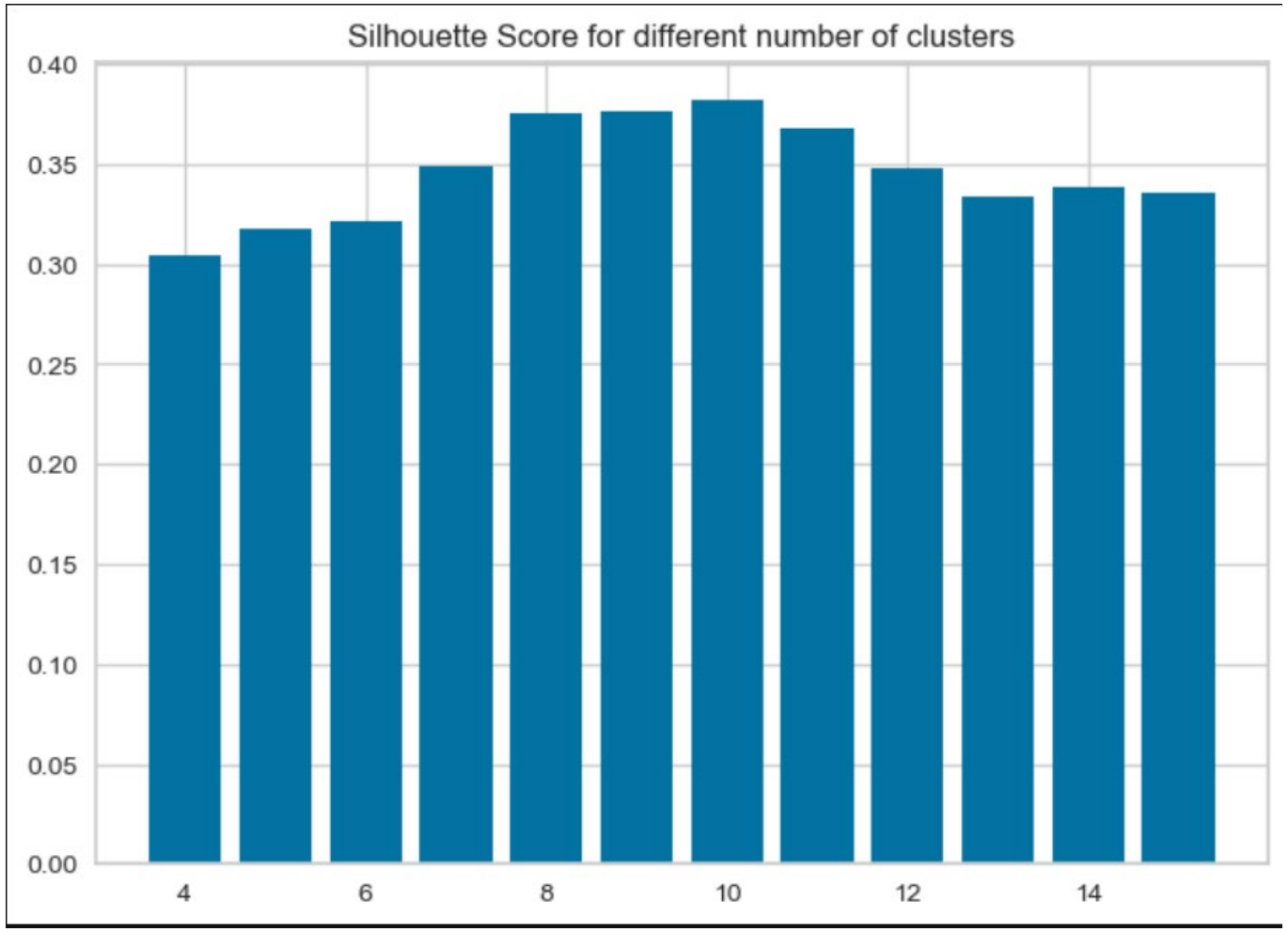
- ✓ We read and examined the same customer data as before.
- ✓ We scaled the data using StandardScaler.



- ✓ It looks like that the optimal number of clusters is around 8.

CLUSTERING – FIND OPTIMAL NUMBER OF CLUSTERS

ELBOW METHOD



CLUSTERING – FIND OPTIMAL NUMBER OF CLUSTERS

- ✓ For clarity, we take it one further step and perform silhouette analysis.
- ✓ The one with highest score is 10 but we choose 8 for not to end up with too many useless clusters.

```
n_clusters = list(range(4,16))
scores_dict = {}
for n_cluster in n_clusters:
    silhouette = \
    SilhouetteVisualizer(KMeans(n_cluster,
                                random_state=SEED))
    silhouette.fit(customer_data_scaled)
    silhouette.poof()
    scores_dict[n_cluster] = \
    silhouette.silhouette_score_.round(4)
scores_dict
```

ELBOW METHOD

```
# Generate a k-means clustering model using this o
kmeans = KMeans(n_clusters=8,random_state=SEED)

kmeans.fit(customer_data_scaled)
```

```
KMeans
KMeans(random_state=123456789)
```

```
# Determine the clusters for the users.
y_kmeans = kmeans.predict(customer_data_scaled)


# Print a DataFrame that shows each customer and t
results = customer_data.copy()
results.insert(0, 'cluster', y_kmeans)
```

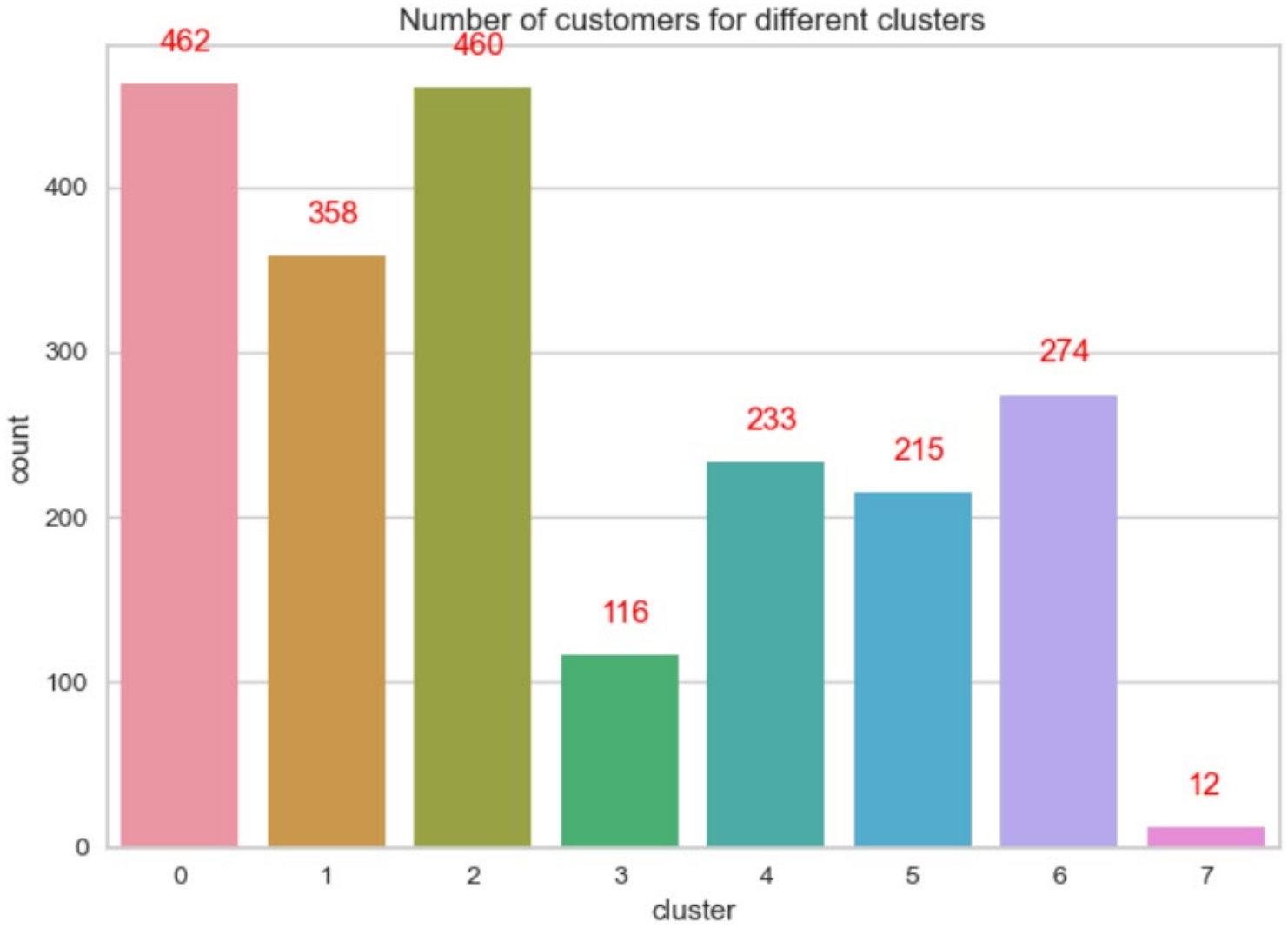
```
results.cluster.value_counts()
```

```
cluster
0    462
2    460
1    358
6    274
4    233
5    215
3    116
7     12
Name: count, dtype: int64
```

- ✓ We trained the model using 8 clusters and inserted the results to the data.

CLUSTERING

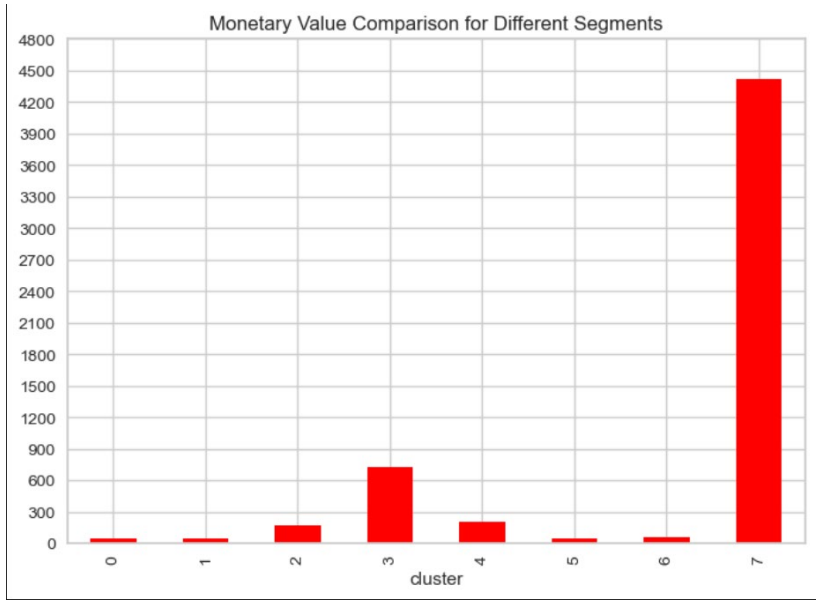
TRAIN THE MODEL

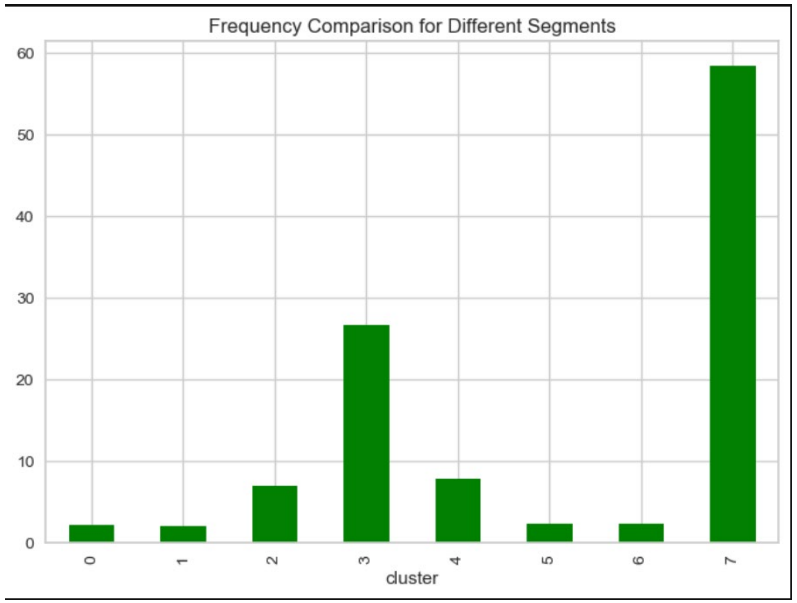


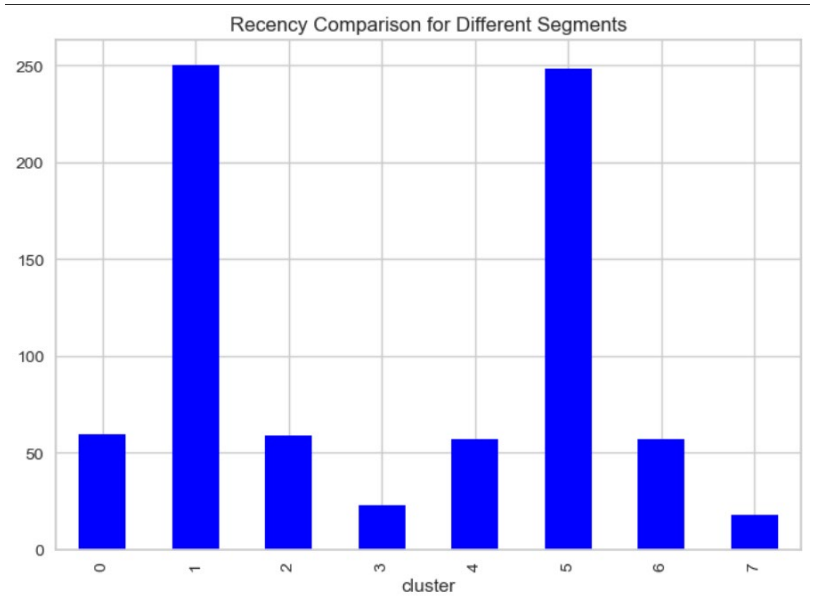
- ✓ Most number of customers are in clusters 0-2-1 and least number of customers are in 7-3.

CLUSTERING – EVALUATE THE MODEL

COUNTPLOT







- ✓ From the comparisons, we see a further segmentation:
  - ❖ Clusters 3-7: Highest spending, highest frequency, lowest recency
  - ❖ Clusters 2-4: Moderate spending, moderate frequency, moderate recency
  - ❖ Clusters 0-6: Lowest spending, lowest frequency, moderate recency
  - ❖ Clusters 1-5: Lowest spending, lowest frequency, highest recency

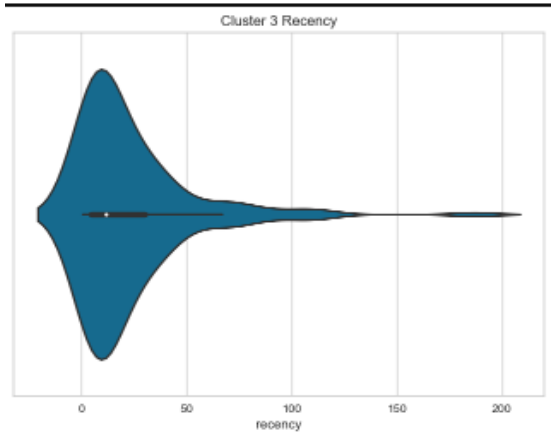
CLUSTERING – EVALUATE THE MODEL

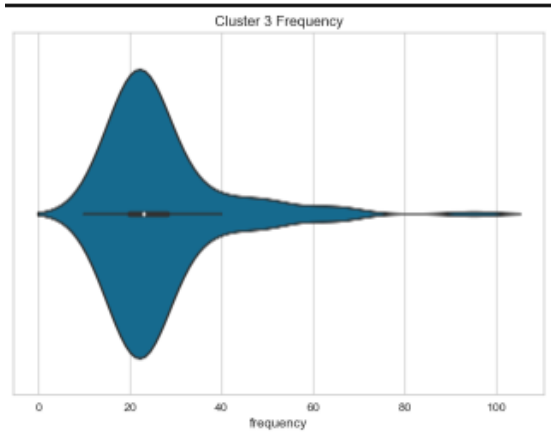
COMPARING MEANS USING BARPLOTS

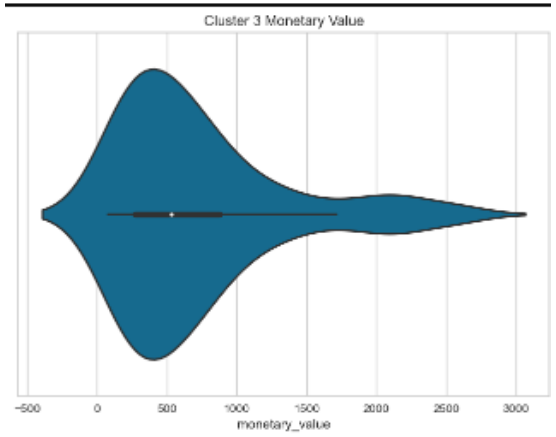
CLUSTERING – EVALUATE THE MODEL

COMPARING RECENCY, FREQUENCY & MONETARY VALUE DENSITIES USING VIOLINPLOTS

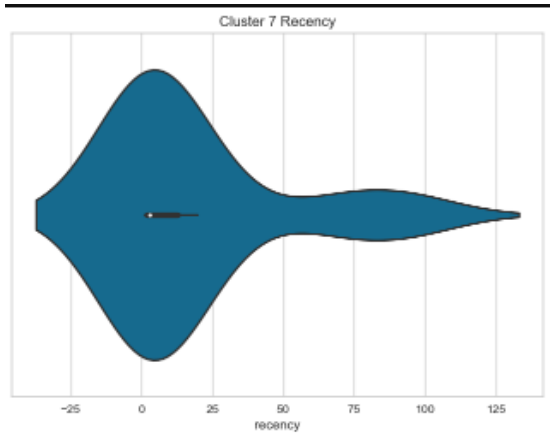
3

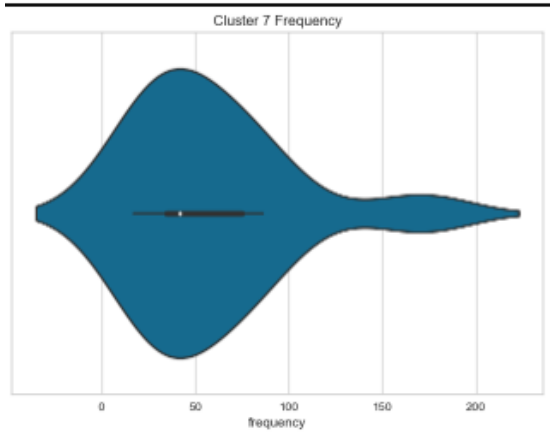

Cluster 3 Recency
0
50
100
150
200
recency

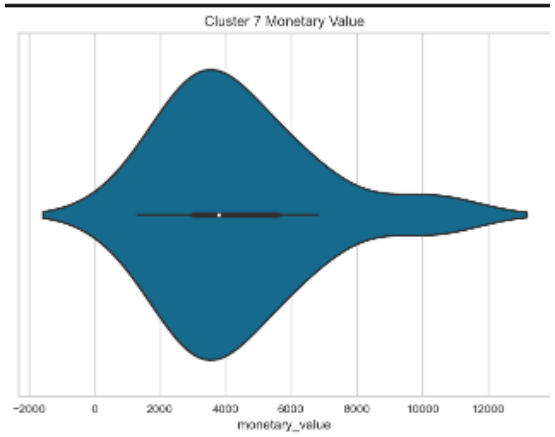

Cluster 3 Frequency
0
20
40
60
80
100
frequency


Cluster 3 Monetary Value
−500
0
500
1000
1500
2000
2500
3000
monetary_value

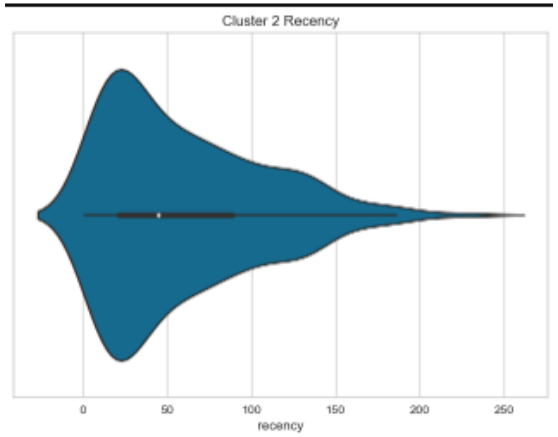
7

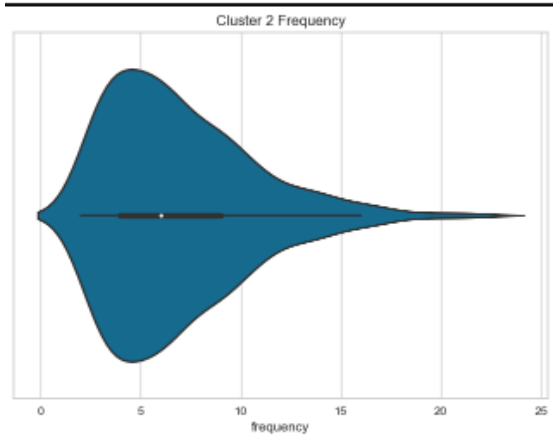

Cluster 7 Recency
−25
0
25
50
75
100
125
recency

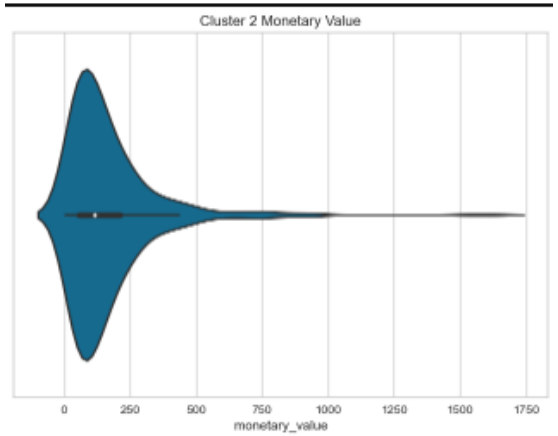

Cluster 7 Frequency
0
50
100
150
200
frequency


Cluster 7 Monetary Value
−2000
0
2000
4000
6000
8000
10000
12000
monetary_value

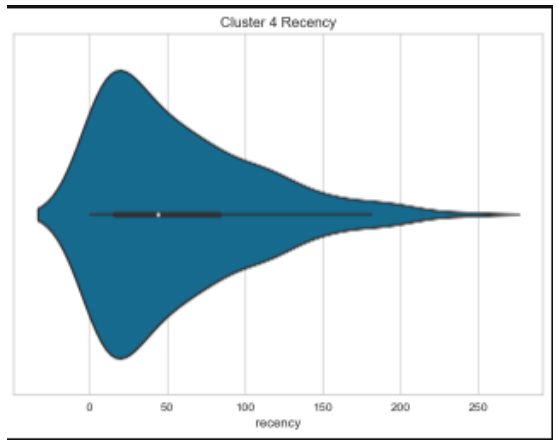
2

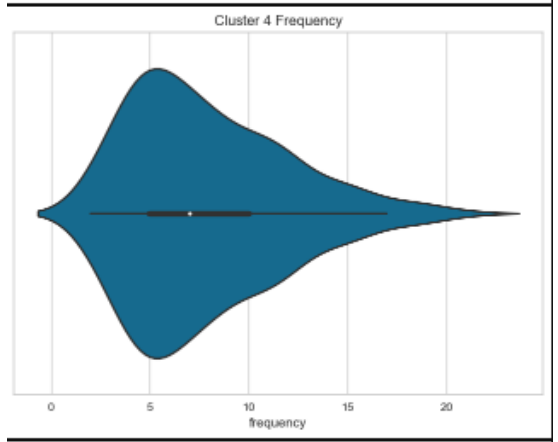

Cluster 2 Recency
0
50
100
150
200
250
recency

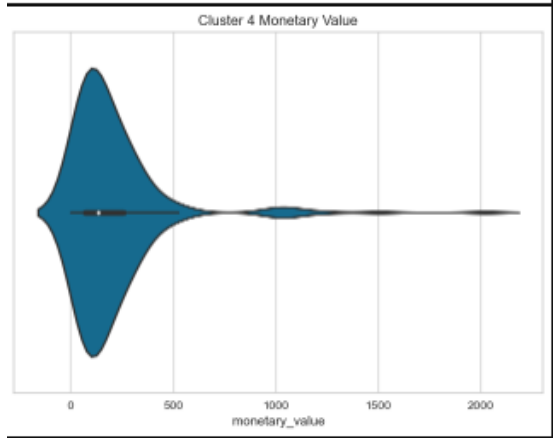

Cluster 2 Frequency
0
5
10
15
20
25
frequency


Cluster 2 Monetary Value
0
250
500
750
1000
1250
1500
1750
monetary_value

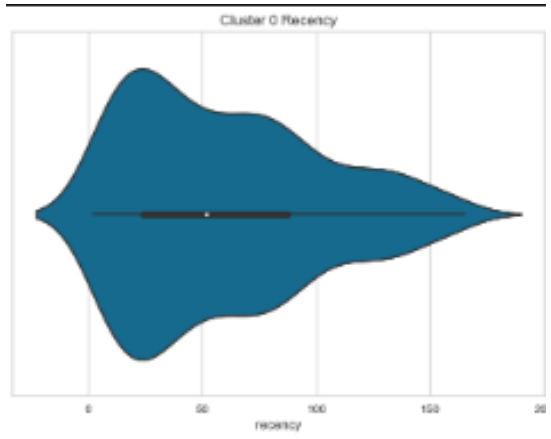
4

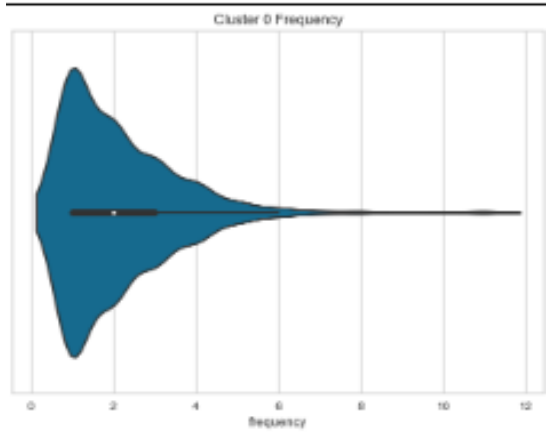

Cluster 4 Recency
0
50
100
150
200
250
recency

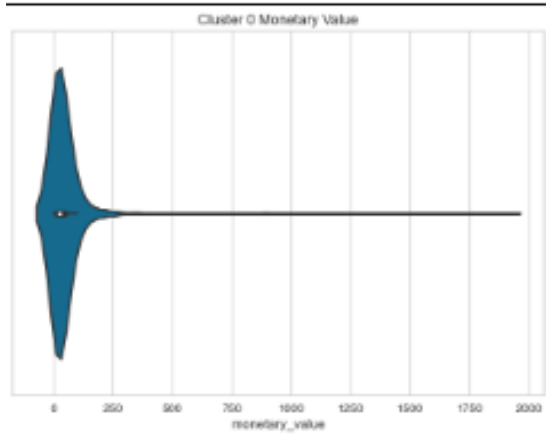

Cluster 4 Frequency
0
5
10
15
20
frequency


Cluster 4 Monetary Value
0
500
1000
1500
2000
monetary_value

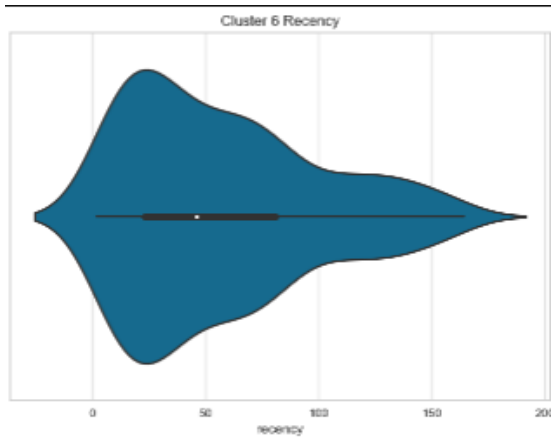
0

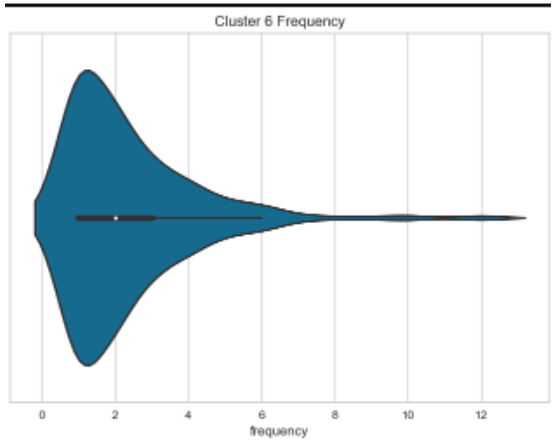

Cluster 0 Recency
recency

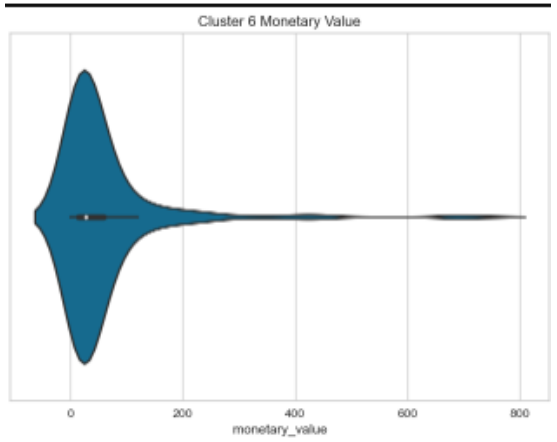

Cluster 0 Frequency
frequency


Cluster 0 Monetary Value
monetary_value

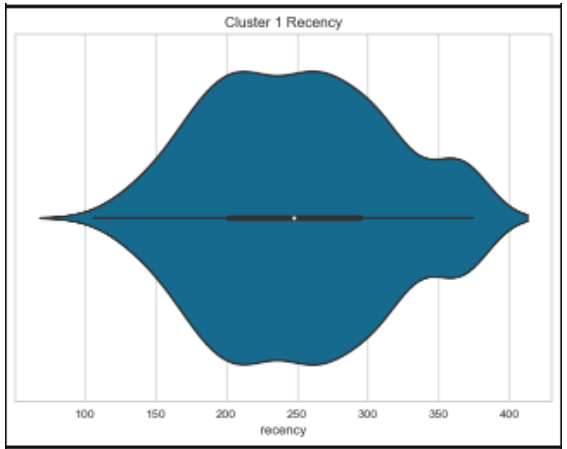
6

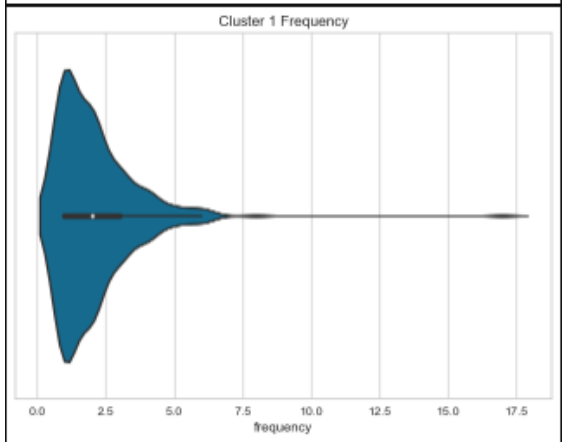

Cluster 6 Recency
recency

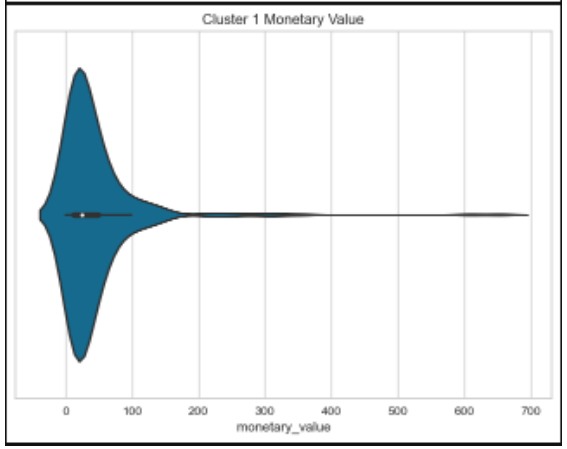

Cluster 6 Frequency
frequency


Cluster 6 Monetary Value
monetary_value

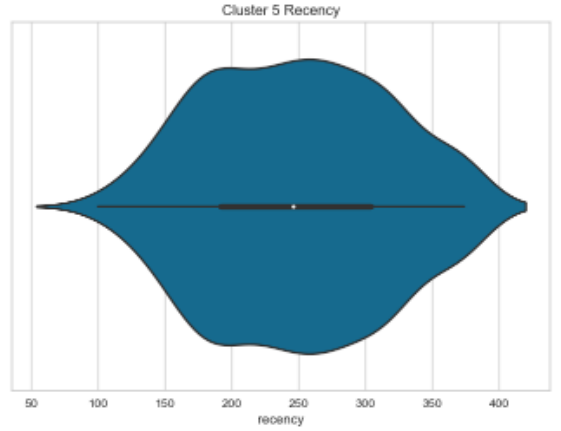
1

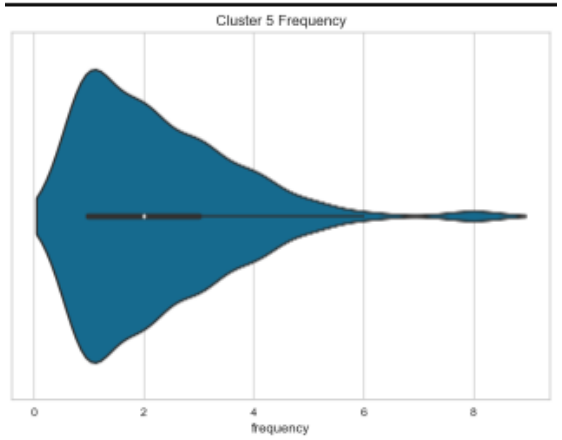

Cluster 1 Recency
recency

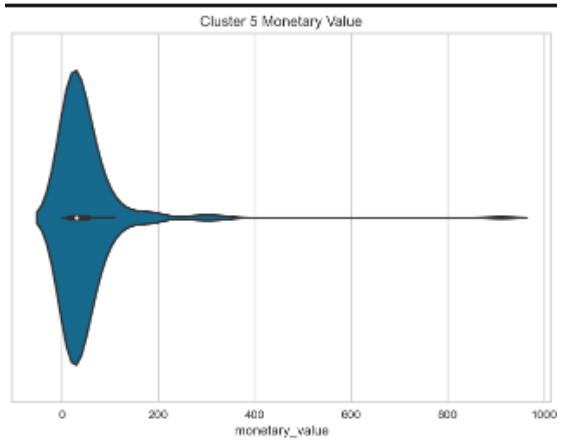

Cluster 1 Frequency
frequency


Cluster 1 Monetary Value
monetary_value

5


Cluster 5 Recency
recency

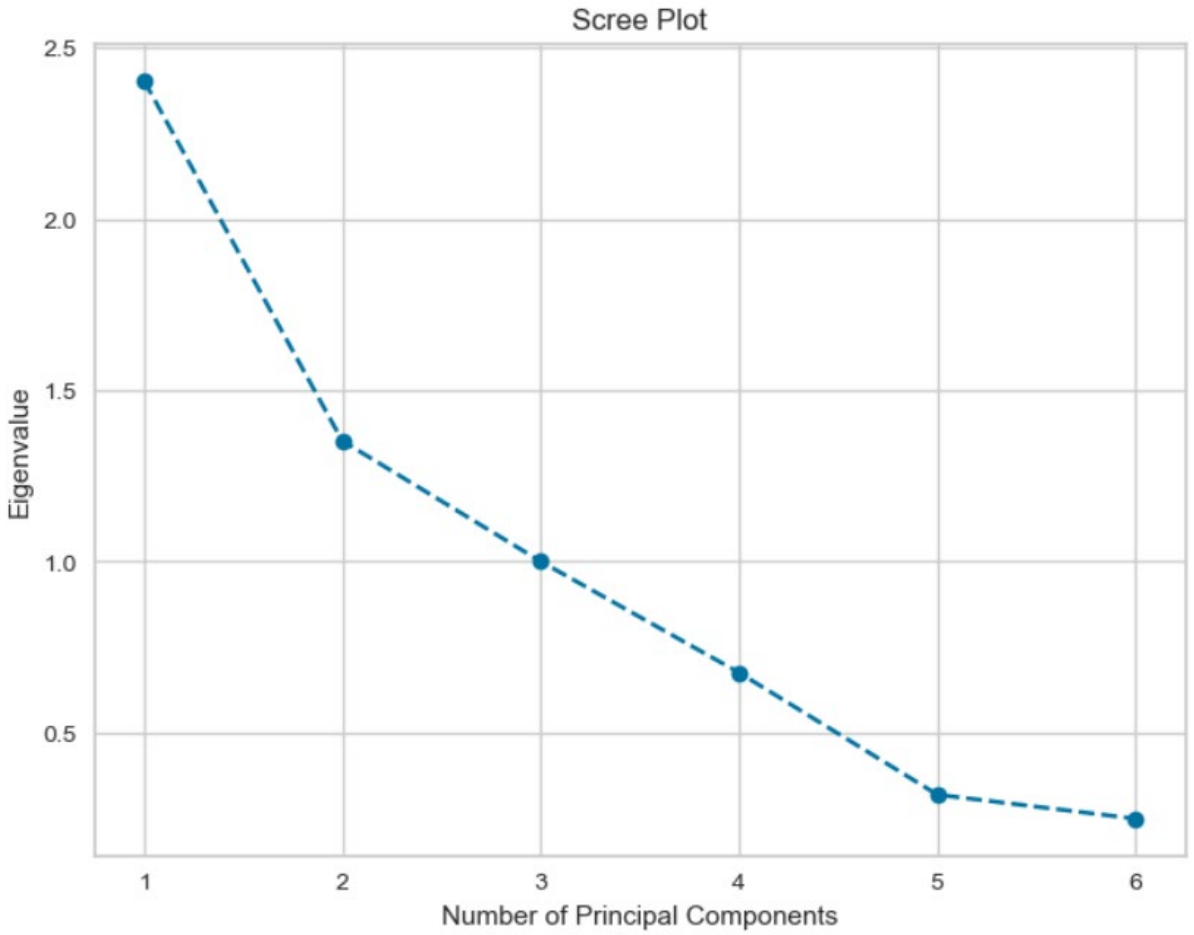

Cluster 5 Frequency
frequency


Cluster 5 Monetary Value
monetary_value

- ✓ The violinplots verify our segmentations.
- ✓ Clusters 3 & 7 includes most valuable customers. These customers can be made more satisfied by some additional perks.
- ✓ Clusters 2 & 4 is the consistent customer mass of the store. Spending of these group can be increased by some ads & campaigns. Emails, some cards by which they can use shopping cards or collaborations with some other brands can be applied on this segment.
- ✓ Clusters 0,6 & 1,5 are the ones with least contribution to monetary gains. Their mean spending is nearly 1/7 of the mean of other clusters combined. These are newer customers compared to others. More specific actions can be done about these:
  - ❖ Location-specific ad & campaigns
  - ❖ Product-specific ad & campaigns
  - ❖ The shopping times of these customers can be tracked and actions can be taken for those periods.
  - ❖ The businesses of competitors in the related regions can be analyzed.
- ✓ If products bought by lowest recency segment are unique to that segment or belongs that with highest proportion they can be simply removed to decrease spending by the store.

CLUSTERING – EVALUATE THE MODEL

CONCLUSIONS FROM COMPARISON



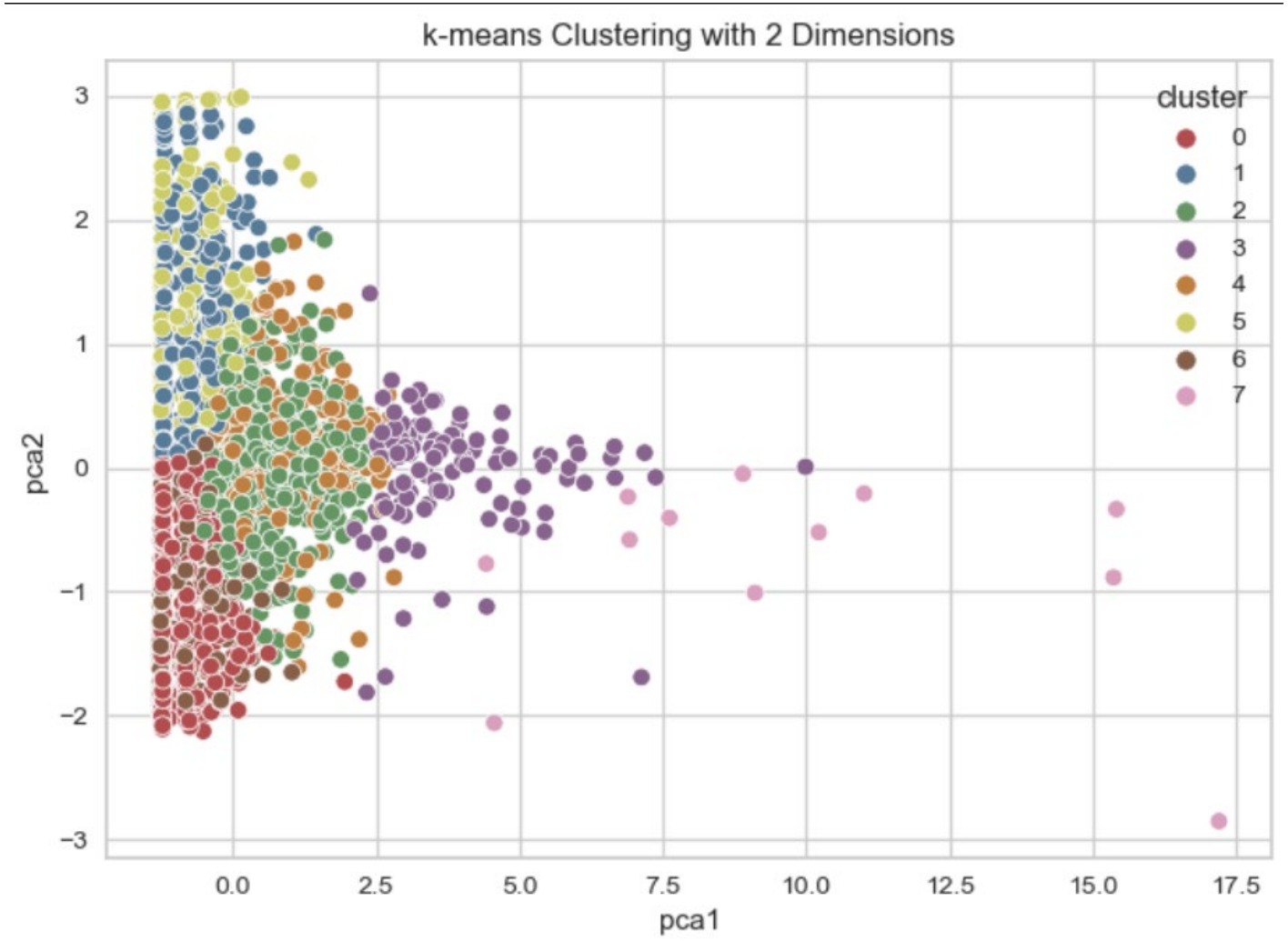
- ✓ By following Cattell's scree test, we choose 2-D PCA because the first bend occurs at 2.

CLUSTERING – EVALUATE THE MODEL

PCA

-

SCREE PLOT



CLUSTERING – EVALUATE THE MODEL

- ✓ As we see in PCA plot, segments can be taken as 3&7, 2&4 and 1&5&0&6. Another choice is that 7, 3, 2&4, 1&5 and 0&6. Some other combinations can be taken into account of course. This choice relates with different aspects of business-decision makings and needs further analysis in terms of business costs.

PERFORM PCA

CONCLUSION

- ✓ Our project goal was to create three machine learning models to predict customer churn and spending and provide a customer segmentation for business purposes.
- ✓ There are 3 main customer segmentations to treat differently.
- ✓ The heaviest customer activity happens during fall term most of the gain comes from customer segments 3&7.
- ✓ Our prediction models have some limitations and biases. The data should be increased and further optimizations could be processed on model. Some validation techniques omitted because of time concerns and this brings some misleading scores.
- ✓ Overall, our models can be beneficial with some little improvements.

APPENDIX

- GitHub Repo
- Evaluating Classification Models: 1 2 3
- Imbalanced Classification: 1
- Overfitting: 1
- Hyperopt: 1
- Lift Charts: 1
- ML Pipelines: 1 2
- PCA: 1 2 3

THANK YOU