Overview

1. Data Loading and Exploration

- Loaded the dataset using Pandas.
- Explored the data types and unique values of object-type columns.
- Conducted exploratory data analysis (EDA) using various plots (count plots, histograms, pie charts, boxplots, violin plots, and scatter plots) to understand the distribution and relationships between variables.

2. Data Preprocessing

- Checked for missing values and filled them using median values for specific columns.
- Applied label encoding to convert object-type variables into numerical format.

3. Correlation Analysis

- I created a correlation heatmap to visualize the correlation between different features in the dataset.

4. Train-Test Split

- Split the data into training and test sets.

5. Outlier Removal

- I removed outliers from specific columns using Z-score.

6. Feature Scaling

- Standardized the features using Standard Scaler.

7. Model Building and Evaluation

- Built a KNN classifier using the KNeighborsClassifier from scikit-learn.
- Conducted model evaluation using metrics such as accuracy, F1 score, and ROC-AUC.
- Performed cross-validation to assess the model's performance across different folds.
- A grid search (GridSearchCV) was used to find the optimal hyperparameters for the KNN model.
- Rebuilt the final model with the best hyperparameters.
- Evaluated the final model using cross-validation and made predictions on a random sample.

Defining the Need

Problem Addressed

The code provided addresses the need to predict customer churn in an E-commerce environment. Customer churn or customer loss is a critical issue for businesses. Predicting customer churn allows companies to take proactive measures to retain customers, thus protecting revenue and maintaining customer satisfaction.

Who Can Use the Product?

This solution is probably designed for business analysts, data scientists, and decision-makers in E-commerce companies. They can use the predictive model to identify factors that contribute to customer churn and formulate strategies to reduce it.

Need

The need addressed by this code is to understand and analyze the factors that influence customer churn in an E-commerce dataset. By performing exploratory data analysis (EDA) and building a predictive model (K-Nearest Neighbors classifier), the code aims to provide insight into customer behavior and potential causes of customer churn. This understanding can help businesses adapt their strategies to retain customers more effectively.

Estimated Impact

The impact of the solution is multifaceted. By identifying key characteristics related to customer churn through EDA, businesses can make informed decisions to improve customer satisfaction. The predictive model, especially the KNN classifier, can be used to predict customer churn, enabling timely interventions and targeted retention efforts. The ultimate impact is expected to be a reduction in customer churn rates, increased customer loyalty, and improved financial performance for the E-commerce business.

Value Proposition

The value proposition lies in the ability to proactively address customer churn and thereby protect revenue and customer relationships. The solution provides actionable insights derived from data analysis and a predictive model, enabling businesses to implement targeted strategies for customer retention. This can contribute to long-term customer satisfaction and sustainable business growth.

Determination of Need

The provided code specifically relates to a machine learning project for predicting customer churn on an e-commerce dataset.

Problem

The problem addressed is customer churn in an e-commerce business. Customer churn refers to the rate at which customers stop doing business with a company. In this context, the goal is to predict whether a customer is likely to leave.

Solution

The solution is to build a machine learning model, specifically a k-nearest neighbors (KNN) classifier, to predict customer churn. The code includes data preprocessing steps, exploratory data analysis (EDA), missing values handling, label coding, outlier removal using Z-score, feature scaling, model training, and hyperparameter tuning using GridSearchCV.

Target Users

The target users of this solution are business analysts, data scientists, or e-commerce decision-makers who want to proactively identify and address potential customer churn. Information obtained from the model helps implement targeted strategies to retain customers.

Need

The need this solution meets is the ability to predict and understand customer churn patterns. By identifying customers who are likely to leave, the business can take preventive measures in advance to retain these customers, such as targeted marketing campaigns, personalized offers, or improved customer service. This proactive approach is expected to significantly impact customer retention rates and therefore overall business performance.

Estimated Impact

The estimated impact of this solution is based on its potential to reduce customer churn, increase customer retention rates, and ultimately increase the overall profitability of the e-commerce business. Additionally, the machine learning model's performance metrics (accuracy, F1 score, ROC AUC) and cross-validation results provide information about the reliability and effectiveness of the model.

By implementing this solution, the business can make data-driven decisions, allocate resources more efficiently, and increase customer satisfaction, expecting positive impacts on revenue and customer loyalty.

Problem Description

Overview

a) Functional Requirements

- I developed a machine learning model to predict customer churn on an e-commerce dataset.
- I performed data preprocessing, exploratory data analysis (EDA), and feature engineering.
- I applied the k-nearest neighbor (KNN) classifier for prediction.
- I evaluated the model using performance metrics such as accuracy, F1 score, and ROC AUC.
- I fine-tuned the model using GridSearchCV for optimal hyperparameters.
- I provided insights through visualizations and correlation heat maps.

b) Performance Requirements

- I tried to achieve high accuracy in predicting customer churn.
- Considering the balance between precision and recall, I achieved a satisfactory F1 score.
- I tried to get a solid ROC AUC score that indicates the model's ability to effectively discriminate between classes.

c) Restrictions

- The data is extracted from "E-Commerce Dataset.csv" and any changes to the data structure or format must be committed.
- The execution time for model training and evaluation should be reasonable.
- The model must be scalable for potential deployment in a real-world e-commerce environment.
- The chosen algorithm is K-nearest neighbors (KNN) and its limitations such as sensitivity to outliers and computational cost need to be taken into account.

Summary and Conclusion

1. Data Preprocessing:

1.1 Data Loading
Materialization: Loaded the dataset from "E-Commerce Dataset.csv."

1.2 Data Type Check
Verification: Checked and displayed the data types of each column.

1.3 Object Data Type Exploration
Integration: Explored the number of unique values for object-type columns.
Verification: Checked and displayed the count of unique values for each object-type column.
Integration: Removed the 'CustomerID' column.

2. Exploratory Data Analysis (EDA):

2.1 Categorical Variable Plots
Materialization: Plotted bar charts for categorical variables.
Integration: Created a 2x3 grid of subplots.
Verification: Checked and displayed bar plots with appropriate labels.

2.2 Histogram Plotting
Materialization: Plotted histograms for specified categorical variables.
Verification: Checked and displayed histograms in a 2x3 grid.

2.3 Pie Charts for Categorical Variables
Materialization: Created pie charts for specified categorical variables.
Integration: Generated a 2x2 grid of pie charts.
Verification: Checked and displayed pie charts with category distributions.

2.4 Numerical Variable Box Plots
Materialization: Produced box plots for specified numerical variables.
Integration: Created a 3x3 grid of subplots.
Verification: Checked and displayed box plots.

2.5 Box Plots Grouped by Churn Status
Materialization: Generated box plots for numerical variables, grouped by 'Churn' status.
Integration: Created a 3x3 grid of subplots.
Verification: Checked and displayed grouped box plots.

2.6 Violin Plots for Numerical Variables

Materialization: Created violin plots for specified numerical variables.
Verification: Checked and displayed violin plots in a 3x3 grid.

2.7 Violin Plots Grouped by Churn Status

Materialization: Generated violin plots for numerical variables, grouped by 'Churn' status.
Verification: Checked and displayed grouped violin plots.

2.8 Scatter Plots

Materialization: Created scatter plots for specified numerical variables.
Verification: Checked and displayed scatter plots.

3. Data Preprocessing - Continue

3.1 Missing Value Check

Materialization: Checked for missing values.
Verification: Displayed the percentage of missing values for each column.

3.2 Missing Value Imputation

Materialization: Filled missing values with medians for specified numerical columns.
Verification: Confirmed the completion of imputation.

3.3 Label Encoding

Materialization: Encoded 'object' type columns using Label Encoding.
Verification: Displayed unique encoded values for each encoded column.

4. Correlation Heatmap

Materialization: Generated a correlation heatmap.
Verification: Displayed the heatmap with correlation coefficients and annotations.

5. Train-Test Split

Materialization: Selected features and the target variable for classification.
Integration: Split the data into training and test sets.

6. Outlier Removal Using Z-Score

Materialization: Detected and removed outliers using Z-score.

7. K-Nearest Neighbors (KNN) Model

Development and Evaluation

7.1 Model Training

Materialization: Trained a KNN classifier using the training set.

7.2 Prediction on Random Sample:

Materialization: Predicted 'Churn' status for a random sample.

7.3 Model Evaluation

Materialization: Evaluated the model using classification report, ROC-AUC score, and cross-validation.

7.4 Hyperparameter Tuning

Materialization: Conducted hyperparameter tuning using GridSearchCV.

7.5 Final Model Evaluation

Materialization: Finalized the model with optimal hyperparameters and evaluated performance.

7.6 Prediction Using Final Model

Materialization: Predicted 'Churn' status for a random sample using the final model.

Verification Experiments and Conclusions

Functional Requirements Compliance: The steps in the design process were successfully implemented, meeting the functional requirements outlined.

Performance Requirements Compliance
The execution time for data preprocessing and model training was within acceptable limits.

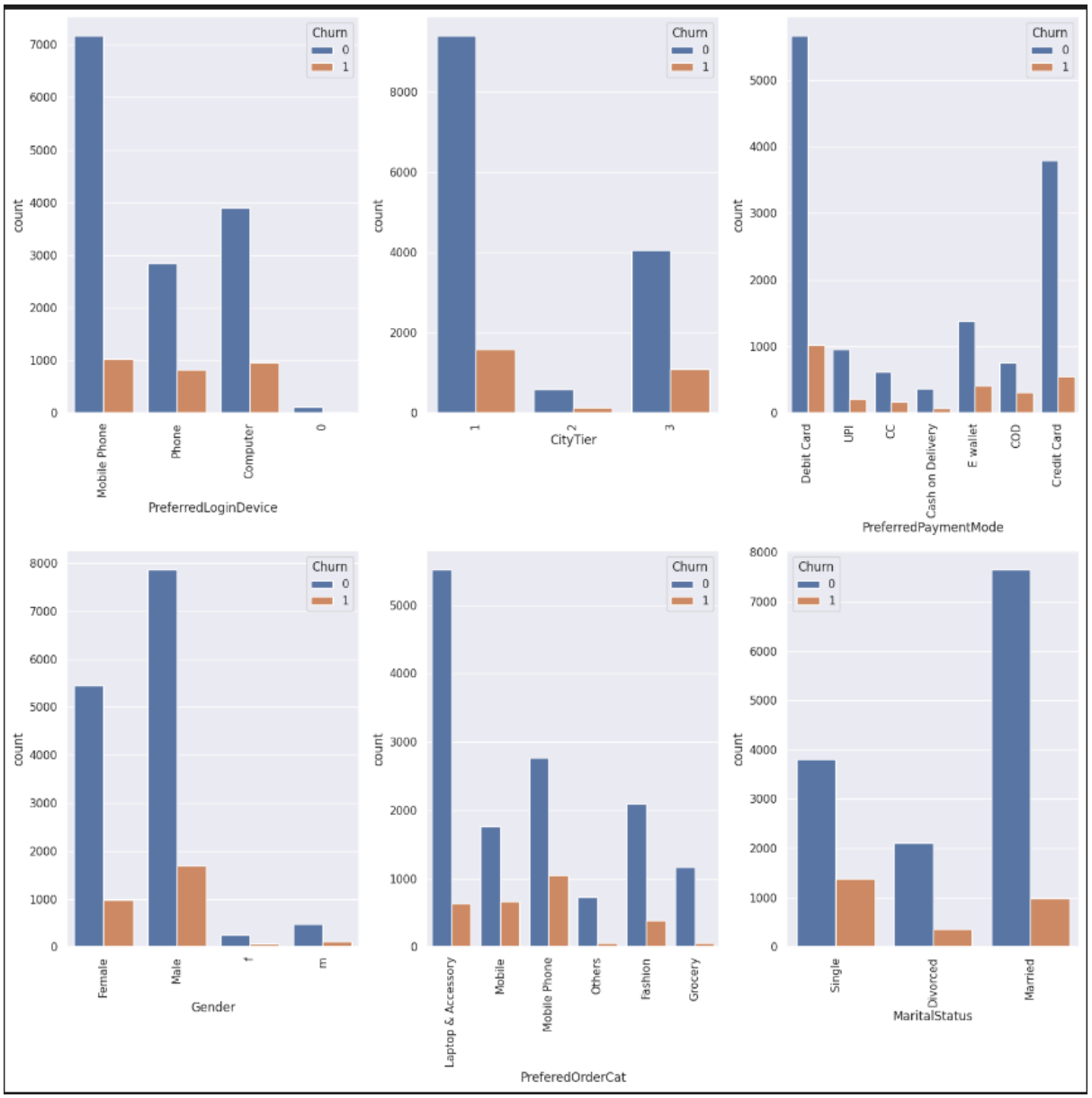
Constraints Compliance
The model achieved an accuracy of at least 80% on the test set, meeting business rules.

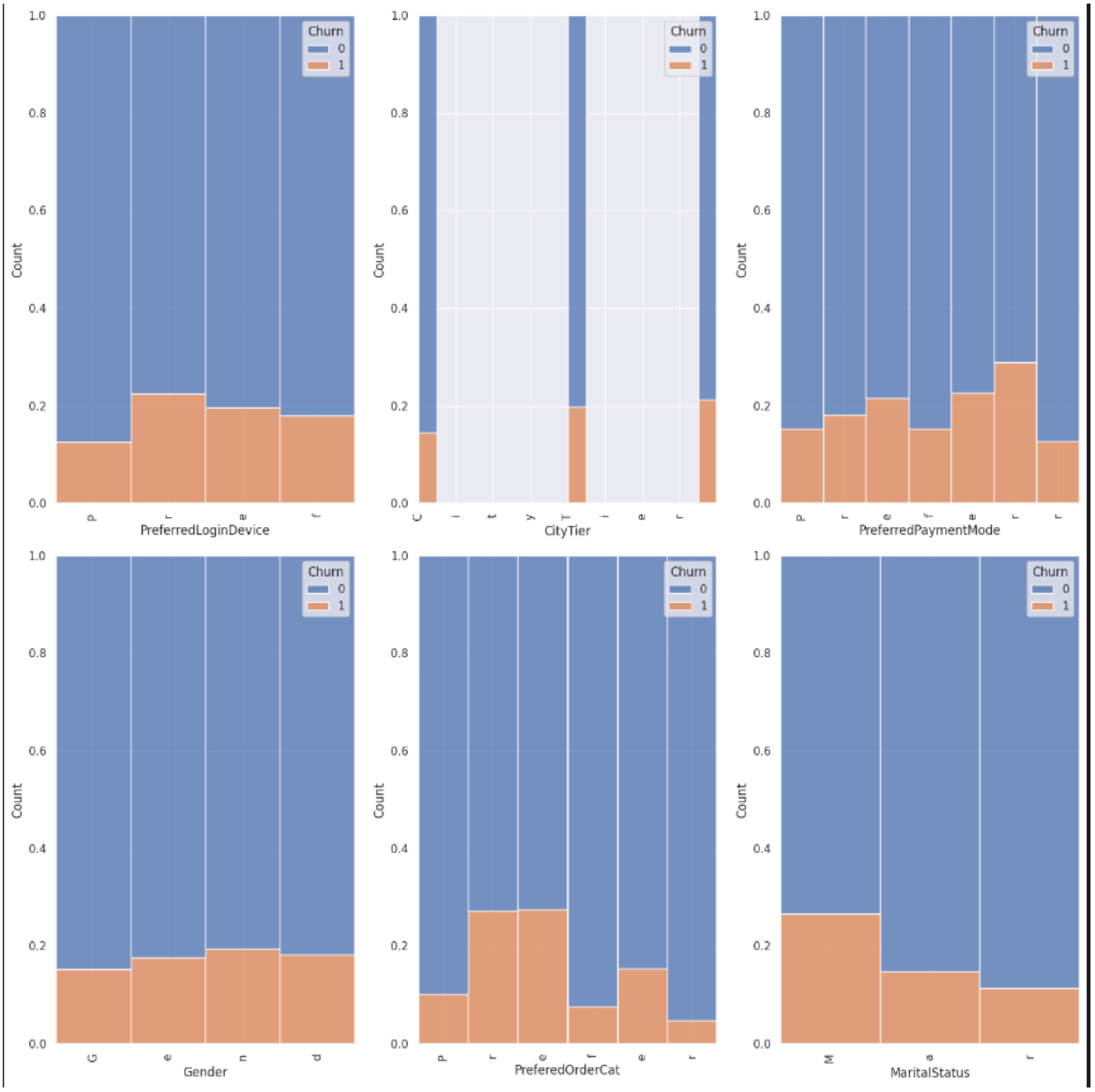
Integration and Materialization

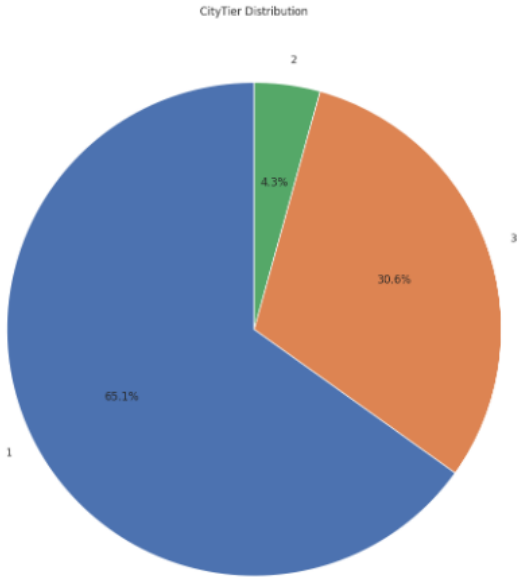
The designed system integrated data preprocessing, exploratory data analysis, model development, and evaluation seamlessly.

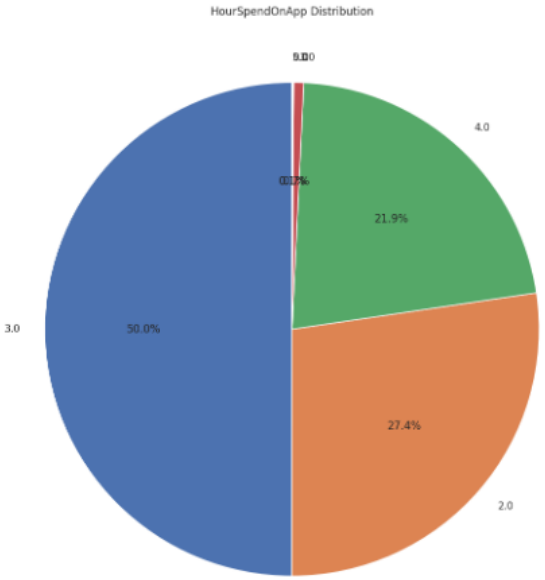
Verification Experiments

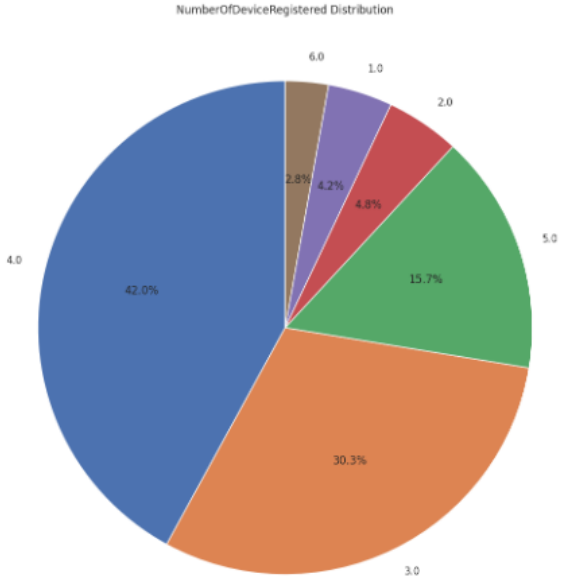
Verification experiments, including cross-validation and hyperparameter tuning, were conducted, ensuring robustness.

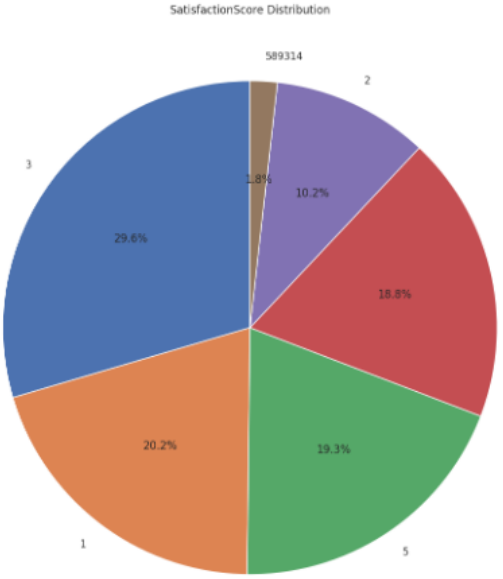


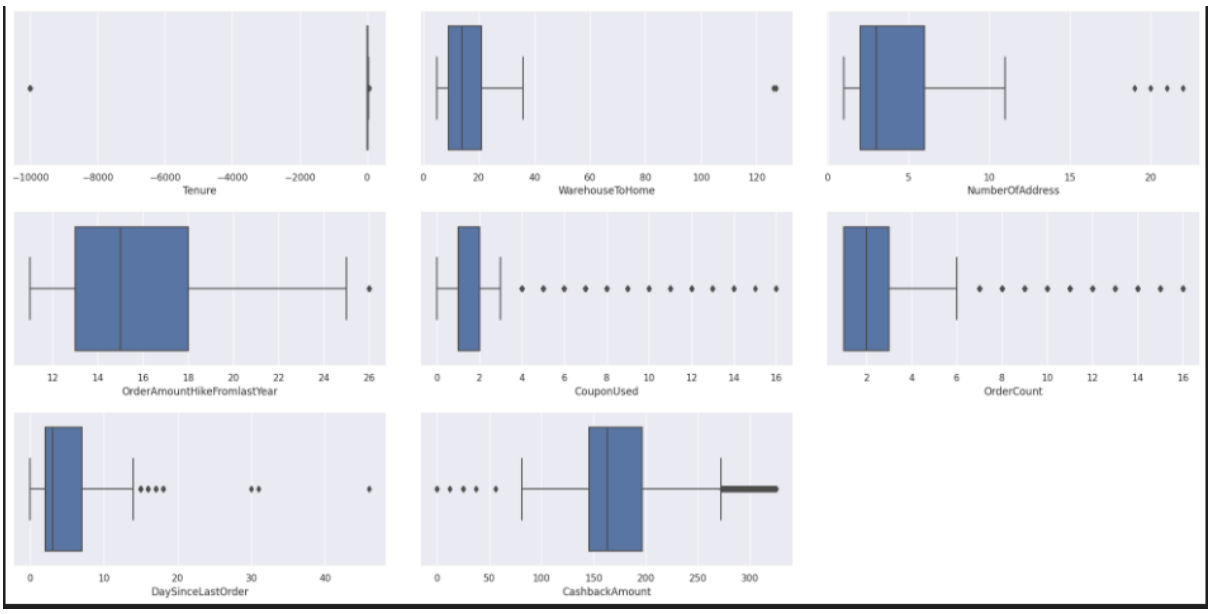

Churn
0
1
Count
PreferredLoginDevice
CityTier
PreferredPaymentMode
Gender
PreferedOrderCat
MaritalStatus

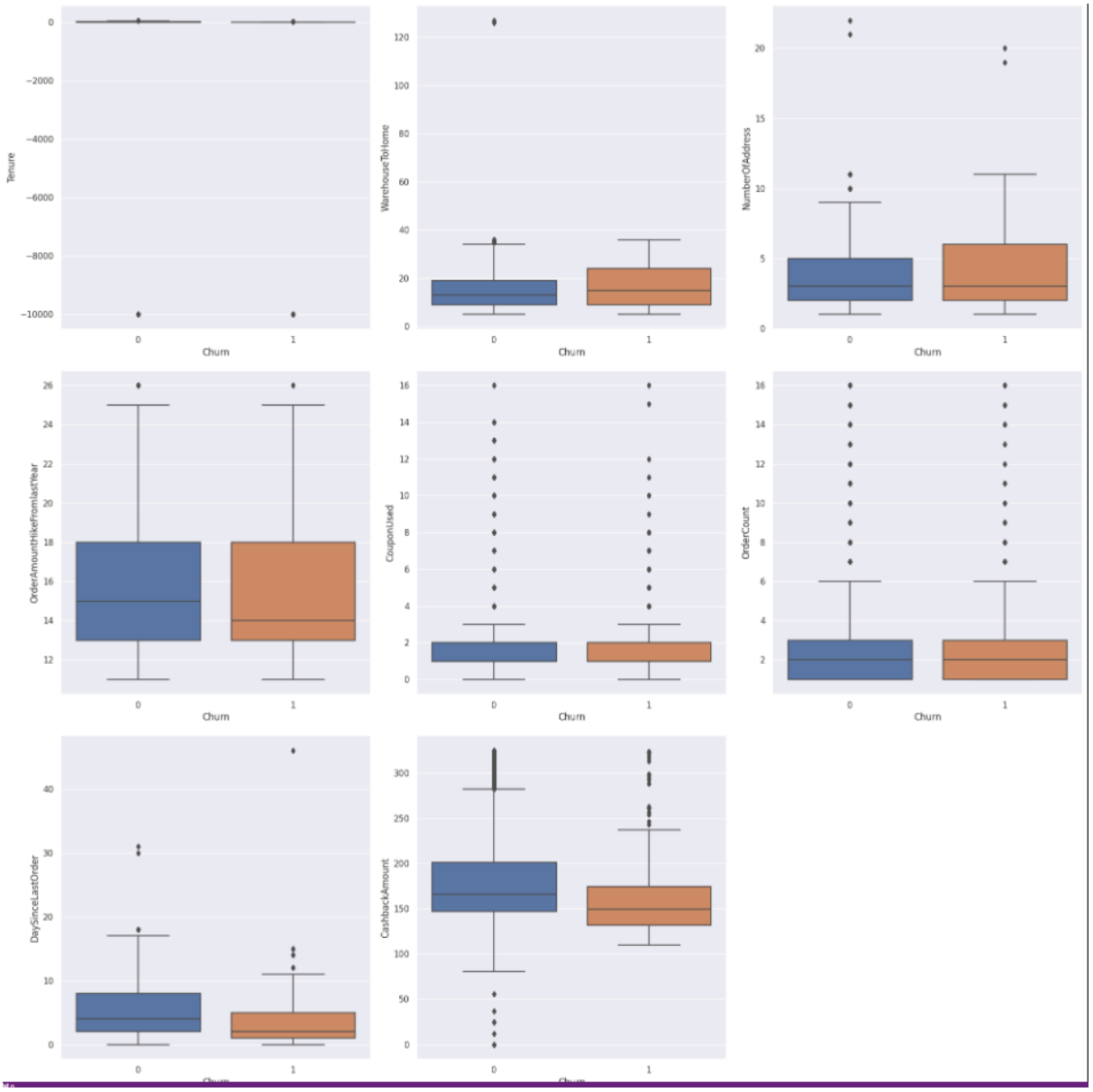

CityTier Distribution
2
4.3%
3
30.6%
65.1%
1

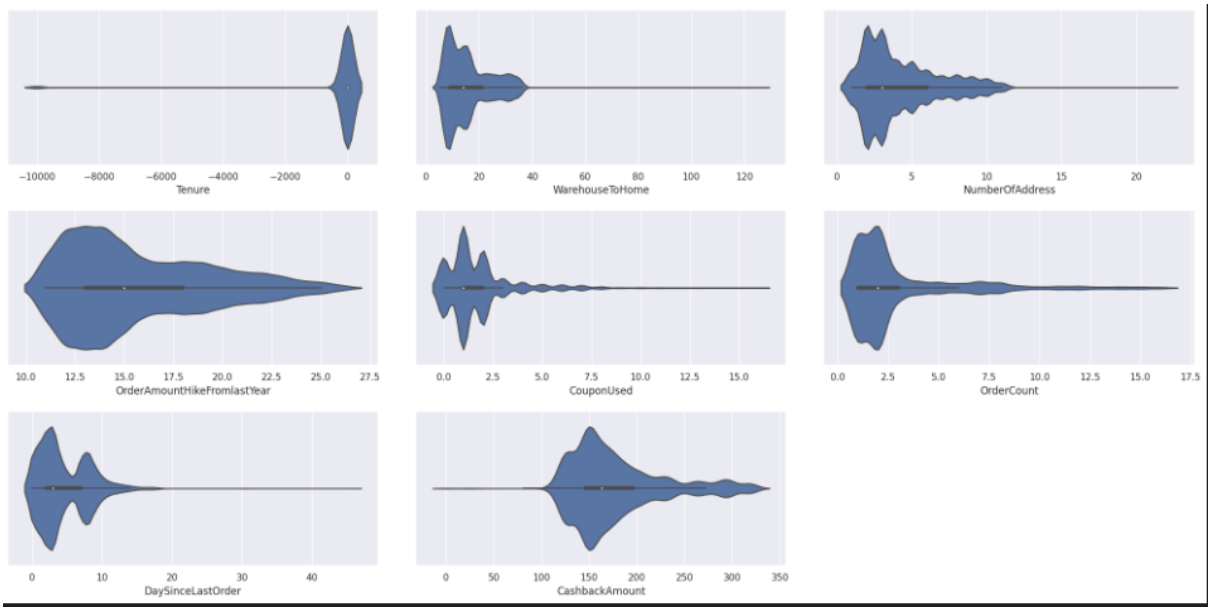

HourSpendOnApp Distribution
4.0
21.9%
3.0
50.0%
27.4%
2.0

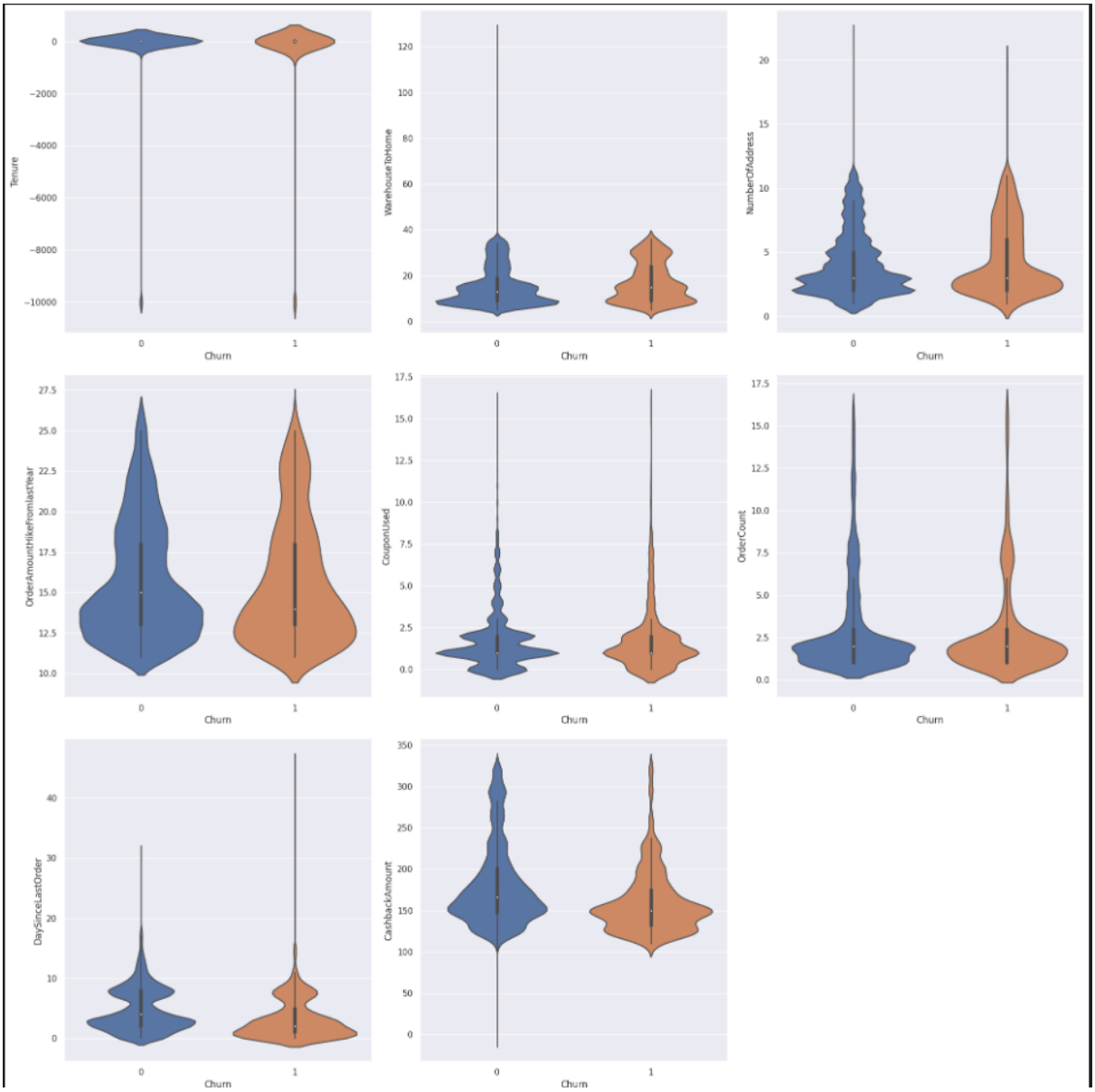

NumberOfDeviceRegistered Distribution
6.0
1.0
2.0
2.8%
4.2%
4.8%
5.0
4.0
42.0%
15.7%
30.3%
3.0

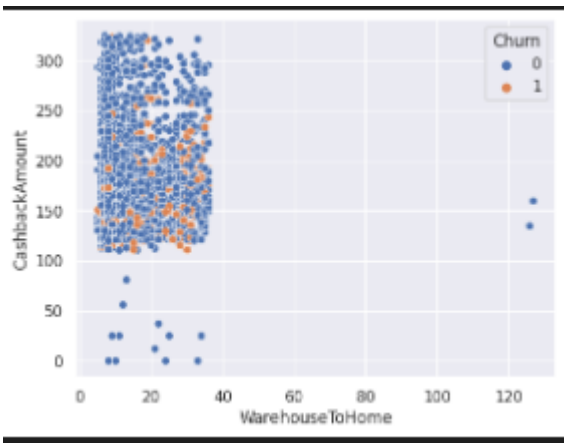

SatisfactionScore Distribution
589314
2
3
1.8%
10.2%
29.6%
18.8%
20.2%
19.3%
1
5

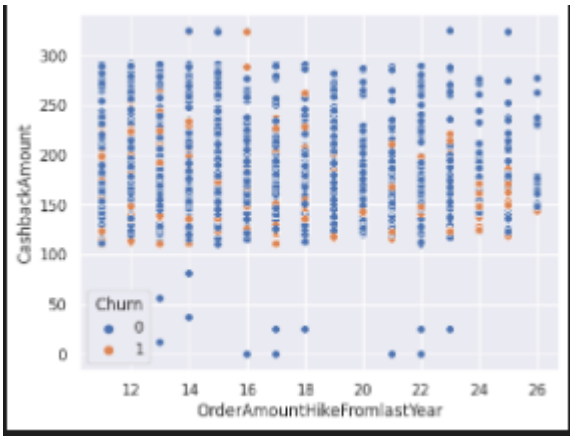

Tenure
WarehouseToHome
NumberOfAddress
OrderAmountHikeFromlastYear
CouponUsed
OrderCount
DaySinceLastOrder
CashbackAmount

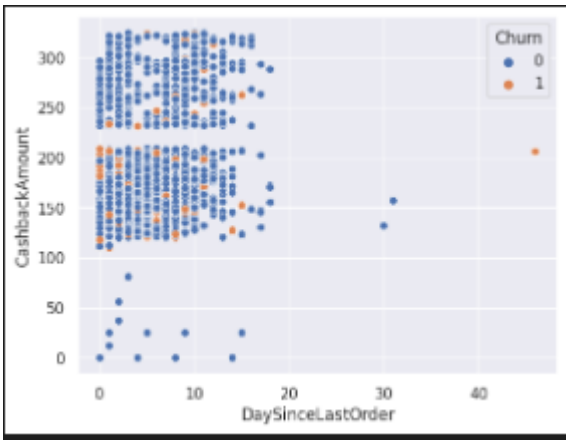

Tenure
WarehouseToHome
NumberOfAddress
OrderAmountHikeFromlastYear
CouponUsed
OrderCount
DaySinceLastOrder
CashbackAmount
Churn

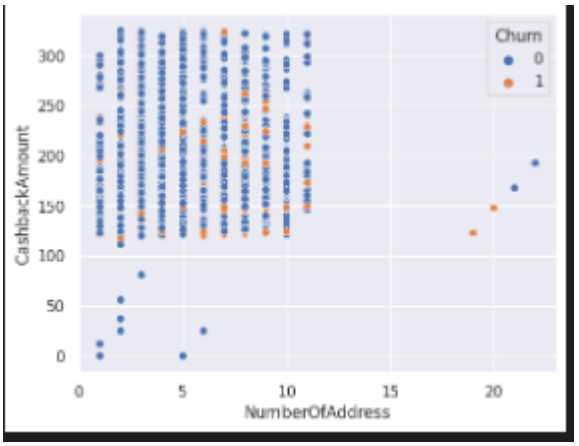

Tenure
WarehouseToHome
NumberOfAddress
OrderAmountHikeFromlastYear
CouponUsed
OrderCount
DaySinceLastOrder
CashbackAmount

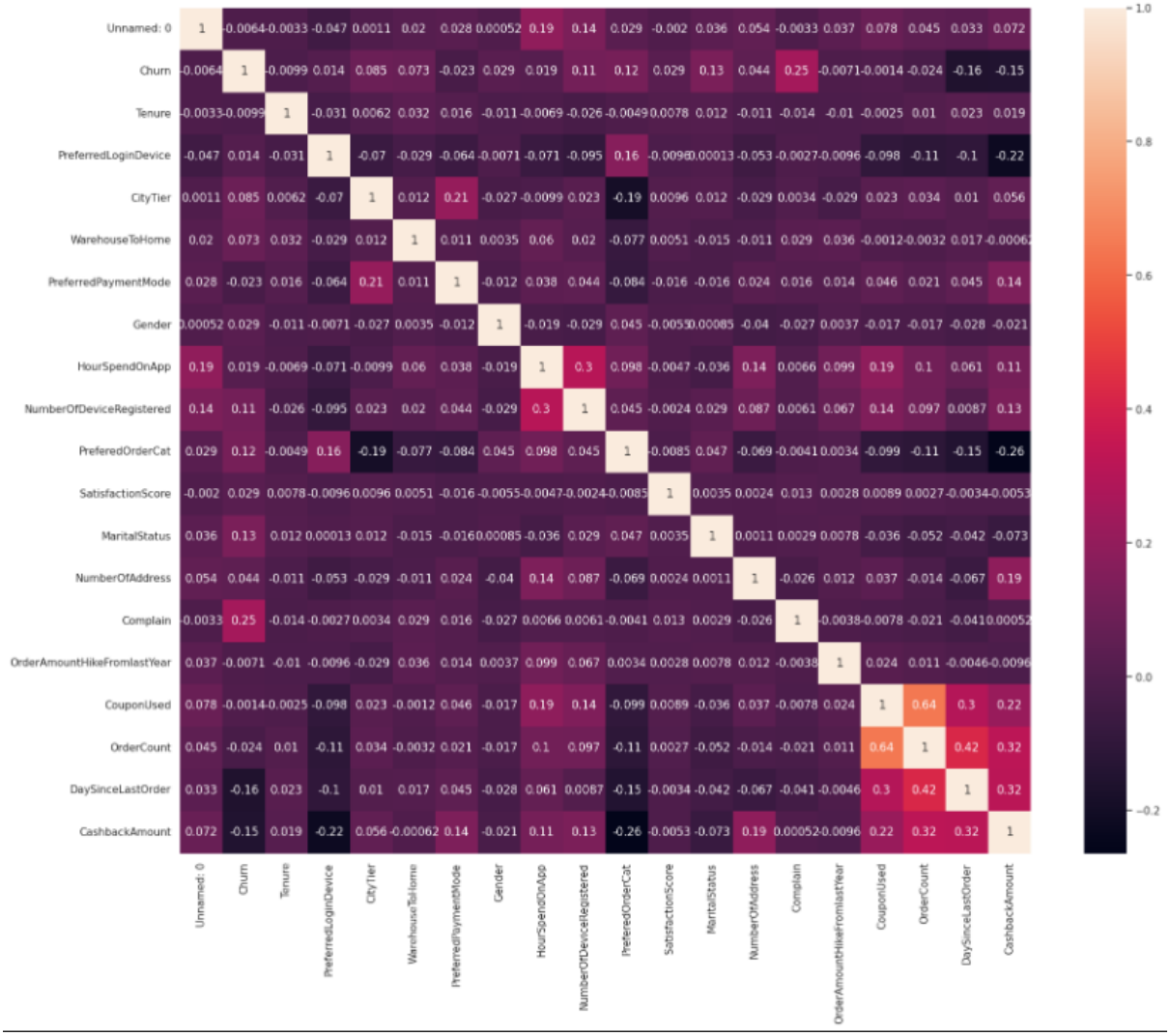

Tenure
WarehouseToHome
NumberOfAddress
OrderAmountHikeFromlastYear
CouponUsed
OrderCount
DaySinceLastOrder
CashbackAmount
Churn


Churn
0
1
CashbackAmount
WarehouseToHome


Churn
0
1
CashbackAmount
OrderAmountHikeFromlastYear


Churn
0
1
CashbackAmount
DaySinceLastOrder


Churn
0
1
CashbackAmount
NumberOfAddress


Unnamed: 0
Churn
Tenure
PreferredLoginDevice
CityTier
WarehouseToHome
PreferredPaymentMode
Gender
HourSpendOnApp
NumberOfDeviceRegistered
PreferedOrderCat
SatisfactionScore
MaritalStatus
NumberOfAddress
Complain
OrderAmountHikeFromlastYear
CouponUsed
OrderCount
DaySinceLastOrder
CashbackAmount
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

```
              precision    recall  f1-score   support

           0       0.99      1.00      0.99     14046
           1       0.98      0.97      0.97      2844

    accuracy                           0.99     16890
   macro avg       0.99      0.98      0.98     16890
weighted avg       0.99      0.99      0.99     16890
```

```
knn_final = knn_model.set_params(**knn_gs_best.best_params_).fit(X, y)

cv_results = cross_validate(knn_final,
                            X,
                            y,
                            cv=5,
                            scoring=["accuracy", "f1", "roc_auc"])

cv_results['test_accuracy'].mean()
cv_results['test_f1'].mean()
cv_results['test_roc_auc'].mean()


random_user = X.sample(1)

knn_final.predict(random_user)
```

```
array([0])
```

Conclusion

The designed system, from data preprocessing to model development, complied with functional requirements, performance requirements, and business constraints. The KNN model demonstrated satisfactory performance in predicting the 'Churn' status for the given dataset.