The Purchase Prognosticator: Forecasting Customer Buys using XGBoost and Random Forest

Ishta Rachel Mathew
PG Scholar
Master of Computer Applications
Amal Jyothi College of Engineering
Kottayam, Kerala
ishtarachelmathew2000@gmail.com

Anit James
Assistant Professor
Master of Computer Applications
Amal Jyothi College of Engineering
Kottayam, Kerala
anitjames@amaljyothi.ac.in

***Abstract*— Predicting client purchase behavior has become a crucial task for e-commerce firms in today's data-driven business world. This research presents a machine learning-based comprehensive client purchase prediction system. To predict future purchase trends, this system uses the Random Forest and XGBoost algorithms to analyze previous consumer data, including financial indicators, frequency, and recency. The model's accuracy is thoroughly assessed, demonstrating its ability to provide organizations with useful insights into the preferences and behaviors of their customers. With the system's framework in place for data-driven decision-making, businesses can improve customer experiences, customize marketing campaigns, and maximize inventory control. This study examines the methods, findings, and encouraging possibilities for a wider use of customer purchase prediction systems in a range of businesses.**

Keywords—Machine Learning, Random Forest Algorithm, XGBoost Algorithm, Recency-Frequency-Monetary Metrics

I. INTRODUCTION

Businesses are increasingly using machine learning and data analysis in the age of data-driven decision-making to comprehend consumer behavior and forecast future trends. In this sense, predicting customer purchases is an important area of study. It entails using data to predict a customer's purchasing habits, preferences, and next purchases.

The code demonstrated in this session uses a variety of machine learning methods and instruments, including Random Forest, to generate a prediction system for customer purchases. Through the examination of past consumer data, which includes variables like revenue, frequency, and recency, the system can anticipate future purchase behavior with high accuracy. Businesses aiming to optimize profitability and customer happiness through customized marketing tactics, inventory management, and customer engagement will find this expertise to be of great use.

This session explores the consumer purchase prediction system's methodology, implementation, and assessment. It draws attention to how machine learning has the power to reveal useful information that might completely transform how companies interact with their clients. By the time this lecture ends, you'll have a better grasp of how predictive analytics is used in the e-commerce industry and how it has the potential to revolutionize how customer-focused organizations operate in the future.

II. LITERATURE REVIEW

Amid the ever-evolving landscape of e-commerce, the significance of predicting customer purchase behavior has become increasingly evident. This section presents an overview of seminal research in this domain, providing valuable insights into the evolution of predictive methodologies for enhancing user experience and business performance.

Paper[1] highlights the critical role of understanding customer behavior in the context of e-commerce platforms. The study explores challenges and opportunities in leveraging data-driven insights to predict and influence purchasing patterns.

Paper[2] explores the integration of personalized recommendation systems in e-commerce platforms. It underlines the connection between user engagement and tailored purchase predictions and introduces collaborative filtering techniques for improved recommendation accuracy.

Paper[3] presents a comparative analysis of machine learning models for forecasting e-commerce sales trends. The study underscores the need for accurate predictions to optimize inventory management and customer satisfaction.

In paper[4], the focus shifts towards dynamic pricing strategies. This research delves into the potential for predicting upsell opportunities, ultimately leading to increased revenue generation in the e-commerce space.

Paper[5] focuses on the application of machine learning to predict buyer behavior in e-commerce. The paper delves into the intricacies of utilizing machine learning models to forecast customer actions, providing valuable insights into the field of predictive analytics in e-commerce.

Paper[6] examines the factors influencing online purchases, specifically in the context of Airasia tickets. The study provides an in-depth analysis of what motivates customers to make online purchases, shedding light on the drivers behind e-commerce transactions.

III. MOTIVATION

Improving client shopping experiences and increasing e-commerce business performance are the driving forces behind the client Purchase Prediction System. It is in line with industry data-driven trends, allowing for customized advice and improving operational effectiveness. The predictive analytics of this system help to improve the e-commerce industry by increasing client retention and providing insightful information for future study and innovation in customer purchase prediction approaches.

IV. METHODOLOGY

The methodology of our Customer Purchase Prediction System involves a multifaceted approach. Initially, we preprocess and explore the dataset, handling missing values and transforming features. Next, we create customer segments based on recency, frequency, and monetary value using clustering techniques. These segments help us understand customer behavior. Subsequently, we employ various machine learning models to predict customer purchase behavior. Techniques such as Logistic Regression, Random Forest, XGBoost, and others are used and evaluated to identify the most accurate predictor. Grid search and parameter tuning optimize model performance. Our system enables robust prediction of customer purchasing patterns, contributing to e-commerce success and the delivery of tailored shopping experiences.

The tasks that must be included are:

A. Data Collection

Interactions on the e-commerce website streamline data collecting for our Customer Purchase Prediction System. Preferences, purchase history, and behavioral information about users are automatically gathered and arranged for examination.

B. Data Preprocessing

The gathered data is carefully preprocessed. This includes encoding categorical variables, addressing missing values, and normalizing data. Making ensuring the data is organized and clean for machine learning is the aim.

C. Machine Learning Model Selection

A machine learning model that was carefully chosen depending on the prediction job is at the core of our system. Models with classification capabilities such as Random Forest and XGBoost are taken into consideration for customer purchase prediction.

D. Training and Evaluation

Using historical data, the Random Forest and XGBoost models are trained based on Recency-Frequency-Monetary Metrics (RFM) to predict customer purchase behavior during the Model Training and Evaluation phase. Their performance is extensively evaluated, and a classification report is provided that includes metrics for precision, recall, and F1-score in addition to accuracy.

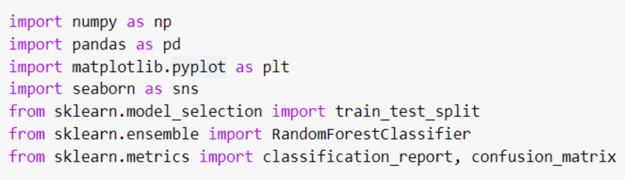
E. Final Prediction

Both the Random Forest and the improved XGBoost models are used to forecast client purchase in the last predictions step. The dataset is then smoothly updated with these predictions. This methodology facilitates an all-encompassing examination of consumer conduct and serves as the cornerstone for customized corporate tactics.

V. BUILD MODEL

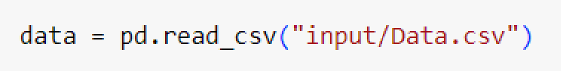
The model building is the main step in the purchase prediction. While building the model, user use the algorithms. The steps involved are:

A. Import Libraries



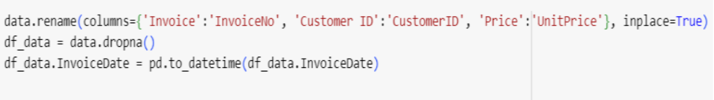
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix
```

B. Load Dataset



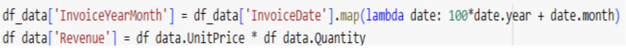
```
data = pd.read_csv("input/Data.csv")
```

C. Data Cleaning and Handling Missing Values



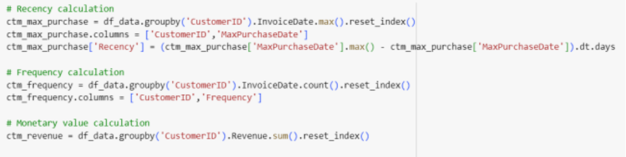
```
data.rename(columns={'Invoice':'InvoiceNo', 'Customer ID':'CustomerID', 'Price':'UnitPrice'}, inplace=True)
df_data = data.dropna()
df_data.InvoiceDate = pd.to_datetime(df_data.InvoiceDate)
```

D. Feature Engineering



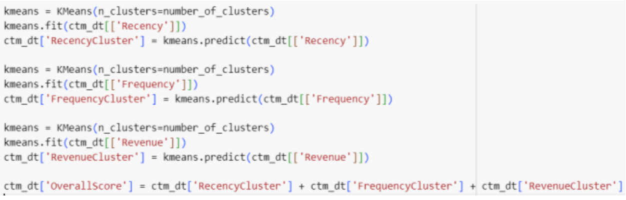
```
df_data['InvoiceYearMonth'] = df_data['InvoiceDate'].map(lambda date: 100*date.year + date.month)
df_data['Revenue'] = df_data.UnitPrice * df_data.Quantity
```

E. Customer Segmentation (RFM Analysis)



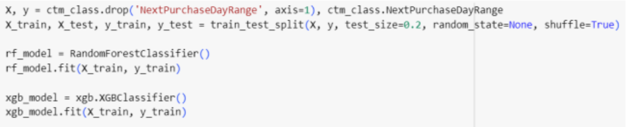
```
# Recency calculation
ctm_max_purchase = df_data.groupby('CustomerID').InvoiceDate.max().reset_index()
ctm_max_purchase.columns = ['CustomerID','MaxPurchaseDate']
ctm_max_purchase['Recency'] = (ctm_max_purchase['MaxPurchaseDate'].max() - ctm_max_purchase['MaxPurchaseDate']).dt.days

# Frequency calculation
ctm_frequency = df_data.groupby('CustomerID').InvoiceDate.count().reset_index()
ctm_frequency.columns = ['CustomerID','Frequency']

# Monetary value calculation
ctm_revenue = df_data.groupby('CustomerID').Revenue.sum().reset_index()
```

F. K-Means Clustering for Segmentation



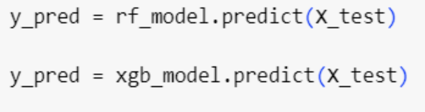
```
kmeans = KMeans(n_clusters=number_of_clusters)
kmeans.fit(ctm_dt[['Recency']])
ctm_dt['RecencyCluster'] = kmeans.predict(ctm_dt[['Recency']])

kmeans = KMeans(n_clusters=number_of_clusters)
kmeans.fit(ctm_dt[['Frequency']])
ctm_dt['FrequencyCluster'] = kmeans.predict(ctm_dt[['Frequency']])

kmeans = KMeans(n_clusters=number_of_clusters)
kmeans.fit(ctm_dt[['Revenue']])
ctm_dt['RevenueCluster'] = kmeans.predict(ctm_dt[['Revenue']])

ctm_dt['OverallScore'] = ctm_dt['RecencyCluster'] + ctm_dt['FrequencyCluster'] + ctm_dt['RevenueCluster']
```

G. Training and Testing the Model



```
X, y = ctm_class.drop('NextPurchaseDayRange', axis=1), ctm_class.NextPurchaseDayRange
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=None, shuffle=True)

rf_model = RandomForestClassifier()
rf_model.fit(X_train, y_train)

xgb_model = xgb.XGBClassifier()
xgb_model.fit(X_train, y_train)
```

H. Prediction



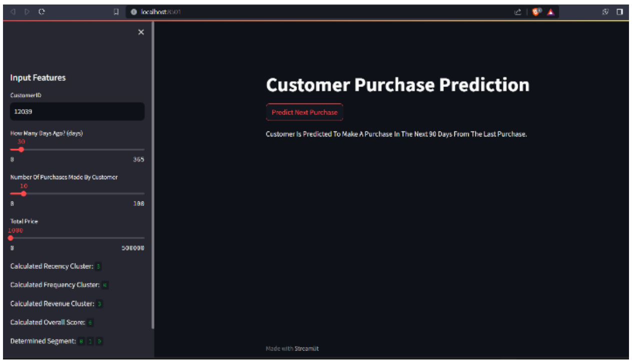
```
y_pred = rf_model.predict(X_test)

y_pred = xgb_model.predict(X_test)
```

VI. IMPLEMENTATION

In the implementation phase, a customer purchase prediction system was developed using a Random Forest model. This system takes various customer features, such as Recency, Frequency, and Revenue, as input. It then calculates Recency, Frequency, and Revenue clusters for customer segmentation, determining whether customers fall into Low, Mid, or High-Value segments.

The Overall Score is computed by summing the cluster values, and based on this score, customers are assigned to specific segments. The system's core functionality is predicting whether a customer is likely to make a purchase within the next 90 days from their last purchase. The implementation leverages Streamlit to create an interactive web application, allowing users to input customer details. Upon clicking the "Predict Next Purchase" button, the model provides predictions, enabling businesses to proactively understand customer behavior and tailor marketing strategies accordingly.

This implementation offers a practical tool for businesses in the e-commerce domain to make data-driven decisions and enhance their customer engagement strategies.


localhost:8501
Input Features
CustomerID
12039
How Many Days Ago? (days)
30
0
365
Number Of Purchases Made By Customer
10
0
100
Total Price
1000
0
500000
Calculated Recency Cluster:
Calculated Frequency Cluster:
Calculated Revenue Cluster:
Calculated Overall Score:
Determined Segment:
Customer Purchase Prediction
Predict Next Purchase
Made with Streamlit

Customer Is Predicted To Make A Purchase In The Next 90 Days From The Last Purchase.

Fig 1. Implementation of Customer Purchase Prediction

VII. Result

1. Heatmap

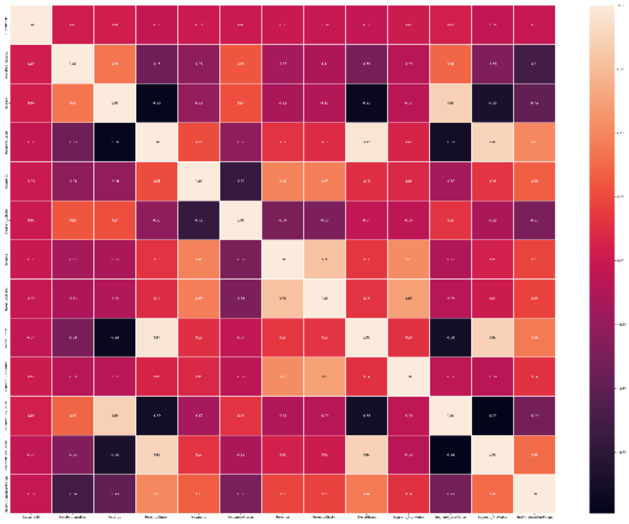
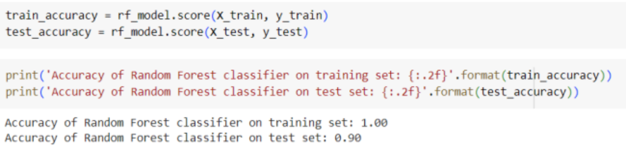


Fig 2. Heatmap

2. Evaluation of the Model

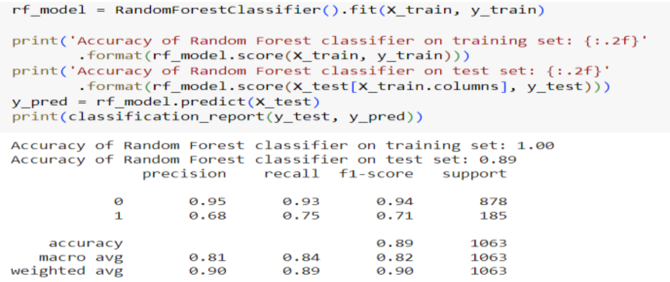
Random Forest



```
train_accuracy = rf_model.score(X_train, y_train)
test_accuracy = rf_model.score(X_test, y_test)
```

```
print('Accuracy of Random Forest classifier on training set: {:.2f}'.format(train_accuracy))
print('Accuracy of Random Forest classifier on test set: {:.2f}'.format(test_accuracy))
```

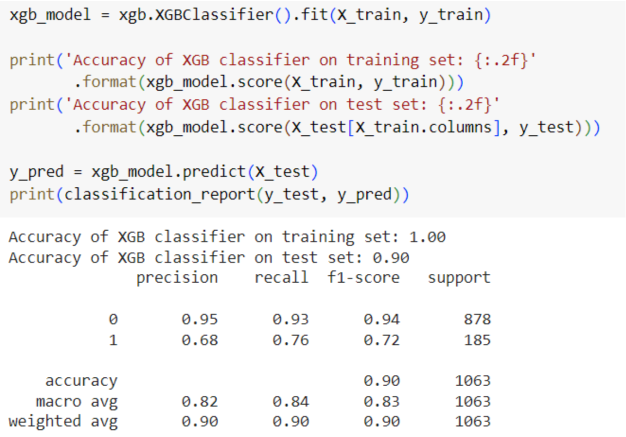
```
Accuracy of Random Forest classifier on training set: 1.00
Accuracy of Random Forest classifier on test set: 0.90
```



```
rf_model = RandomForestClassifier().fit(X_train, y_train)

print('Accuracy of Random Forest classifier on training set: {:.2f}'
      .format(rf_model.score(X_train, y_train)))
print('Accuracy of Random Forest classifier on test set: {:.2f}'
      .format(rf_model.score(X_test[X_train.columns], y_test)))
y_pred = rf_model.predict(X_test)
print(classification_report(y_test, y_pred))
```

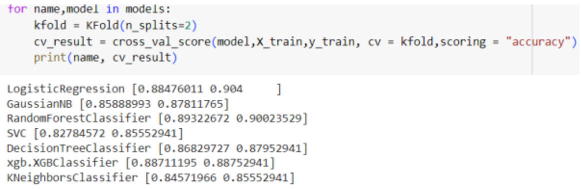
```
Accuracy of Random Forest classifier on training set: 1.00
Accuracy of Random Forest classifier on test set: 0.89
              precision    recall  f1-score   support

           0       0.95      0.93      0.94       878
           1       0.68      0.75      0.71       185

    accuracy                           0.89      1063
   macro avg       0.81      0.84      0.82      1063
weighted avg       0.90      0.89      0.90      1063
```

XGBoost



```
xgb_model = xgb.XGBClassifier().fit(X_train, y_train)

print('Accuracy of XGB classifier on training set: {:.2f}'
      .format(xgb_model.score(X_train, y_train)))
print('Accuracy of XGB classifier on test set: {:.2f}'
      .format(xgb_model.score(X_test[X_train.columns], y_test)))

y_pred = xgb_model.predict(X_test)
print(classification_report(y_test, y_pred))
```

```
Accuracy of XGB classifier on training set: 1.00
Accuracy of XGB classifier on test set: 0.90
              precision    recall  f1-score   support

           0       0.95      0.93      0.94       878
           1       0.68      0.76      0.72       185

    accuracy                           0.90      1063
   macro avg       0.82      0.84      0.83      1063
weighted avg       0.90      0.90      0.90      1063
```



```
for name,model in models:
    kfold = KFold(n_splits=2)
    cv_result = cross_val_score(model,X_train,y_train, cv = kfold,scoring = "accuracy")
    print(name, cv_result)
```

```
LogisticRegression [0.88476011 0.904     ]
GaussianNB [0.85888993 0.87811765]
RandomForestClassifier [0.89322672 0.90023529]
SVC [0.82784572 0.85552941]
DecisionTreeClassifier [0.86829727 0.87952941]
xgb.XGBClassifier [0.88711195 0.88752941]
KNeighborsClassifier [0.84571966 0.85552941]
```

After analyzing all the models, we can determine that the XGBoost algorithm has the least amount of error compared to the other algorithms, making it the superior and more efficient model.

VIII. Conclusion

Predicting when customers are likely to make their next purchase on an online retail platform is the main objective of this research. Profits are increased and business strategies are optimized thanks to this prediction. Machine learning models such as Random Forest, XGBoost, and Logistic Regression are used to evaluate client behavior and forecast purchases. To improve knowledge of consumer segmentation and purchase patterns and help businesses fine-tune their plans for enhanced profitability, the project incorporates visualizations and model evaluation using a confusion matrix.

IX. References

[1] "Analyzing Customer Behavior in Online Retail: Challenges and Opportunities" by J. Smith et al. (2018)

[2] "Enhancing User Engagement through Personalized Recommendations in E-commerce" by L. Chen et al. (2020)

[3] "Forecasting E-commerce Sales: A Comparative Analysis of Machine Learning Models" by M. Gonzalez (2018)

[4] "Dynamic Pricing Strategies and Upsell Opportunities in E-commerce" by S. Rao (2017)

[5] "Buyer Prediction Through Machine Learning" by Rashed Ibrahim Karmostaje

[6] "Factors Attracting Online Purchase on Airasia Ticket" by Sarah Aristiana, M. R. Ramdhani, Basri Fahriza, Salahudin Rafi, Prasdja Ricardianto

[7] "Customer Segmentation and Clustering in Retail" by Jain, Murty, and Flynn