SMARTBRIDGE
Let's Bridge the Gap
a Veranda Enterprise

Smart Internz

Final Project Report Template

Ecommerce Shipping Prediction Using Machine Learning

1. Introduction

1.1. Project overviews

The current state of ecommerce shipping processes is fraught with inefficiencies that result in delayed deliveries and increased costs. Predicting delivery times accurately remains a significant challenge due to the multitude of influencing factors, including weather conditions, traffic, and other external variables. These challenges hinder ecommerce platforms from providing reliable delivery estimates, ultimately affecting customer satisfaction and loyalty.

The impact of these inefficiencies is substantial, as delayed deliveries can lead to increased operational costs and a decline in customer can result in negative reviews and a potential loss of future business. By improving the accuracy of delivery time predictions, ecommerce trust. When customers receive inaccurate delivery estimates, it diminishes their overall shopping experience, which platforms can enhance their operational efficiency, reduce shipping delays, and lower costs. Accurate predictions will not only improve customer satisfaction by providing dependable delivery estimates but also strengthen the competitive position of the ecommerce platform in the market.

Ecommerce shipping prediction is a crucial aspect of online retail, as it involves estimating whether a product will be delivered on time. This estimation is based on various factors, such as the origin and destination of the package, the shipping method selected by the customer, the carrier used for shipping, and potential delays or issues that may arise during the shipping process. By utilizing machine learning models, accurate predictions about shipping times can be made based on historical data and real-time updates from carriers.

1.2. Objectives

- Accurate Delivery Time Predictions:

The primary objective of this project is to predict whether an ecommerce shipment will be delivered on time using machine learning algorithms. By analyzing various factors influencing delivery times, the goal is to enhance the precision of these predictions, thereby providing more reliable delivery estimates to customers.

- Enhanced Customer Experience:

The project aims to improve the overall customer experience. By providing accurate and reliable delivery estimates, customers can plan better and have a more satisfying shopping experience. Reliable delivery information builds customer trust and loyalty, encouraging repeat business and positive reviews.

- Reduction of Delivery Delays:

The project seeks to significantly reduce delivery delays by accurately predicting potential issues before they arise. By anticipating delays due to factors like traffic, weather, or carrier-specific issues, the project aims to minimize the occurrence of late deliveries, thereby ensuring timely delivery of products to customers.

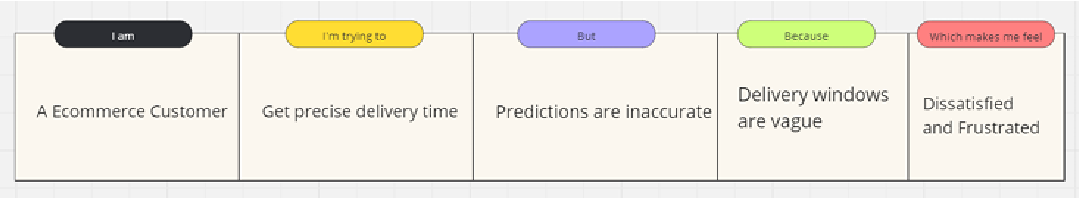
The objectives of this project focus on leveraging machine learning to achieve accurate delivery time predictions, improving shipping logistics, reducing delivery delays and costs, and enhancing the overall customer experience.

2. Project Initialization and Planning Phase

2.1. Define Problem Statement

Problem Statement 1: Customer Frustration with Delivery Estimates

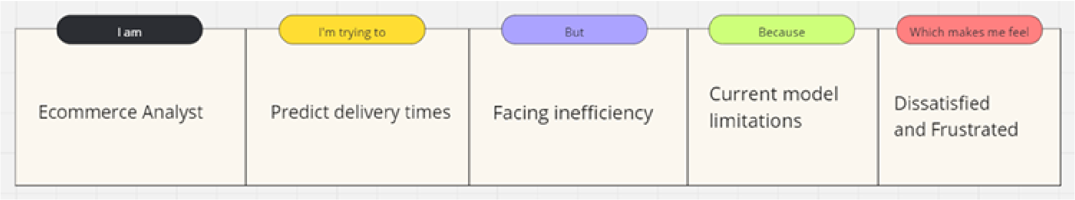
Customers are increasingly frustrated with the vague and often inaccurate delivery windows provided by online stores. This lack of precision leads to missed expectations and overall dissatisfaction. When customers place an order, they want to know exactly when their product will arrive so they can plan accordingly. However, the current delivery time estimates provided by many ecommerce platforms do not reflect real-time conditions and often fail to account for variables such as weather, traffic, and carrier delays. This uncertainty makes customers feel frustrated and dissatisfied, as they are unable to rely on the information given to them regarding their purchases.



I am	I'm trying to	But	Because	Which makes me feel
A Ecommerce Customer	Get precise delivery time	Predictions are inaccurate	Delivery windows are vague	Dissatisfied and Frustrated

Problem Statement 2: Inefficient Shipping Logistics and High Costs

Ecommerce platforms are struggling with inefficient shipping logistics, which lead to delayed deliveries and increased operational costs. These inefficiencies stem from the inability to accurately predict delivery times by considering various influencing factors such as order history, real-time traffic, weather conditions, and carrier-specific issues. Without a system that integrates these elements, platforms face challenges in optimizing their shipping processes. This lack of accurate predictions results in inefficiency and higher costs, ultimately affecting both the business and customer satisfaction.



I am	I'm trying to	But	Because	Which makes me feel
Ecommerce Analyst	Predict delivery times	Facing inefficiency	Current model limitations	Dissatisfied and Frustrated

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	An Ecommerce Customer	Know when my order will arrive	The delivery times are vague and inaccurate	They don't reflect real-time conditions	Frustrated and dissatisfied
PS-2	An Ecommerce Analyst	Improve shipping efficiency and customer satisfaction	I can't predict delivery times accurately	I lack a system that considers various real-time factors	Inefficient Dissatisfied and costly

2.2. Project Proposal (Proposed Solution)

The proposed solution involves harnessing advanced machine learning techniques to predict delivery times with precision. By analysing a diverse array of factors that influence delivery schedules, including historical data and real-time updates, this approach aims to provide ecommerce platforms with reliable estimates.

Approach: -

The project will begin with thorough data preprocessing steps to ensure data quality and consistency. This includes handling categorical variables, managing outliers, and addressing class imbalance using the Synthetic Minority Over-sampling Technique (SMOTE). These steps are crucial for preparing the dataset for meaningful analysis and modelling.

Extensive exploratory data analysis (EDA) will be conducted to uncover insights and patterns within the data. This phase will inform the selection and implementation of machine learning models, such as Logistic Regression, XGBoost, Ridge Classifier, KNN, Random Forest, and SVM. Each model will undergo rigorous hyperparameter tuning to optimize performance and ensure accurate delivery time predictions.

Key Features of the Solution:

- Implementation of Diverse Machine Learning Models: The solution will deploy a range of models known for their effectiveness in predictive analytics, allowing for a comprehensive comparison to identify the best-performing model.
- Detailed Exploratory Data Analysis: Through thorough EDA, the project aims to gain deep insights into the dataset, uncovering hidden patterns that could enhance predictive accuracy.
- Handling Data Challenges: Techniques like SMOTE will be employed to address class imbalance, ensuring that the models are trained on balanced data representative of real-world scenarios.
- Optimized Model Performance: Hyperparameter tuning will be carried out to fine-tune each model's parameters, maximizing their predictive capabilities and ensuring robust performance in predicting delivery times.
- Model Comparison and Selection: The project will systematically compare the performance of various models to select the most effective one for accurate delivery time predictions. This approach aims to provide ecommerce platforms with a reliable tool to enhance operational efficiency and customer satisfaction.

The proposed solution leverages cutting-edge machine learning methodologies to address the challenges faced by ecommerce platforms in predicting delivery times accurately. By implementing a robust framework that integrates data preprocessing, model selection, and optimization techniques, the project aims to deliver tangible improvements in shipping logistics and overall customer experience.

2.3. Initial Project Planning

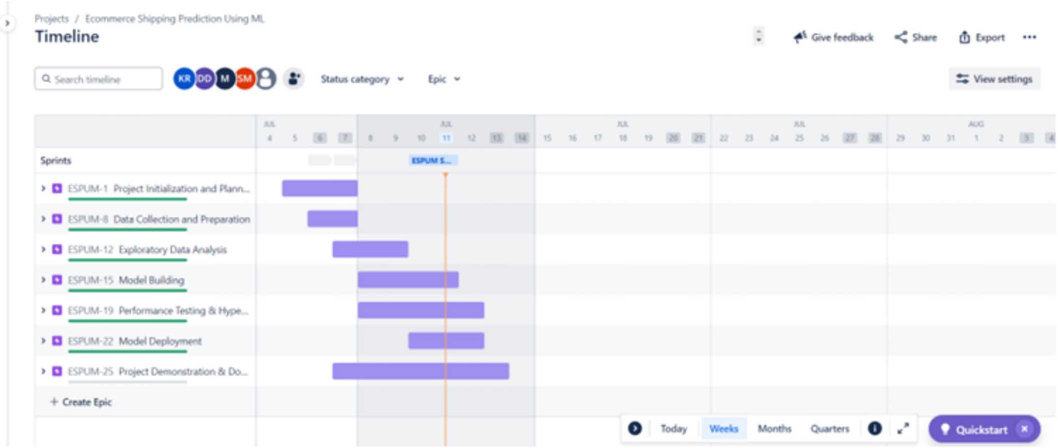
The project was divided into several sprints, each meticulously planned to achieve specific objectives within set timeframes. Each sprint was structured around functional requirements (epics) that defined the overarching goals and tasks to be completed. These tasks were broken down into user stories, each assigned a story point value, priority, team members responsible, and planned start and end dates.

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members	Sprint Start Date	Sprint End Date (Planned)
Sprint-1	Project Initialization and Planning	ESPUM -6	Project Proposal (Proposed Solution)	2	High	Murari	July 05, 2024	July 08, 2024
Sprint-1	Project Initialization and Planning	ESPUM -2	Defining / Specify the business problem	1	High	Murari	July 05, 2024	July 08, 2024
Sprint-1	Project Initialization and Planning	ESPUM -5	Social or Business Impact.	2	Low	Divya Darshini	July 05, 2024	July 08, 2024
Sprint-1	Project Initialization and Planning	ESPUM -4	Literature Survey	2	Medium	Ravinder	July 05, 2024	July 08, 2024
Sprint-6	Project Demonstration & Documentation	ESPUM -27	Project Documentation	1	Medium	Meghani	July 07, 2024	July 08, 2024

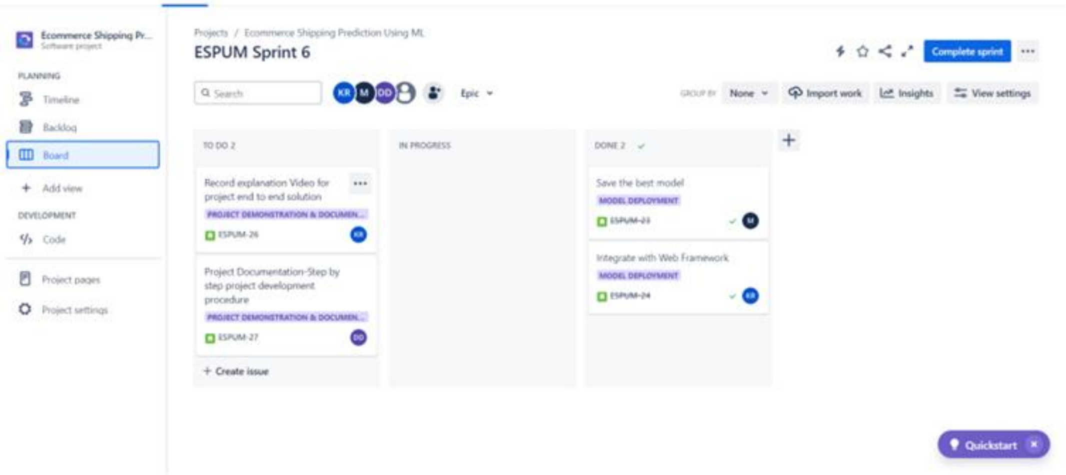
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members	Sprint Start Date	Sprint End Date (Planned)
Sprint-2	Data Collection and Preparation	ESPUM -10	Data Collection	1	Medium	Meghani	July 06, 2024	July 08, 2024
Sprint-2	Data Collection and Preparation	ESPUM -11	Data Preparation	3	High	Meghani, Ravinder, Murari, Divya Darshini	July 06, 2024	July 08, 2024
Sprint-3	Exploratory Data Analysis	ESPUM -13	Data Quality Report	2	High	Divya Darshini	July 07, 2024	July 09, 2024
Sprint-3	Exploratory Data Analysis	ESPUM -14	Visual Analysis	2	Medium	Ravinder	July 07, 2024	July 09, 2024
Sprint-4	Model Building	ESPUM -16	Feature Selection	3	High	Murari	July 08, 2024	July 08, 2024
Sprint-4	Model Building	ESPUM -17	Training the model in multiple algorithms	2	High	Meghani, Ravinder	July 08, 2024	July 010, 2024
Sprint-4	Model Building	ESPUM -18	Testing the models	2	High	Ravinder	July 09, 2024	July 11, 2024
Sprint-5	Performance Testing & Hyperparameter Tuning	ESPUM -20	Testing model with multiple evaluation metrics	3	High	Divya Darshini	July 08, 2024	July 12, 2024
Sprint-5	Performance Testing & Hyperparameter Tuning	ESPUM -21	Comparing model accuracy before & after applying hyperparameter tuning	3	High	Murari	July 11, 2024	July 12, 2024
Sprint-6	Model Deployment	ESPUM -4	Save the best model	2	Medium	Murari	July 10, 2024	July 12, 2024

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members	Sprint Start Date	Sprint End Date (Planned)
Sprint-6	Model Deployment	ESPUM -4	Integrate with Web Framework	3	High	Ravinder	July 11, 2024	July 12, 2024
Sprint-7	Project Demonstration & Documentation	ESPUM -4	Record explanation Video for project end to end solution	2	Low	Meghani, Ravinder, Murari, Divya Darshini	July 12, 2024	July 12, 2024
Sprint-7	Project Demonstration & Documentation	ESPUM -4	Project Documentation-Step by step project development procedure	2	Medium	Meghani, Ravinder, Murari, Divya Darshini	July 07, 2024	July 13, 2024

Tracking project using jira software:



ESPUM-1 Project Initialization and Plann...
ESPUM-8 Data Collection and Preparation
ESPUM-12 Exploratory Data Analysis
ESPUM-15 Model Building
ESPUM-19 Performance Testing & Hype...
ESPUM-22 Model Deployment
ESPUM-25 Project Demonstration & Do...



TO DO 2

Record explanation Video for project end to end solution
PROJECT DEMONSTRATION & DOCUMEN...
ESPUM-26

Project Documentation-Step by step project development procedure
PROJECT DEMONSTRATION & DOCUMEN...
ESPUM-27

IN PROGRESS

DONE 2 ✓

Save the best model
MODEL DEPLOYMENT
ESPUM-23

Integrate with Web Framework
MODEL DEPLOYMENT
ESPUM-24

3. Data Collection and Preprocessing Phase

3.1. Data Collection Plan and Raw Data Sources Identified

The data collection plan and identified raw data sources are pivotal components of the project focused on enhancing ecommerce operations through predictive modelling. Here's a concise overview:

Data Collection Plan:

The project involves analysing an ecommerce dataset that includes critical variables such as shipping mode, customer details, product specifics, and discount offers. The primary objective is to build a predictive model capable of determining whether an order will be delivered on time. Key steps in the data collection plan include:

- E-commerce Transaction Records and Customer Databases: Utilizing detailed information on orders, customer demographics, and purchasing behavior to inform predictive modeling.
- Exploration of Online Data Sources: Investigating platforms like Kaggle and GitHub to identify relevant datasets that complement project objectives and dataset requirements.

Raw Data Sources Identified:

The project has identified datasets sourced primarily from Kaggle and UCI, renowned repositories for data science competitions and datasets:

➔ **Kaggle Dataset:**

- Source: E-Commerce Shipping Data
- Description: Contains essential variables crucial for machine learning analyses, including shipping mode, customer details, product specifics, and discount offers.
- Location/URL: https://www.kaggle.com/datasets/prachi13/customer-analytics/data
- Format: CSV
- Size: 431 kB
- Access Permissions: Public access

These raw data sources provide foundational data for the project, facilitating comprehensive preprocessing, feature engineering, model building, and evaluation stages. The goal is to develop a robust predictive model that enhances ecommerce efficiency by accurately predicting delivery times and improving customer satisfaction.

3.2. Data Quality Report

The Data Quality Report Template will summarize data quality issues from the selected source, including severity levels and resolution plans. It will aid in systematically identifying and rectifying data discrepancies.

Data Source	Data Quality Issue	Severity	Resolution Plan
Kaggle Dataset	Missing values in the(shipping mode, customer details, product specifics, and discount offers)	Moderate	Use mean/median imputation.
Kaggle Dataset	Categorical data in the dataset	Moderate	Encoding has to be done in the data.
Kaggle Dataset	Outliers in attributes.	High	Outliers are detected using quartiles and replaced with the standard value(median).

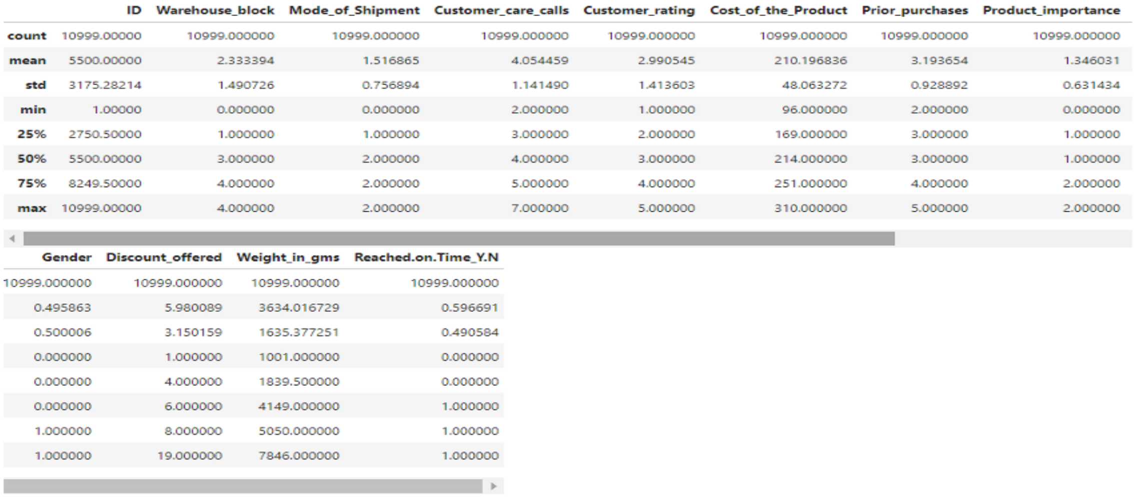
3.3. Data Exploration and Preprocessing

Data exploration and preprocessing are foundational steps in the data analysis pipeline, essential for ensuring data quality and preparing datasets for meaningful analysis. Here's a descriptive overview:

Data Overview:

The dataset comprises 10,999 rows and 12 columns, providing a comprehensive view of ecommerce transaction records. Descriptive statistics reveal key insights into the numerical attributes, highlighting central tendencies and dispersion.

Descriptive statistics:



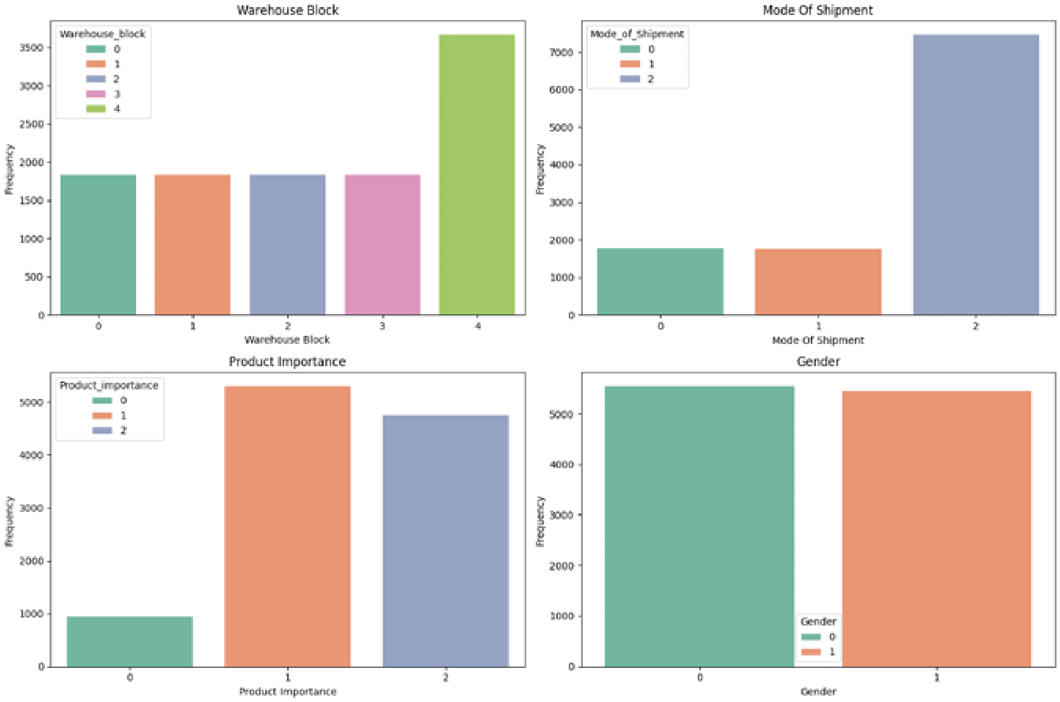
	ID	Warehouse_block	Mode_of_Shipment	Customer_care_calls	Customer_rating	Cost_of_the_Product	Prior_purchases	Product_importance
count	10999.00000	10999.000000	10999.000000	10999.000000	10999.000000	10999.000000	10999.000000	10999.000000
mean	5500.00000	2.333394	1.516865	4.054459	2.990545	210.196836	3.193654	1.346031
std	3175.28214	1.490726	0.756894	1.141490	1.413603	48.063272	0.928892	0.631434
min	1.00000	0.000000	0.000000	2.000000	1.000000	96.000000	2.000000	0.000000
25%	2750.50000	1.000000	1.000000	3.000000	2.000000	169.000000	3.000000	1.000000
50%	5500.00000	3.000000	2.000000	4.000000	3.000000	214.000000	3.000000	1.000000
75%	8249.50000	4.000000	2.000000	5.000000	4.000000	251.000000	4.000000	2.000000
max	10999.00000	4.000000	2.000000	7.000000	5.000000	310.000000	5.000000	2.000000

Gender	Discount_offered	Weight_in_gms	Reached.on.Time_Y.N
10999.000000	10999.000000	10999.000000	10999.000000
0.495863	5.980089	3634.016729	0.596691
0.500006	3.150159	1635.377251	0.490584
0.000000	1.000000	1001.000000	0.000000
0.000000	4.000000	1839.500000	0.000000
0.000000	6.000000	4149.000000	1.000000
1.000000	8.000000	5050.000000	1.000000
1.000000	19.000000	7846.000000	1.000000

Univariate Analysis:

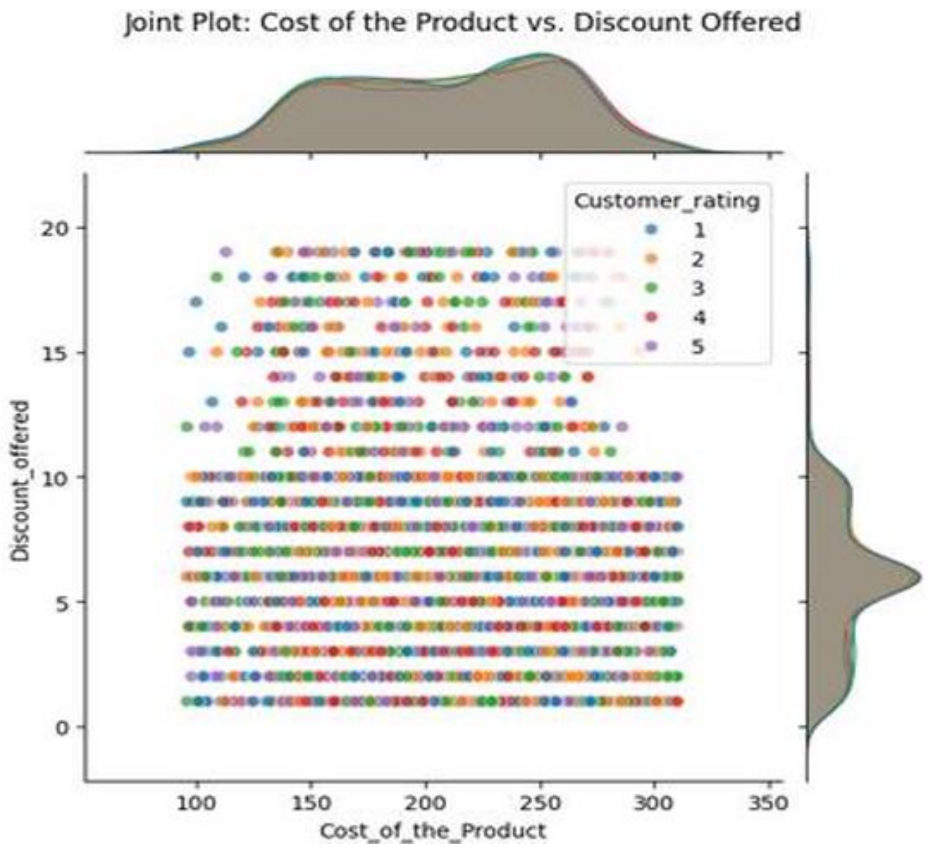
- Categorical Attributes: Analyzing categorical variables such as shipping mode and product categories to understand their distributions and frequencies.

- Numerical Attributes: Assessing numerical variables like order amounts and delivery times to identify outliers and understand their distributions.



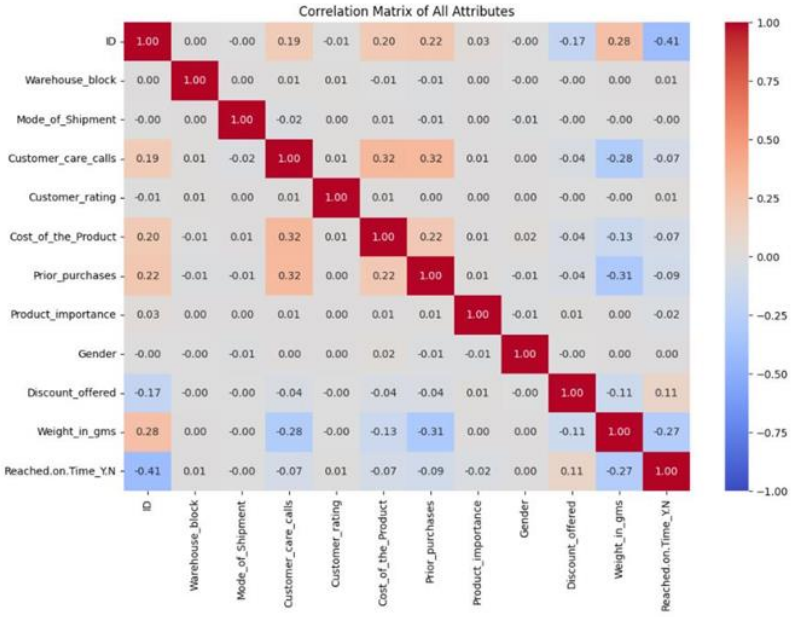
Bivariate Analysis:

Examining relationships between pairs of variables to uncover correlations and dependencies. This analysis helps in understanding how variables interact and influence each other.



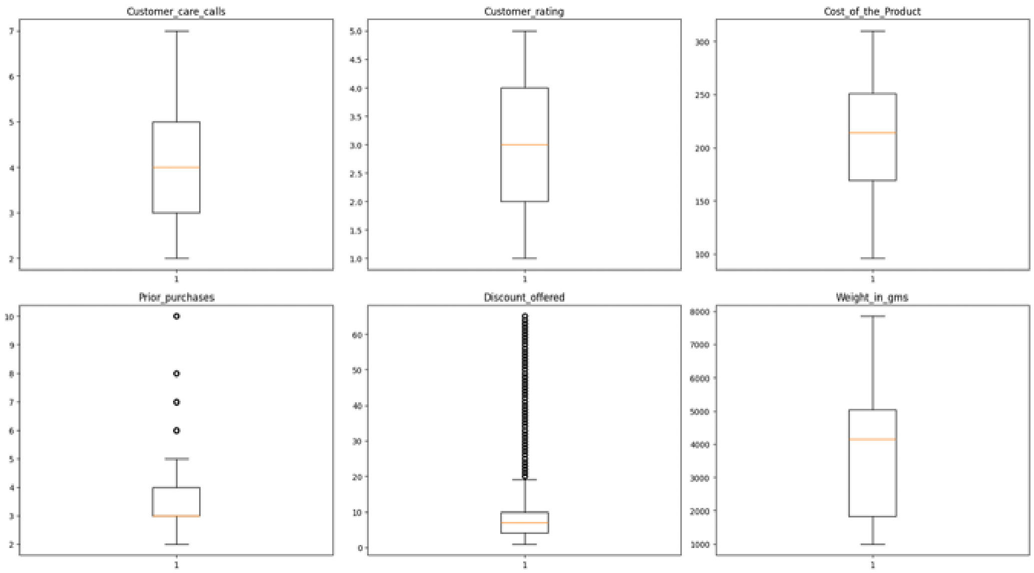
Multivariate Analysis:

Exploring interactions among multiple variables simultaneously to uncover complex patterns and relationships within the dataset. This analysis provides deeper insights into the interdependencies of various factors.



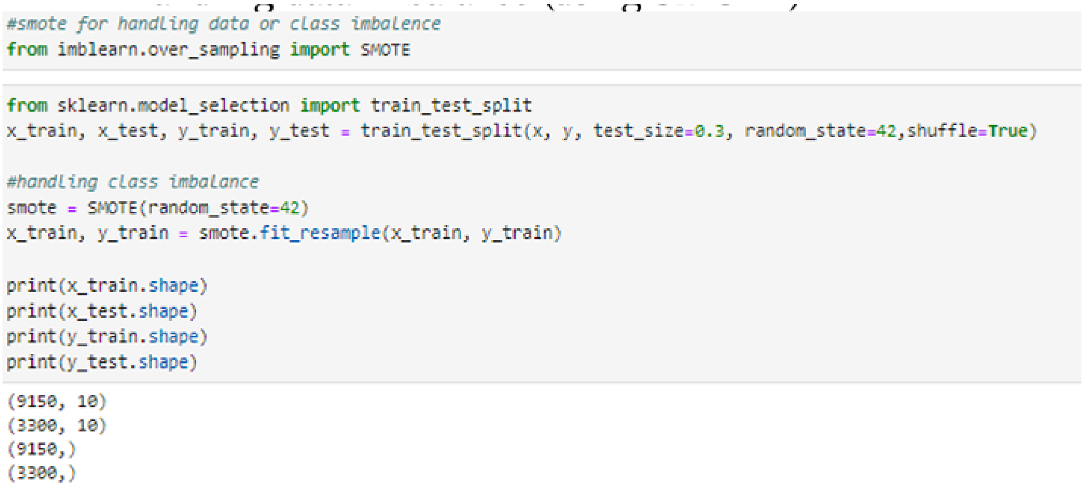
Outliers and Anomalies:

Detecting outliers and anomalies that deviate significantly from the expected patterns. Strategies involve replacing outliers with standardized values, such as median values, to mitigate their impact on analysis and modeling.



Data Preprocessing:

- Handling Missing Data: Addressing missing values through techniques like imputation to ensure completeness and accuracy in the dataset.
- Data Transformation: Converting categorical values into numerical representations suitable for machine learning algorithms.
- Feature Engineering: Enhancing dataset quality by addressing data imbalance using techniques like Synthetic Minority Over-sampling Technique (SMOTE), ensuring a balanced representation for robust model training.



```
#smote for handling data or class imbalence
from imblearn.over_sampling import SMOTE
```

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=42,shuffle=True)

#handling class imbalance
smote = SMOTE(random_state=42)
x_train, y_train = smote.fit_resample(x_train, y_train)

print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(9150, 10)
(3300, 10)
(9150,)
(3300,)
```

Data Transformation and Saving Processed Data:

Converting data into a format suitable for analysis and model building, ensuring that all preprocessing steps are documented and reproducible. The processed data is saved for further analysis and model development.

Data exploration and preprocessing are critical stages that lay the groundwork for effective data analysis. By understanding the dataset's characteristics, addressing quality issues, and transforming data for optimal model performance, these steps ensure that insights derived are accurate, reliable, and actionable, ultimately supporting informed decision-making in ecommerce logistics and operations.

4. Model Development Phase

4.1. Feature Selection Report

Feature selection is a critical step in the machine learning pipeline that involves identifying and selecting the most relevant features (or variables) from the dataset for use in model training and prediction.

Selected features: Warehouse_block, Mode_of_Shipment, Customer_care_calls, Customer_rating, Cost_of_the_Product, Prior_purchases, Product_importance, Gender, Discount_offered, Weight_in_gms.

4.2. Model Selection Report

This involves in exploring different algorithms or models and selecting a set of suitable models for the prediction based on type of dataset, what wea re going to predict and etc.

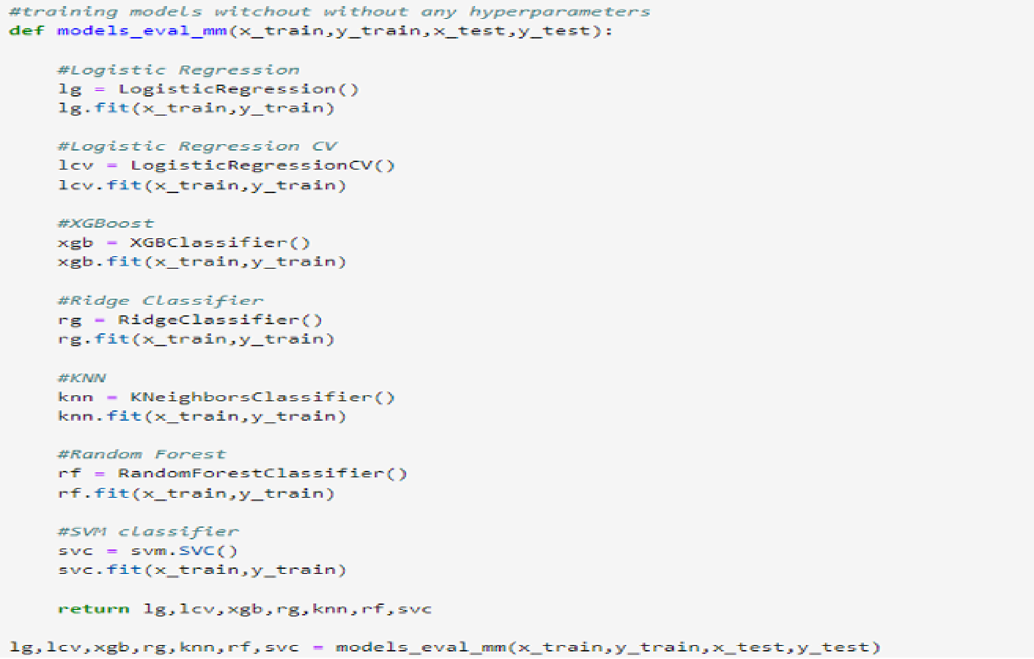
Here we considered 7 models for the training:

- Logistic Regression
- Logistic Regression with Cross Validation (LogisticRegressionCV)
- XGBoost Classifier
- Ridge Classifier
- K-Nearest Neighbors Classifier
- Random Forest Classifier
- Support Vector Machine Classifier (SVC).

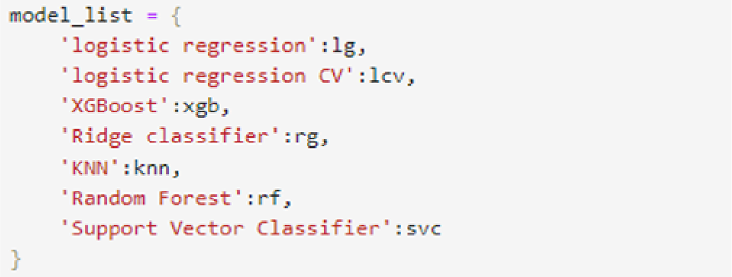
These models will be trained, and the results will be considered for choosing for best model for prediction or leverage in the project.

4.3. Initial Model Training Code, Model Validation and Evaluation Report

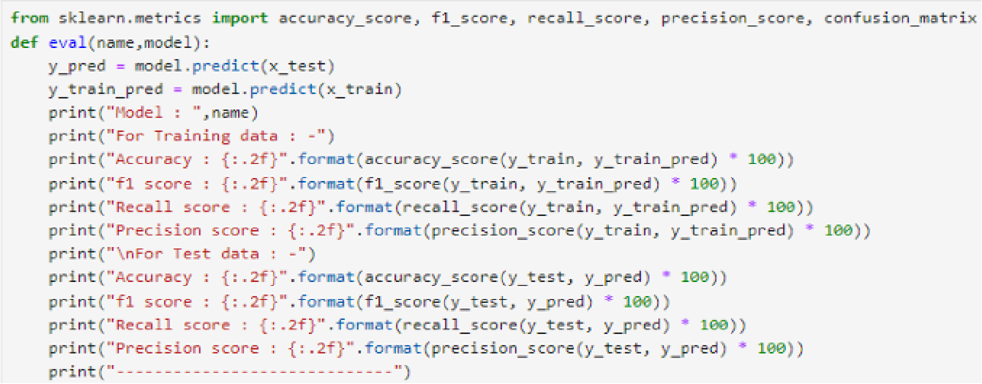
- Initial Model Training Code:



```
#training models witchout without any hyperparameters
def models_eval_mm(x_train,y_train,x_test,y_test):

    #Logistic Regression
    lg = LogisticRegression()
    lg.fit(x_train,y_train)

    #Logistic Regression CV
    lcv = LogisticRegressionCV()
    lcv.fit(x_train,y_train)

    #XGBoost
    xgb = XGBClassifier()
    xgb.fit(x_train,y_train)

    #Ridge Classifier
    rg = RidgeClassifier()
    rg.fit(x_train,y_train)

    #KNN
    knn = KNeighborsClassifier()
    knn.fit(x_train,y_train)

    #Random Forest
    rf = RandomForestClassifier()
    rf.fit(x_train,y_train)

    #SVM classifier
    svc = svm.SVC()
    svc.fit(x_train,y_train)

    return lg,lcv,xgb,rg,knn,rf,svc

lg,lcv,xgb,rg,knn,rf,svc = models_eval_mm(x_train,y_train,x_test,y_test)
```

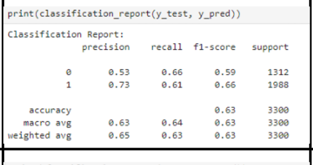
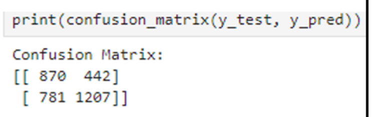
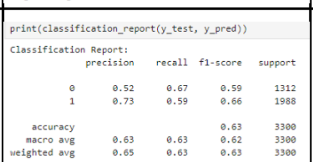
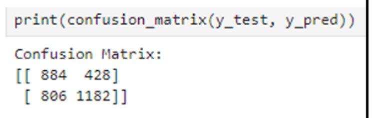
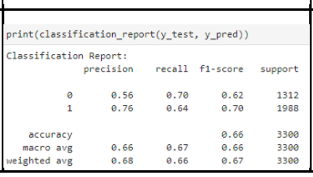
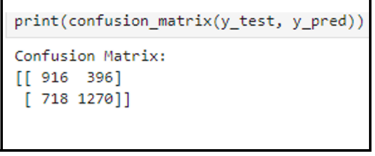
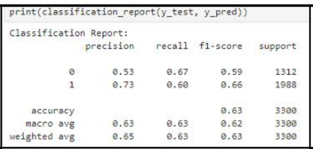
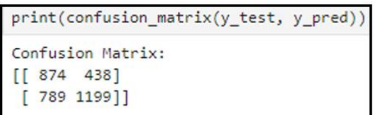
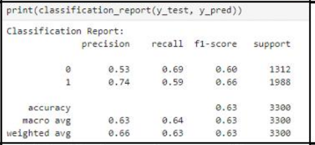
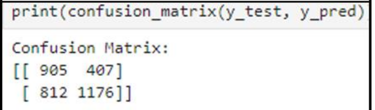
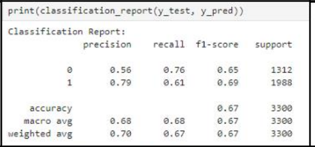
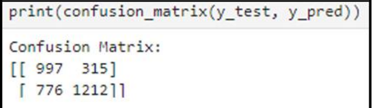
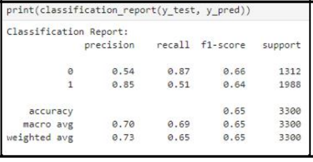
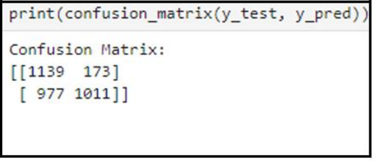


```
model_list = {
    'logistic regression':lg,
    'logistic regression CV':lcv,
    'XGBoost':xgb,
    'Ridge classifier':rg,
    'KNN':knn,
    'Random Forest':rf,
    'Support Vector Classifier':svc
}
```



```
from sklearn.metrics import accuracy_score, f1_score, recall_score, precision_score, confusion_matrix
def eval(name,model):
    y_pred = model.predict(x_test)
    y_train_pred = model.predict(x_train)
    print("Model : ",name)
    print("For Training data : -")
    print("Accuracy : {:.2f}".format(accuracy_score(y_train, y_train_pred) * 100))
    print("f1 score : {:.2f}".format(f1_score(y_train, y_train_pred) * 100))
    print("Recall score : {:.2f}".format(recall_score(y_train, y_train_pred) * 100))
    print("Precision score : {:.2f}".format(precision_score(y_train, y_train_pred) * 100))
    print("\nFor Test data : -")
    print("Accuracy : {:.2f}".format(accuracy_score(y_test, y_pred) * 100))
    print("f1 score : {:.2f}".format(f1_score(y_test, y_pred) * 100))
    print("Recall score : {:.2f}".format(recall_score(y_test, y_pred) * 100))
    print("Precision score : {:.2f}".format(precision_score(y_test, y_pred) * 100))
    print("-----------------------------")
```

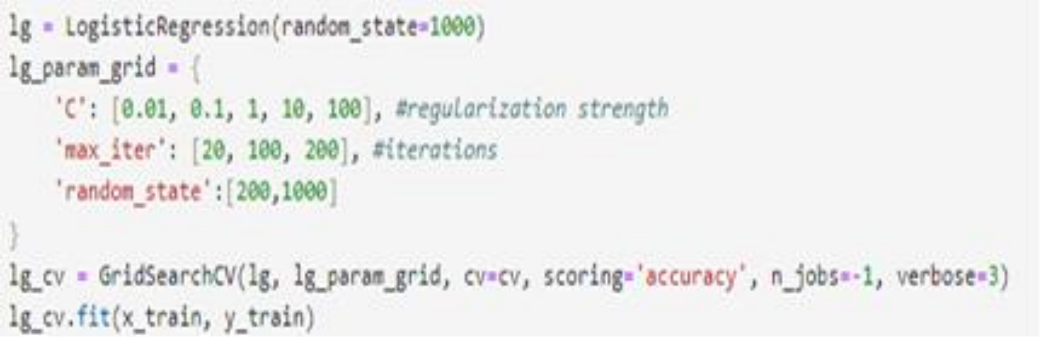
- Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix
Logistic Regression	 print(classification_report(y_test, y_pred)) Classification Report: precision recall f1-score support 0 0.53 0.66 0.59 1312 1 0.73 0.61 0.66 1988 accuracy 0.63 3300 macro avg 0.63 0.64 0.63 3300 weighted avg 0.65 0.63 0.63 3300	62.94	 print(confusion_matrix(y_test, y_pred)) Confusion Matrix: [[870 442] [781 1207]]
Logistic Regression CV	 print(classification_report(y_test, y_pred)) Classification Report: precision recall f1-score support 0 0.52 0.67 0.59 1312 1 0.73 0.59 0.66 1988 accuracy 0.63 3300 macro avg 0.63 0.63 0.62 3300 weighted avg 0.65 0.63 0.63 3300	62.61	 print(confusion_matrix(y_test, y_pred)) Confusion Matrix: [[884 428] [806 1182]]
XGBoost Classifier	 print(classification_report(y_test, y_pred)) Classification Report: precision recall f1-score support 0 0.56 0.70 0.62 1312 1 0.76 0.64 0.70 1988 accuracy 0.66 3300 macro avg 0.66 0.67 0.66 3300 weighted avg 0.68 0.66 0.67 3300	66.24	 print(confusion_matrix(y_test, y_pred)) Confusion Matrix: [[916 396] [718 1270]]
Ridge Classifier	 print(classification_report(y_test, y_pred)) Classification Report: precision recall f1-score support 0 0.53 0.67 0.59 1312 1 0.73 0.60 0.66 1988 accuracy 0.63 3300 macro avg 0.63 0.63 0.62 3300 weighted avg 0.65 0.63 0.63 3300	62.82	 print(confusion_matrix(y_test, y_pred)) Confusion Matrix: [[874 438] [789 1199]]
K-Nearest Neighbors Classifier	 print(classification_report(y_test, y_pred)) Classification Report: precision recall f1-score support 0 0.53 0.69 0.60 1312 1 0.74 0.59 0.66 1988 accuracy 0.63 3300 macro avg 0.63 0.64 0.63 3300 weighted avg 0.66 0.63 0.63 3300	63.06	 print(confusion_matrix(y_test, y_pred) Confusion Matrix: [[905 407] [812 1176]]
Random Forest Classifier	 print(classification_report(y_test, y_pred)) Classification Report: precision recall f1-score support 0 0.56 0.76 0.65 1312 1 0.79 0.61 0.69 1988 accuracy 0.67 3300 macro avg 0.68 0.68 0.67 3300 weighted avg 0.70 0.67 0.67 3300	66.58	 print(confusion_matrix(y_test, y_pred)) Confusion Matrix: [[997 315] [776 1212]]
Support Vector Machine Classifier	 print(classification_report(y_test, y_pred)) Classification Report: precision recall f1-score support 0 0.54 0.87 0.66 1312 1 0.85 0.51 0.64 1988 accuracy 0.65 3300 macro avg 0.70 0.69 0.65 3300 weighted avg 0.73 0.65 0.65 3300	65.15	 print(confusion_matrix(y_test, y_pred)) Confusion Matrix: [[1139 173] [977 1011]]

5. Model Optimization and Tuning Phase

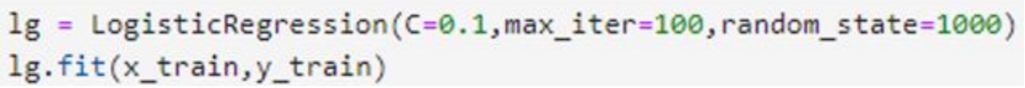
5.1. Hyperparameter Tuning Documentation

- logistic regression:



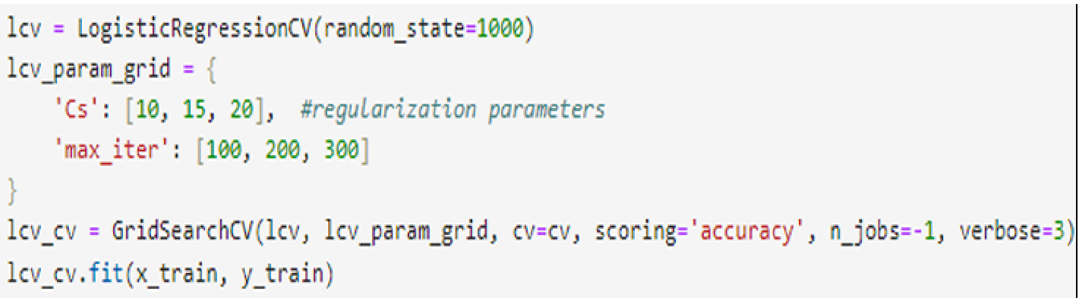
```
lg = LogisticRegression(random_state=1000)
lg_param_grid = {
    'C': [0.01, 0.1, 1, 10, 100], #regularization strength
    'max_iter': [20, 100, 200], #iterations
    'random_state':[200,1000]
}
lg_cv = GridSearchCV(lg, lg_param_grid, cv=cv, scoring='accuracy', n_jobs=-1, verbose=3)
lg_cv.fit(x_train, y_train)
```

Tunied model:



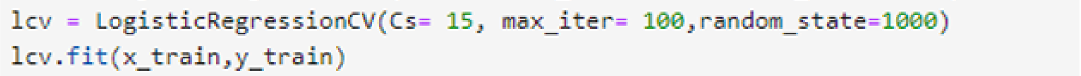
```
lg = LogisticRegression(C=0.1,max_iter=100,random_state=1000)
lg.fit(x_train,y_train)
```

- Logistic Regression with Cross Validation (LogisticRegressionCV)



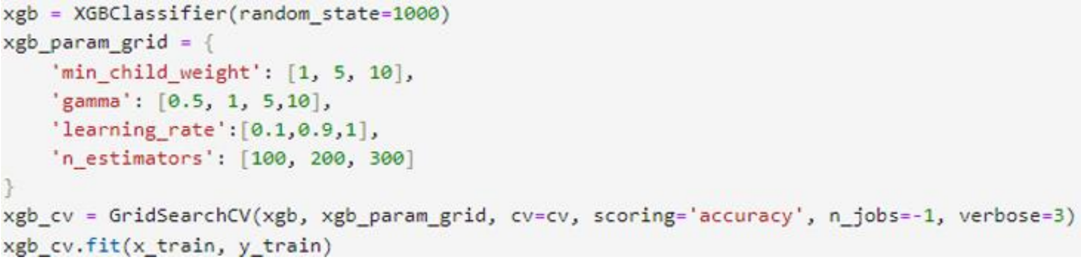
```
lcv = LogisticRegressionCV(random_state=1000)
lcv_param_grid = {
    'Cs': [10, 15, 20],  #regularization parameters
    'max_iter': [100, 200, 300]
}
lcv_cv = GridSearchCV(lcv, lcv_param_grid, cv=cv, scoring='accuracy', n_jobs=-1, verbose=3)
lcv_cv.fit(x_train, y_train)
```

Tunied model:



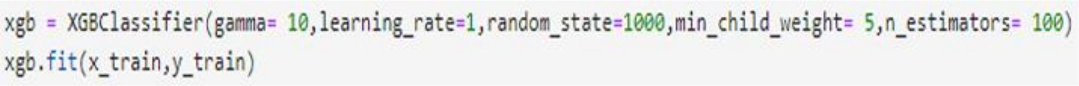
```
lcv = LogisticRegressionCV(Cs= 15, max_iter= 100,random_state=1000)
lcv.fit(x_train,y_train)
```

- XGBoost Classifier



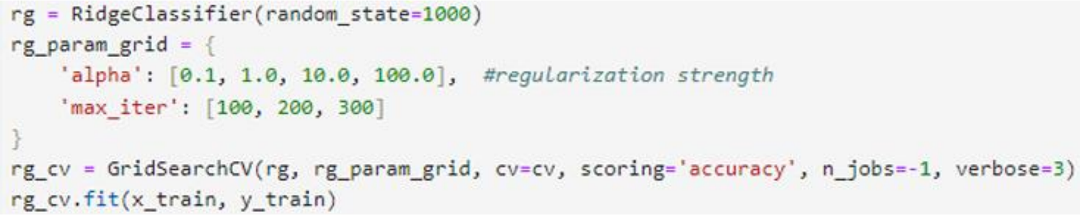
```
xgb = XGBClassifier(random_state=1000)
xgb_param_grid = {
    'min_child_weight': [1, 5, 10],
    'gamma': [0.5, 1, 5,10],
    'learning_rate':[0.1,0.9,1],
    'n_estimators': [100, 200, 300]
}
xgb_cv = GridSearchCV(xgb, xgb_param_grid, cv=cv, scoring='accuracy', n_jobs=-1, verbose=3)
xgb_cv.fit(x_train, y_train)
```

Tunied model:



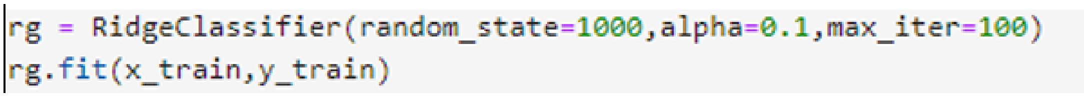
```
xgb = XGBClassifier(gamma= 10,learning_rate=1,random_state=1000,min_child_weight= 5,n_estimators= 100)
xgb.fit(x_train,y_train)
```

- Ridge Classifier



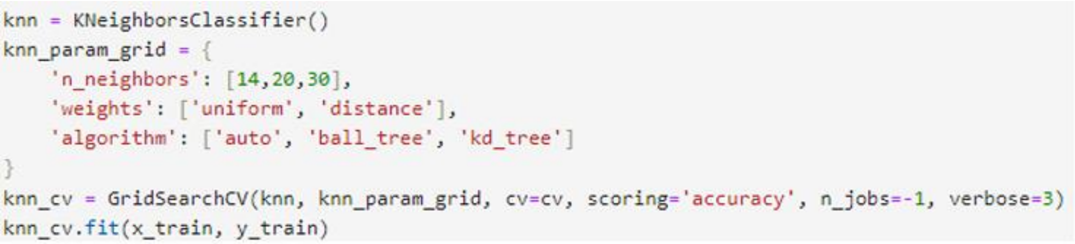
```
rg = RidgeClassifier(random_state=1000)
rg_param_grid = {
    'alpha': [0.1, 1.0, 10.0, 100.0],  #regularization strength
    'max_iter': [100, 200, 300]
}
rg_cv = GridSearchCV(rg, rg_param_grid, cv=cv, scoring='accuracy', n_jobs=-1, verbose=3)
rg_cv.fit(x_train, y_train)
```

Tunied model:



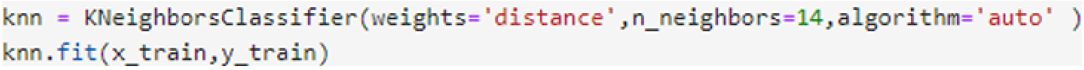
```
rg = RidgeClassifier(random_state=1000,alpha=0.1,max_iter=100)
rg.fit(x_train,y_train)
```

- K-Nearest Neighbours Classifier



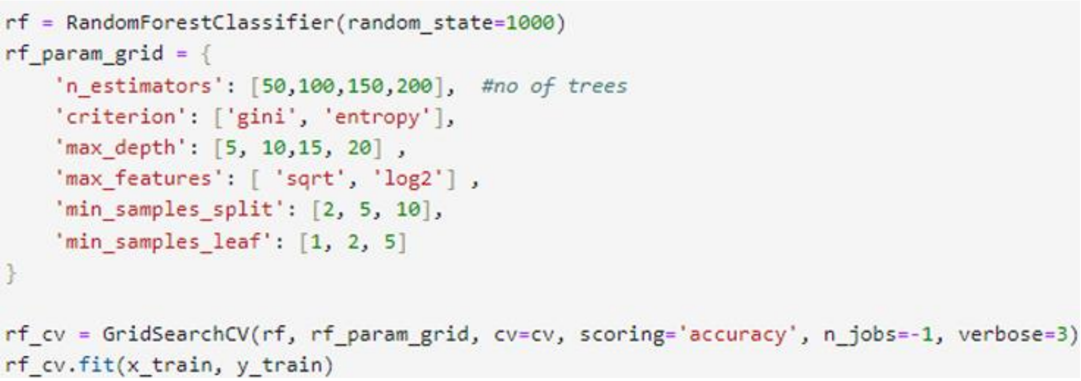
```
knn = KNeighborsClassifier()
knn_param_grid = {
    'n_neighbors': [14,20,30],
    'weights': ['uniform', 'distance'],
    'algorithm': ['auto', 'ball_tree', 'kd_tree']
}
knn_cv = GridSearchCV(knn, knn_param_grid, cv=cv, scoring='accuracy', n_jobs=-1, verbose=3)
knn_cv.fit(x_train, y_train)
```

Tunied model:



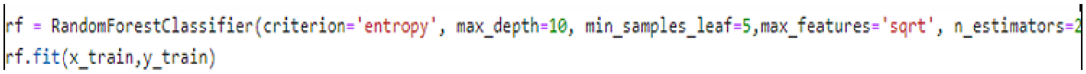
```
knn = KNeighborsClassifier(weights='distance',n_neighbors=14,algorithm='auto' )
knn.fit(x_train,y_train)
```

- Random Forest Classifier



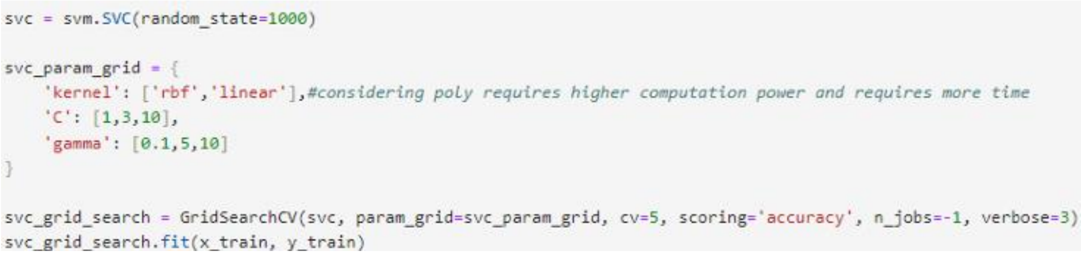
```
rf = RandomForestClassifier(random_state=1000)
rf_param_grid = {
    'n_estimators': [50,100,150,200],  #no of trees
    'criterion': ['gini', 'entropy'],
    'max_depth': [5, 10,15, 20] ,
    'max_features': [ 'sqrt', 'log2'] ,
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 5]
}

rf_cv = GridSearchCV(rf, rf_param_grid, cv=cv, scoring='accuracy', n_jobs=-1, verbose=3)
rf_cv.fit(x_train, y_train)
```

Tunied model:



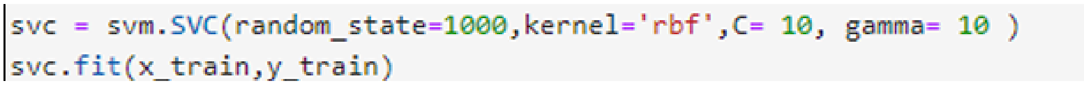
```
rf = RandomForestClassifier(criterion='entropy', max_depth=10, min_samples_leaf=5,max_features='sqrt', n_estimators=2
rf.fit(x_train,y_train)
```

- Support Vector Machine Classifier (SVC).



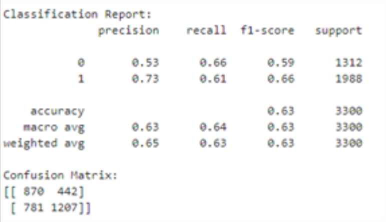
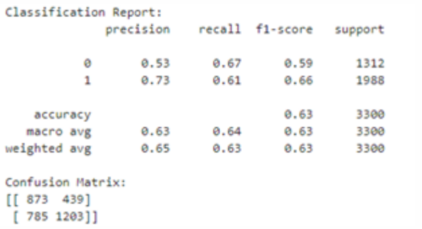
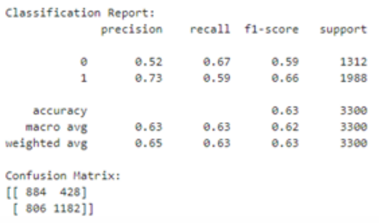
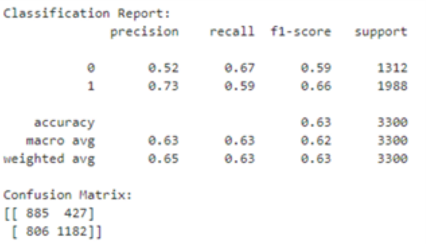
```
svc = svm.SVC(random_state=1000)

svc_param_grid = {
    'kernel': ['rbf','linear'],#considering poly requires higher computation power and requires more time
    'C': [1,3,10],
    'gamma': [0.1,5,10]
}

svc_grid_search = GridSearchCV(svc, param_grid=svc_param_grid, cv=5, scoring='accuracy', n_jobs=-1, verbose=3)
svc_grid_search.fit(x_train, y_train)
```

Tunied model:

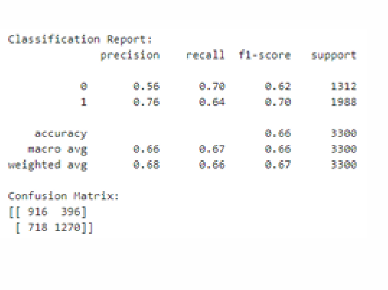


```
svc = svm.SVC(random_state=1000,kernel='rbf',C= 10, gamma= 10 )
svc.fit(x_train,y_train)
```

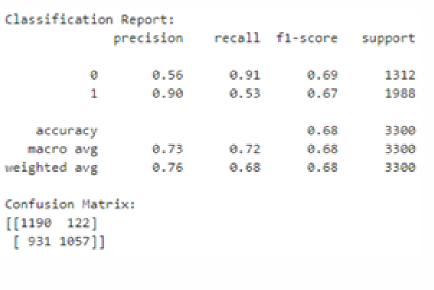
5.2. Performance Metrics Comparison Report

Model	Baseline Metric	Optimized Metric
logistic regression	 Classification Report: precision recall f1-score support 0 0.53 0.66 0.59 1312 1 0.73 0.61 0.66 1988 accuracy 0.63 3300 macro avg 0.63 0.64 0.63 3300 weighted avg 0.65 0.63 0.63 3300 Confusion Matrix: [[870 442] [781 1207]]	 Classification Report: precision recall f1-score support 0 0.53 0.67 0.59 1312 1 0.73 0.61 0.66 1988 accuracy 0.63 3300 macro avg 0.63 0.64 0.63 3300 weighted avg 0.65 0.63 0.63 3300 Confusion Matrix: [[873 439] [785 1203]]
logistic regression CV	 Classification Report: precision recall f1-score support 0 0.52 0.67 0.59 1312 1 0.73 0.59 0.66 1988 accuracy 0.63 3300 macro avg 0.63 0.63 0.62 3300 weighted avg 0.65 0.63 0.63 3300 Confusion Matrix: [[884 428] [806 1182]]	 Classification Report: precision recall f1-score support 0 0.52 0.67 0.59 1312 1 0.73 0.59 0.66 1988 accuracy 0.63 3300 macro avg 0.63 0.63 0.62 3300 weighted avg 0.65 0.63 0.63 3300 Confusion Matrix: [[885 427] [806 1182]]

XGBoost

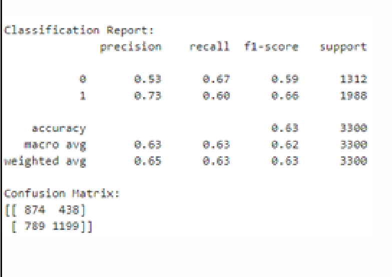


```
Classification Report:
              precision    recall  f1-score   support

           0       0.56      0.70      0.62      1312
           1       0.76      0.64      0.70      1988

    accuracy                           0.66      3300
   macro avg       0.66      0.67      0.66      3300
weighted avg       0.68      0.66      0.67      3300

Confusion Matrix:
[[ 916  396]
 [ 718 1270]]
```

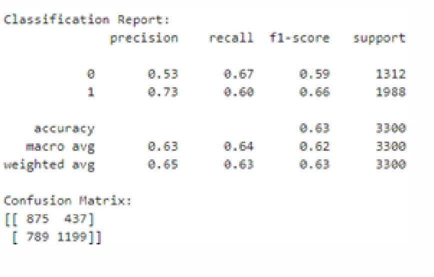


```
Classification Report:
              precision    recall  f1-score   support

           0       0.56      0.91      0.69      1312
           1       0.90      0.53      0.67      1988

    accuracy                           0.68      3300
   macro avg       0.73      0.72      0.68      3300
weighted avg       0.76      0.68      0.68      3300

Confusion Matrix:
[[1190  122]
 [ 931 1057]]
```

Ridge classifier

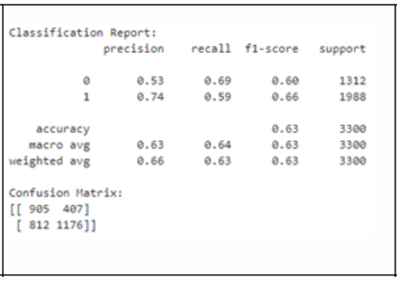


```
Classification Report:
              precision    recall  f1-score   support

           0       0.53      0.67      0.59      1312
           1       0.73      0.60      0.66      1988

    accuracy                           0.63      3300
   macro avg       0.63      0.63      0.62      3300
weighted avg       0.65      0.63      0.63      3300

Confusion Matrix:
[[ 874  438]
 [ 789 1199]]
```

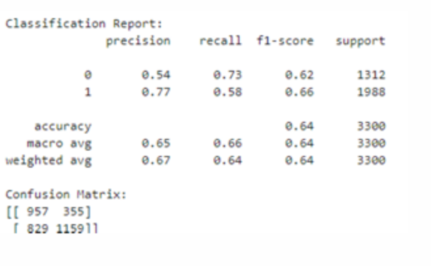


```
Classification Report:
              precision    recall  f1-score   support

           0       0.53      0.67      0.59      1312
           1       0.73      0.60      0.66      1988

    accuracy                           0.63      3300
   macro avg       0.63      0.64      0.62      3300
weighted avg       0.65      0.63      0.63      3300

Confusion Matrix:
[[ 875  437]
 [ 789 1199]]
```

KNN

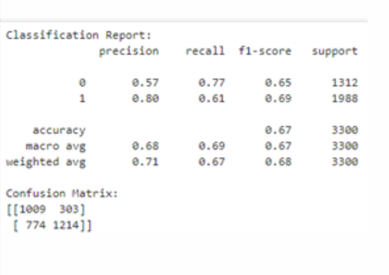


```
Classification Report:
              precision    recall  f1-score   support

           0       0.53      0.69      0.60      1312
           1       0.74      0.59      0.66      1988

    accuracy                           0.63      3300
   macro avg       0.63      0.64      0.63      3300
weighted avg       0.66      0.63      0.63      3300

Confusion Matrix:
[[ 905  407]
 [ 812 1176]]
```

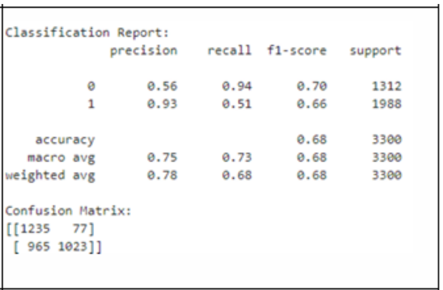


```
Classification Report:
              precision    recall  f1-score   support

           0       0.54      0.73      0.62      1312
           1       0.77      0.58      0.66      1988

    accuracy                           0.64      3300
   macro avg       0.65      0.66      0.64      3300
weighted avg       0.67      0.64      0.64      3300

Confusion Matrix:
[[ 957  355]
 [ 829 1159]]
```

Random Forest



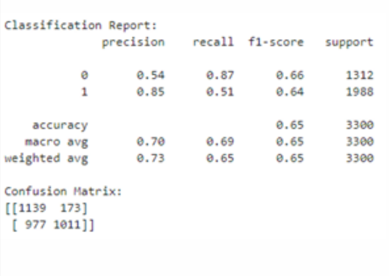
```
Classification Report:
              precision    recall  f1-score   support

           0       0.57      0.77      0.65      1312
           1       0.80      0.61      0.69      1988

    accuracy                           0.67      3300
   macro avg       0.68      0.69      0.67      3300
weighted avg       0.71      0.67      0.68      3300

Confusion Matrix:
[[1009  303]
 [ 774 1214]]
```



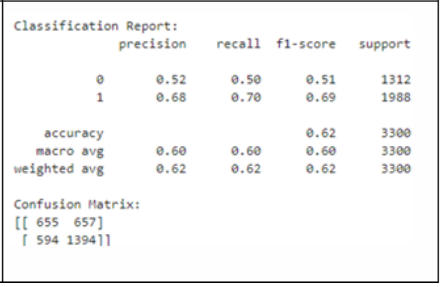
```
Classification Report:
              precision    recall  f1-score   support

           0       0.56      0.94      0.70      1312
           1       0.93      0.51      0.66      1988

    accuracy                           0.68      3300
   macro avg       0.75      0.73      0.68      3300
weighted avg       0.78      0.68      0.68      3300

Confusion Matrix:
[[1235   77]
 [ 965 1023]]
```

Support Vector

Classifier



```
Classification Report:
              precision    recall  f1-score   support

           0       0.54      0.87      0.66      1312
           1       0.85      0.51      0.64      1988

    accuracy                           0.65      3300
   macro avg       0.70      0.69      0.65      3300
weighted avg       0.73      0.65      0.65      3300

Confusion Matrix:
[[1139  173]
 [ 977 1011]]
```



```
Classification Report:
              precision    recall  f1-score   support

           0       0.52      0.50      0.51      1312
           1       0.68      0.70      0.69      1988

    accuracy                           0.62      3300
   macro avg       0.60      0.60      0.60      3300
weighted avg       0.62      0.62      0.62      3300

Confusion Matrix:
[[ 655  657]
 [ 594 1394]]
```

5.3. Final Model Selection Justification

Selected model: Random Forest

Justification/ Reasoning:

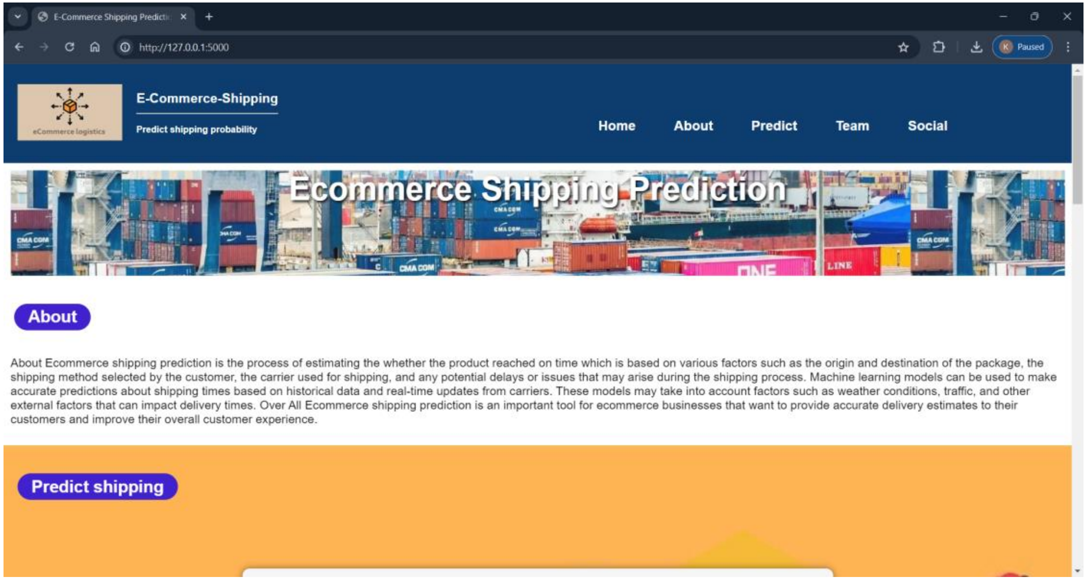
The Random Forest model was chosen as the final optimized model due to its superior performance metrics. It achieved the highest accuracy of 68.42%, demonstrating its effectiveness in making accurate predictions. Additionally, it exhibited a high precision score of 93.00%, indicating its reliability in correctly identifying true positives. Random Forest's ensemble approach helps in minimizing overfitting and improving generalization to new data. These characteristics align well with the project's objectives of enhancing delivery time predictions, making Random Forest the most suitable choice.

6. Results

6.1. Output Screenshots

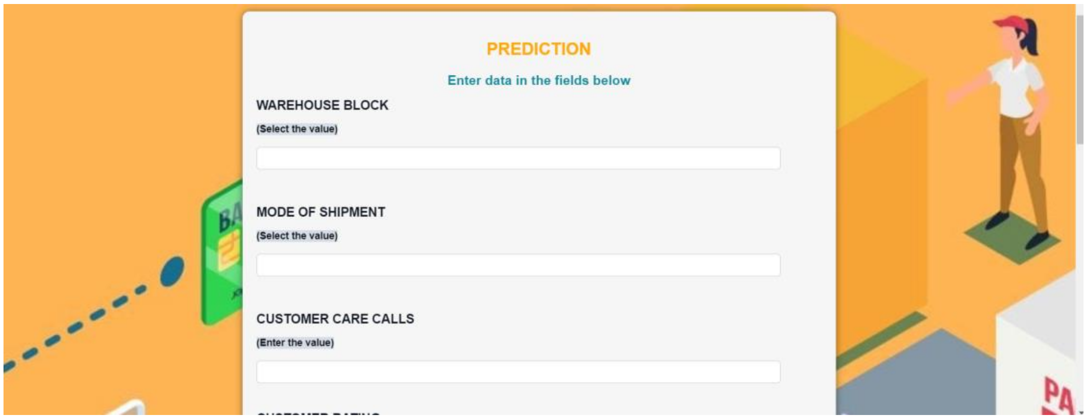
- **Developed webpage:**

➔ **Start of the page:**



About Ecommerce shipping prediction is the process of estimating the whether the product reached on time which is based on various factors such as the origin and destination of the package, the shipping method selected by the customer, the carrier used for shipping, and any potential delays or issues that may arise during the shipping process. Machine learning models can be used to make accurate predictions about shipping times based on historical data and real-time updates from carriers. These models may take into account factors such as weather conditions, traffic, and other external factors that can impact delivery times. Over All Ecommerce shipping prediction is an important tool for ecommerce businesses that want to provide accurate delivery estimates to their customers and improve their overall customer experience.

➔ **In between:**



PREDICTION

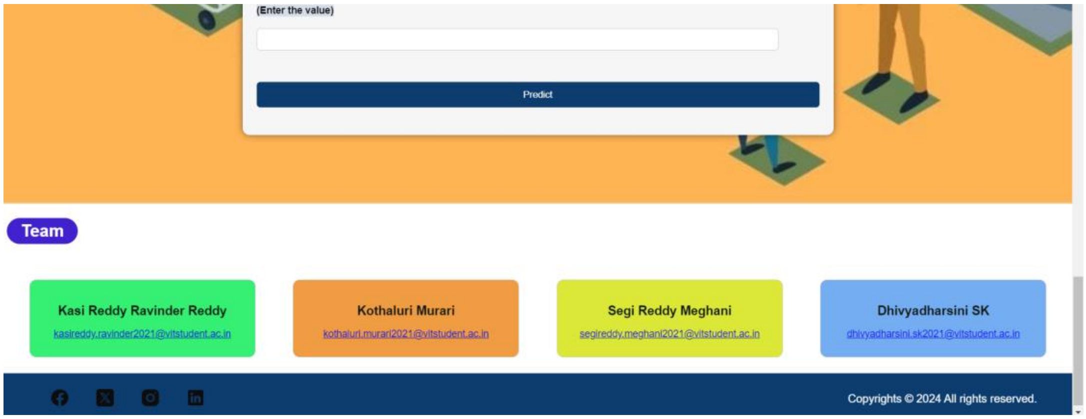
Enter data in the fields below

WAREHOUSE BLOCK
(Select the value)

MODE OF SHIPMENT
(Select the value)

CUSTOMER CARE CALLS
(Enter the value)

➔ **At the end:**



(Enter the value)

Kasi Reddy Ravinder Reddy
kasireddy.ravinder2021@vitstudent.ac.in

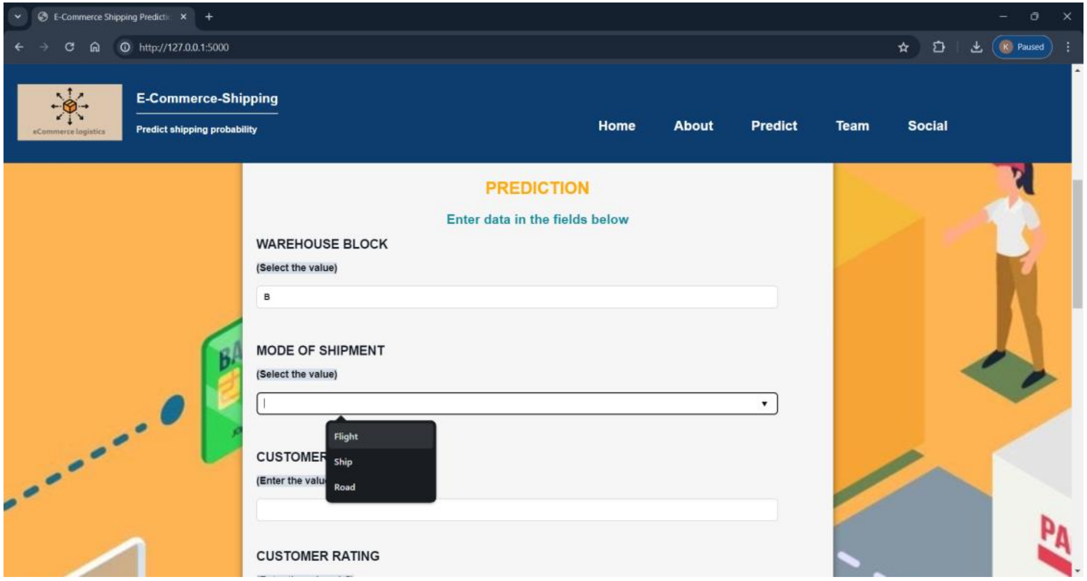
Kothaluri Murari
kothaluri.murari2021@vitstudent.ac.in

Segi Reddy Meghani
segireddy.meghani2021@vitstudent.ac.in

Dhivyadharsini SK
dhivyadharsini.sk2021@vitstudent.ac.in

Copyrights © 2024 All rights reserved.

➔ **For taking inputs:**
For categorical attributes, form provides options to select.



E-Commerce-Shipping
Predict shipping probability

PREDICTION

Enter data in the fields below

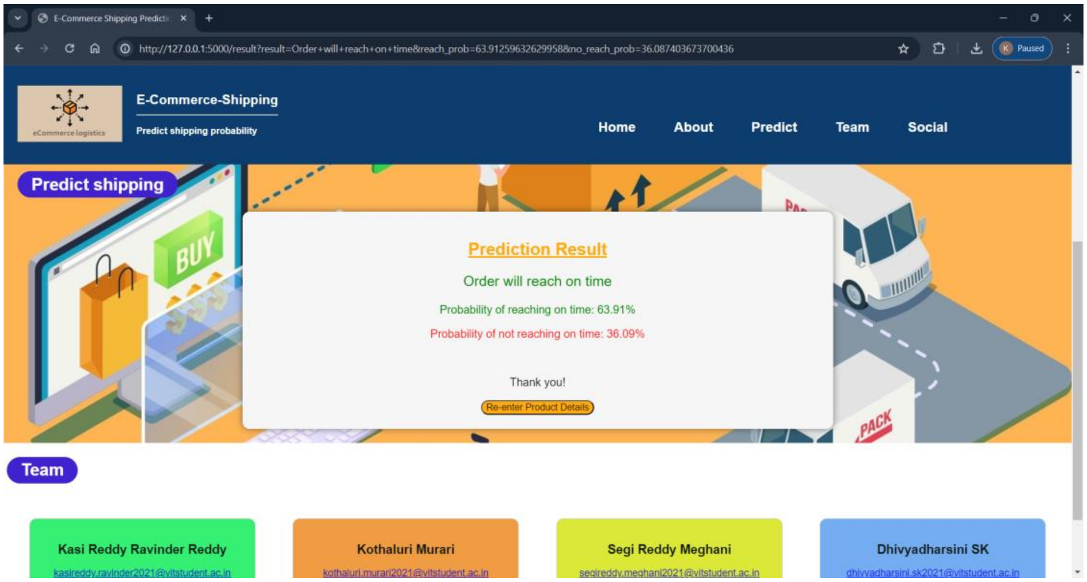
WAREHOUSE BLOCK
(Select the value)
B

MODE OF SHIPMENT
(Select the value)

CUSTOMER
(Enter the valu

CUSTOMER RATING

➔ **After filling the product details, click on the 'Predict' button it will display the results.**



http://127.0.0.1:5000/result?result=Order+will+reach+on+time&reach_prob=63.912596326299558&no_reach_prob=36.087403673700436

Prediction Result

Order will reach on time

Probability of reaching on time: 63.91%

Probability of not reaching on time: 36.09%

Thank you!

Kasi Reddy Ravinder Reddy
kasireddy.ravinder2021@vitstudent.ac.in

Kothaluri Murari
kothaluri.murari2021@vitstudent.ac.in

Segi Reddy Meghani
segireddy.meghani2021@vitstudent.ac.in

Dhivyadharsini SK
dhivyadharsini.sk2021@vitstudent.ac.in

One can click on the 'Re-enter Product Details' button to get the results other products or navigate to form to enter details.

7. Advantages & Disadvantages

Advantages of Developing this Model:

- Improved Delivery Predictions: Enhances accuracy in estimating delivery times, leading to better customer satisfaction and retention.
- Operational Efficiency: Streamlines logistics and reduces costs associated with delayed deliveries and inefficient routing.
- Data-Driven Decision Making: Facilitates informed decisions based on predictive analytics, optimizing resource allocation and inventory management.
- Competitive Edge: Positions the ecommerce platform as reliable and customer-centric, standing out in a competitive market.
- Customer Experience Enhancement: Provides customers with more reliable delivery estimates, improving overall experience and trust.

Disadvantages of Developing this Model:

- User Interface Complexity: Requires a user-friendly interface to communicate predictions effectively, which can be challenging to design and implement.
- Real-Time Data Dependency: Relies on up-to-date data for accurate predictions, necessitating robust data acquisition and integration processes.
- Variable Involvements: Complexities arise from the involvement of multiple variables (e.g., weather conditions, traffic) affecting delivery times, requiring comprehensive data handling and preprocessing.

- Model Maintenance: Regular updates and maintenance of the model are necessary to adapt to changing business conditions and data patterns.
- Ethical Considerations: Potential issues related to privacy concerns and data security when handling sensitive customer information for predictive purposes.

Developing and deploying such a predictive model involves leveraging data effectively while addressing these challenges to achieve sustainable improvements in ecommerce logistics and customer service.

8. Conclusion

The project aimed to enhance ecommerce shipping operations through the development of a predictive model for delivery time estimation. Beginning with project initialization and planning, tasks included defining the business problem, conducting extensive literature surveys, and assessing social and business impacts. Data collection from diverse sources such as Kaggle and UCI provided a robust foundation, encompassing essential variables like shipping mode, customer details, and product specifics.

The project progressed through meticulous data exploration and preprocessing stages, where data quality issues such as missing values and outliers were addressed. Feature selection techniques identified key predictors, crucial for building accurate models. Model selection involved evaluating various classifiers including Logistic Regression, XGBoost, and Random Forest, with the latter demonstrating superior performance in accuracy and precision.

Implementation phases encompassed model training, hyperparameter tuning, and performance testing, ensuring optimal model performance. Challenges such as real-time data integration and user interface design complexities were addressed to deliver actionable insights and reliable delivery predictions. Ethical considerations regarding data privacy and security were prioritized throughout the project lifecycle.

Ultimately, the developed predictive model promises to optimize ecommerce logistics, reduce delivery delays, and enhance customer satisfaction through more accurate delivery time estimates. By leveraging machine learning and data-driven insights, this project contributes to operational efficiency and competitiveness in the ecommerce sector, paving the way for future advancements in predictive analytics and customer-centric service delivery.

9. Future Scope

The project on ecommerce shipping prediction using machine learning holds significant potential for future advancements and applications. Here are key points outlining its future scope:

- **Integration of Real-Time Data Sources**: Enhancing the model's accuracy by incorporating real-time data sources such as traffic updates, weather conditions, and order processing dynamics. This integration will enable more precise predictions aligned with current operational conditions.
- **Advanced Predictive Analytics:** Incorporating advanced analytics techniques like deep learning and ensemble methods to further improve prediction accuracy. These techniques can capture complex patterns and dependencies within data that traditional models may overlook.
- **Optimization of Delivery Routes:** Expanding the model's capabilities to not only predict delivery times but also optimize delivery routes in real-time. This optimization can minimize transportation costs and reduce carbon footprint, aligning with sustainable logistics practices.
- **Enhanced User Interface and Accessibility:** Developing a more intuitive and interactive user interface (UI) for stakeholders, including customers and logistics managers. This UI will provide real-time tracking of shipments and personalized delivery estimates, enhancing user experience.
- **Expansion to Multimodal Logistics:** Extending the predictive model to encompass multimodal logistics, including air, sea, and land transportation. This expansion will cater to diverse ecommerce platforms operating in global markets with varying logistics infrastructures.
- **Predictive Maintenance in Supply Chain:** Implementing predictive maintenance strategies based on insights from shipping prediction models. This proactive approach can prevent equipment failures and optimize inventory management, reducing operational downtime.
- **Enhanced Data Security Measures:** Strengthening data security protocols to safeguard sensitive customer information and transactional data used in predictive analytics. This includes compliance with data protection regulations and industry standards.

In conclusion, the future scope of ecommerce shipping prediction using machine learning is poised for advancements in predictive accuracy, operational efficiency, and customer satisfaction. By leveraging emerging technologies and embracing continuous innovation, this project can lead to transformative changes in ecommerce logistics, setting new benchmarks for service reliability and operational excellence in the digital marketplace.

10. Appendix

10.1. Source Code:

- Link: https://github.com/Ravinder352/Ecommerce-Shipping-Prediction-Using-Machine-Learning/blob/main/5.%20Project%20Executable%20Files/Training/shipment_prediction.ipynb

10.2. GitHub & Project Demo Link

- GitHub link: https://github.com/Ravinder352/Ecommerce-Shipping-Prediction-Using-Machine-Learning
- Project Demo Link: https://drive.google.com/file/d/1VJSU0M1NAskbcJ1jxyt74kvOv1TfYdLU/view?usp=sharing