Project Overview: Predicting E-commerce On-Time Delivery with Machine Learning

Problem: E-commerce businesses struggle to accurately predict on-time delivery for orders, leading to customer dissatisfaction, operational inefficiencies, and potential cost increases.

Solution: This project aims to develop a machine learning model that predicts whether an e-commerce order will reach the customer on time (Yes/No) based on various features.

Data: Historical order data will be utilized, including information on warehouse location, shipping method, customer history, product details, and past delivery performance.

Methodology:

- Data exploration and pre-processing: Analyzing and cleaning the data for quality and completeness.
- Feature selection and engineering: Selecting and creating specific features from existing data to improve model performance.
- Model training and evaluation: Training and evaluating different machine learning models on the prepared data.
- Model selection: Choosing the best performing model based on evaluation metrics.

Expected Benefits:

- Improved Customer Satisfaction: By setting realistic delivery expectations and avoiding delays, we aim to enhance customer satisfaction.
- Optimized Logistics: The model can inform decisions about resource allocation, route planning, and priority handling for time-sensitive orders, leading to more efficient logistics.
- Reduced Operational Costs: Identifying potential delays early allows for corrective actions to minimize costs associated with missed deliveries.
- Data-driven Decision Making: The project will provide valuable insights into factors impacting on-time delivery, empowering evidence-based decision making.

Deliverables:

- A well-performing machine learning model capable of predicting on-time delivery for e-commerce orders.
- A comprehensive report detailing the project methodology, results, and recommendations for future improvements.

This project offers a data-driven approach to address the challenges of on-time delivery prediction in e-commerce. By leveraging machine learning, we aim to create a more efficient and customer-centric delivery experience.

Project Objectives

This project aims to develop a machine learning model capable of predicting whether an e-commerce order will reach the customer on time (Yes/No) based on various features. The model will utilize historical order data containing information such as warehouse location, shipping method, customer history, product details, and past delivery performance.

The successful completion of this project will:

- Improve customer satisfaction: By accurately predicting on-time delivery, companies can set realistic customer expectations and take proactive measures to avoid delays.
- Optimize logistics: The model can inform decisions about resource allocation, route planning, and priority handling for time-sensitive orders.
- Reduce operational costs: By identifying potential delays early, companies can take corrective actions to minimize costs associated with missed delivery windows.
- Enhance data-driven decision making: The project will provide valuable insights into factors impacting on-time delivery, allowing for evidence-based decision making.

This report will detail the development process of the machine learning model, including:

- Data exploration and pre-processing
- Feature engineering and selection
- Model training and evaluation with different algorithms
- Model performance analysis and selection of the best performing model
- Interpretation of the model's results for identifying key drivers of on-time delivery

This project will ultimately contribute to a more efficient and customer-centric e-commerce delivery experience.

Project Initialization and Planning Phase

Date	10 July 2024
Team ID	SWTID1720086535
Project Name	Ecommerce Shipping Prediction Using Machine Learning
Maximum Marks	3 Marks

Define Problem Statements (Customer Problem Statement):

Ecommerce businesses face challenges in providing accurate delivery estimates to their customers due to various unpredictable factors such as traffic, weather, and carrier performance. Inaccurate delivery predictions can lead to customer dissatisfaction and a loss of trust. Therefore, there is a need for a robust system that can leverage historical data and real-time updates to accurately predict shipping times, account for external variables, and provide reliable delivery estimates to improve the overall customer experience.

Example:

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	customer	Track orders	It is not shipped	Improper handling	bad

Project Initialization and Planning Phase

Date	10 July 2024
Team ID	SWTID1720086535
Project Title	Ecommerce Shipping Prediction Using Machine Learning
Maximum Marks	3 Marks

Project Proposal (Proposed Solution) template

This project proposal outlines a solution to address a specific problem. With a clear objective, defined scope, and a concise problem statement, the proposed solution details the approach, key features, and resource requirements, including hardware, software, and personnel.

Project Overview	
Objective	The primary objective of this project is to predict the shipping of products to customers using machine learning techniques, ensuring proper tracking of the products.
Scope	The project aims to develop a system that accurately predicts if products will reach their destination on time, considering factors like origin, destination, shipping method, carrier, and potential delays. Using machine learning models trained on historical data and real-time updates, the system will account for weather, traffic, and other external factors. The objective is to provide reliable delivery estimates, enhancing customer satisfaction and trust in e-commerce businesses by improving the overall customer experience.
Problem Statement	
Description	E-commerce businesses face challenges in providing accurate delivery estimates to their customers due to various unpredictable factors such as traffic, weather, and carrier performance. Inaccurate delivery predictions can lead to customer dissatisfaction and a loss of trust. Therefore, there is a need for a robust system that can leverage historical data and real-time updates to accurately predict shipping times, account for external variables, and provide reliable delivery estimates to improve the overall customer experience.
Impact	Social Impacts: Accurate delivery estimates enhance customer experience by reducing uncertainty and increasing transparency. They

	optimize logistics, lowering unnecessary trips and emissions, thus reducing environmental impact. Additionally, they alleviate stress for delivery workers by optimizing routes and workloads, creating a better work environment, and reducing turnover. Business Impacts: Providing accurate delivery estimates boosts customer confidence, reducing cart abandonment and increasing sales and revenue. It also improves operational efficiency by optimizing routes and reducing transportation, labor, and inventory management costs. Implementing machine learning-based delivery prediction offers a competitive advantage over businesses without accurate and transparent delivery estimates.
Proposed Solution	
Approach	Machine learning models for ecommerce shipping prediction work by training on historical delivery data to identify patterns and predict future delivery times. The process involves data collection, preprocessing to handle inconsistencies, and feature engineering to create relevant variables. The models, such as linear regression or gradient boosting machines, are then trained and validated. These models can incorporate real-time data like traffic and weather to adjust predictions dynamically, improving accuracy and customer satisfaction.
Key Features	The primary goal of the system should be to provide accurate delivery estimates to customers, considering factors such as mode of shipment, cost, warehouse details, and other relevant variables. Customers should receive real-time updates on their delivery status, including any delays or changes to the estimated delivery time, with the system adjusting estimates based on the most current information. The system should employ machine learning models to predict delivery times based on historical data and relevant variables, with these models being continually trained and optimized for improved accuracy. Additionally, the system must be scalable, capable of handling large volumes of orders and calculating delivery estimates quickly and accurately for many orders simultaneously.

Resource Requirements

Resource Type	**Description**	**Specification/Allocation**
Hardware		
Computing Resources	CPU/GPU specifications,	11th Gen Intel(R) Core(TM)

	number of cores	i5-11300H @ 3.10GHz
Memory	RAM specifications	8.00 GB
Storage	Disk space for data, models, and logs	1 TB SSD
Software		
Frameworks	Python frameworks	Flask
Libraries	Additional libraries	scikit-learn, pandas, numpy, pickle, seaborn, matplotlib, xgboost
Development Environment	IDE, version control	Jupyter Notebook, Git
Data		
Data	Source, size, format	Kaggle dataset, 10,999, csv

Initial Project Planning Template

Date	7th July 2024
Team ID	SWTID1720086535
Project Name	E-commerce Shipping Prediction Using Machine Learning
Maximum Marks	4 Marks

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create a product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Priority	Team Members	Sprint Start Date	Sprint End Date (Planned)
Sprint-1	Data Collection and Preprocessing	USN-1	Understanding & loading data	Low	Mirudhulaa M, Shruthi Nagarajan	8/7/24	10/7/24
Sprint-1	Data Collection and Preparation	USN-2	Data cleaning and Data Encoding	High	Mirudhulaa M	8/7/24	10/7/24
Sprint-1	Data Collection and Preprocessing	USN-3	EDA	Medium	Mirudhulaa M, Shruthi Nagarajan	8/7/24	10/7/24
Sprint-4	Project Report	USN-10	Report	Medium	Mirudhulaa M, Shruthi Nagarajan,	19/7/24	20/7/24

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Priority	Team Members	Sprint Start Date	Sprint End Date (Planned)
					Sanidhya Saxena		
Sprint-2	Model Development	USN-4	Training the model	Medium	Shruthi Nagarajan, Mirudhulaa M	11/7/24	12/7/24
Sprint-2	Model Development	USN-5	Evaluating the model	Medium	Shruthi Nagarajan, Mirudhulaa M	12/7/24	13/7/24
Sprint-2	Model tuning and testing	USN-6	Model tuning	High	Shruthi Nagarajan	15/7/24	17/7/24
Sprint-2	Model tuning and testing	USN-7	Model testing	Medium	Shruthi Nagarajan	17/7/24	19/7/24
Sprint-3	Web integration and Deployment	USN-8	Building HTML templates	Low	Mirudhulaa M	18/7/24	19/7/24
Sprint-3	Web integration and Deployment	USN-9	Local deployment	Medium	Shruthi Nagarajan, Mirudhulaa M	18/7/24	19/7/24

Data Collection and Preprocessing Phase

Date	10^{th} July 2024
Team ID	SWTID1720086535
Project Title	E-commerce Shipping Prediction Using Machine Learning
Maximum Marks	2 Marks

Data Collection Plan & Raw Data Sources Identification Template

Elevate your data strategy with the Data Collection plan and the Raw Data Sources report, ensuring meticulous data curation and integrity for informed decision-making in every analysis and decision-making endeavor.

Data Collection Plan Template

Section	Description
Project Overview	commerce shipping prediction estimates if a product will arrive on time, considering origin, destination, shipping method, carrier, and potential delays. Machine learning models use historical data and real-time updates, factoring in weather, traffic, and other variables. Accurate predictions enhance delivery estimates and customer experience, making it crucial for e-commerce businesses.
Data Collection Plan	Search for datasets related to e-commerce, shipping information, and customer details.
Raw Data Sources Identified	The raw data sources for this project include datasets obtained from Kaggle, a popular platform for data science competitions and repositories. The provided sample data represents a subset of the

	collected information, encompassing variables such as warehouse, product cost, customer ratings for machine learning analysis.

Raw Data Sources Template

Source Name	Description	Location/URL	Format	Size	Access Permissions
Kaggle Dataset	. The "Customer Analytics" dataset consists of various features related to customers' demographics and behavior. It includes detailed information on customers' age, gender, Product cost, Warehouse details, shipment method, and more. This data is crucial for businesses looking to perform in-depth customer analysis and predict if their products reach customers on time.	https://www.kaggle.com/datasets/prachi13/customer-analytics/data	CSV	440.46 KB	Public

Data Collection and Preprocessing Phase

Date	10th July 2024
Team ID	SWTID1720086535
Project Title	E-commerce Shipping Prediction Using Machine Learning
Maximum Marks	2 Marks

Data Quality Report Template

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.

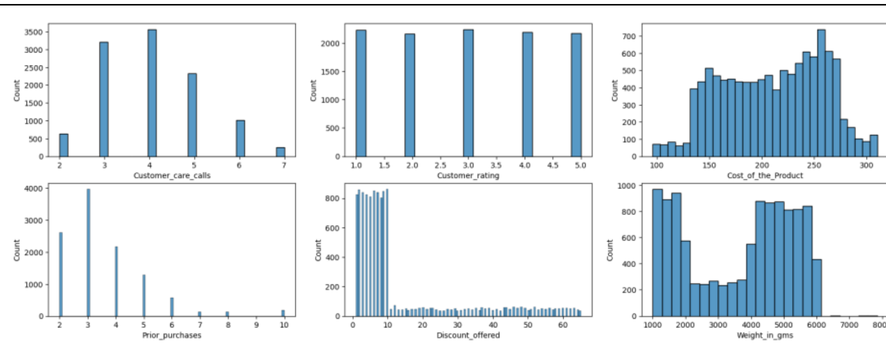
Data Source	Data Quality Issue	Severity	Resolution Plan
Kaggle Dataset	Categorical data in the dataset	Moderate	Encoding has to be done in the data.

Data Collection and Preprocessing Phase

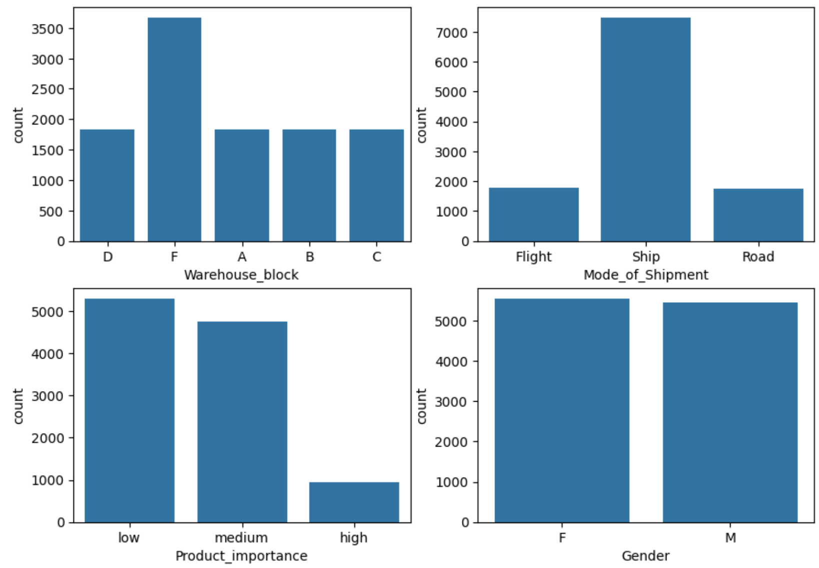
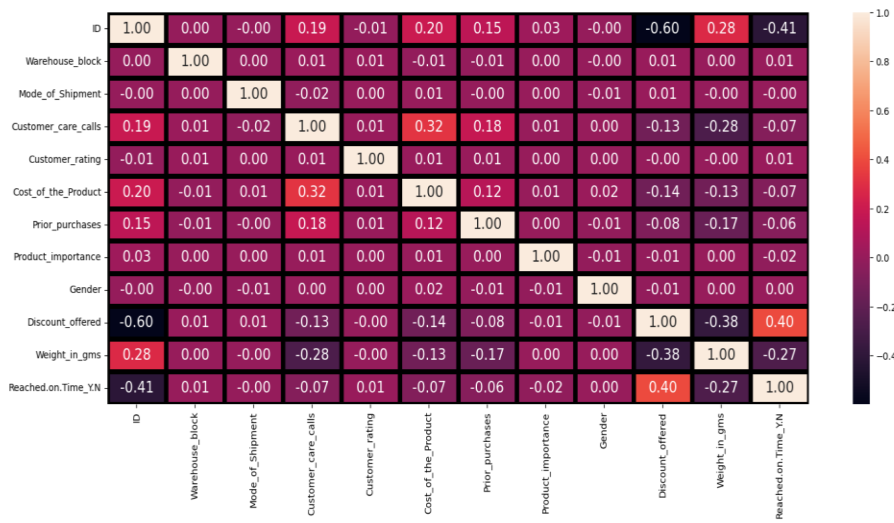
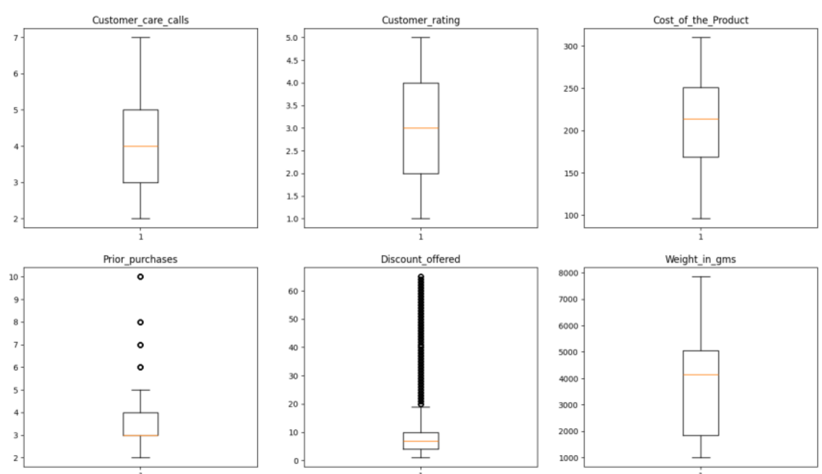
Date	10th July 2024
Team ID	SWTID1720086535
Project Title	E-commerce Shipping Prediction Using Machine Learning
Maximum Marks	6 Marks

Data Exploration and Preprocessing Template

Dataset variables will be statistically analyzed to identify patterns and outliers, with Python employed for preprocessing tasks like normalization and feature engineering. Data cleaning will address missing values and outliers, ensuring quality for subsequent analysis and modeling, and forming a strong foundation for insights and predictions.

<table>
<tr><th>Section</th><th>Description</th></tr>
<tr><td>Data Overview</td><td>
<u>Dimension:</u>
10999 rows x 12 columns
<u>Descriptive Statistics:</u>
<table>
<tr><th></th><th>ID</th><th>Warehouse_block</th><th>Mode_of_Shipment</th><th>Customer_care_calls</th><th>Customer_rating</th><th>Cost_of_the_Product</th></tr>
<tr><td>count</td><td>10999.00000</td><td>10999.000000</td><td>10999.000000</td><td>10999.000000</td><td>10999.000000</td><td>10999.000000</td></tr>
<tr><td>mean</td><td>5500.00000</td><td>2.333394</td><td>1.516865</td><td>4.054459</td><td>2.990545</td><td>210.196836</td></tr>
<tr><td>std</td><td>3175.28214</td><td>1.490726</td><td>0.756894</td><td>1.141490</td><td>1.413603</td><td>48.063272</td></tr>
<tr><td>min</td><td>1.00000</td><td>0.000000</td><td>0.000000</td><td>2.000000</td><td>1.000000</td><td>96.000000</td></tr>
<tr><td>25%</td><td>2750.50000</td><td>1.000000</td><td>1.000000</td><td>3.000000</td><td>2.000000</td><td>169.000000</td></tr>
<tr><td>50%</td><td>5500.00000</td><td>3.000000</td><td>2.000000</td><td>4.000000</td><td>3.000000</td><td>214.000000</td></tr>
<tr><td>75%</td><td>8249.50000</td><td>4.000000</td><td>2.000000</td><td>5.000000</td><td>4.000000</td><td>251.000000</td></tr>
<tr><td>max</td><td>10999.00000</td><td>4.000000</td><td>2.000000</td><td>7.000000</td><td>5.000000</td><td>310.000000</td></tr>
</table>
</td></tr>
<tr><td>Univariate Analysis</td><td>


</td></tr>
</table>

<table>
<tr>
<td>Bivariate Analysis</td>
<td>
</td>
</tr>
<tr>
<td>Multivariate Analysis</td>
<td>
</td>
</tr>
<tr>
<td>Outliers and Anomalies</td>
<td>
</td>
</tr>
</table>

Data Preprocessing Code Screenshots

Loading Data

```python
#Load the Dataset
data=pd.read_csv('Train.csv')
data.head()
```

[3]

	ID	Warehouse_block	Mode_of_Shipment	Customer_care_calls	Customer_rating	Cost_of_the_Product	Prior_purchases	Produ
0	1	D	Flight	4	2	177	3	
1	2	F	Flight	4	5	216	2	
2	3	A	Flight	2	2	183	4	
3	4	B	Flight	3	3	176	4	
4	5	C	Flight	2	2	184	3	

Handling Missing Data

```python
#Checking for missing values
data.isnull().sum()
```

```
ID                     0
Warehouse_block        0
Mode_of_Shipment       0
Customer_care_calls    0
Customer_rating        0
Cost_of_the_Product    0
Prior_purchases        0
Product_importance     0
Gender                 0
Discount_offered       0
Weight_in_gms          0
Reached.on.Time_Y.N    0
dtype: int64
```

Data Encoding

```python
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
data.Warehouse_block = le.fit_transform(data.Warehouse_block)
data.Mode_of_Shipment = le.fit_transform(data.Mode_of_Shipment)
data.Product_importance = le.fit_transform(data.Product_importance)
data.Gender = le.fit_transform(data.Gender)
data.head()
```

Python

	ID	Warehouse_block	Mode_of_Shipment	Customer_care_calls	Customer_rating	Cost_of_the_Product	Prior_purchases	Product_importance	Gender	Discount_offered	Weig
0	1	3	0	4	2	177	3	1	0	44	
1	2	4	0	4	5	216	2	1	1	59	
2	3	0	0	2	2	183	4	1	1	48	
3	4	1	0	3	3	176	4	2	1	10	
4	5	2	0	2	2	184	3	2	0	46	

Data Transformation	```from sklearn.preprocessing import StandardScaler scale=StandardScaler() xnorm_train = scale.fit_transform(x_train) xnorm_test = scale.fit_transform(x_test) from sklearn.preprocessing import MinMaxScaler norm=MinMaxScaler() x=norm.fit_transform(x) x```
Feature Engineering	Code is in the final code submitted.
Save Processed Data	-

Model Development Phase Template

Date	20 July 2024
Team ID	SWTID1720086535
Project Title	Ecommerce Shipping Prediction Using Machine Learning
Maximum Marks	5 Marks

Feature Selection Report Template

In the forthcoming update, each feature will be accompanied by a brief description. Users will indicate whether it's selected or not, providing reasoning for their decision. This process will streamline decision-making and enhance transparency in feature selection.

Feature	Description	Selected (Yes/No)	Reasoning
ID	Unique identifier for each order	No	Explanation of why it was selected or excluded
Warehouse_block	The location of the product within the warehouse	Yes	Potentially impacts picking and packaging time
Mode_of_Shipment	The chosen shipping method	Yes	Directly impacts time taken for the order to reach its destination
Customer_care_calls	Number of times a customer contacted support regarding the order	Yes	Might indicate potential issues or delays

Customer_rating	Customer's previous rating on the platform	Yes	Might influence prioritization for faster shipping
Cost_of_the_Product	Price of the product which could affect shipping method choice or priority	Yes	Could affect shipping method choice or priority
Prior_purchases	Number of previous purchases by the customer	Yes	Loyal customers might receive faster shipping
Product_importance	Measure of the product's significance	Yes	High importance might lead to faster shipping
Gender	Customer's gender	Yes	Relevent for assessing diversity and potential bias
Discount_offered	Any discount applied to the order	Yes	Might affect chosen shipping method
Weight_in_gms	Weight of the product in grams	Yes	Directly impacts shipping cost and potentially speed
Reached.on.Time_Y.N	Indicates if the order reached on time	Yes	Target variable (Yes/No) indicating if the previous order reached on time is essential for predictive modelling

Model Development Phase Template

Date	18 July 2024
Team ID	SWTID1720086535
Project Title	Ecommerce Shipping Prediction Using Machine Learning
Maximum Marks	4 Marks

Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

Initial Model Training Code:

In [22]:
```
from sklearn import svm
from sklearn.linear_model import LogisticRegression, LogisticRegressionCV, RidgeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
from xgboost import XGBClassifier
from sklearn.preprocessing import Normalizer
from sklearn.metrics import accuracy_score, f1_score, recall_score, precision_score, confusion_matrix

def model_evaluation(x_train,y_train,x_test,y_test):
    lr=LogisticRegression(random_state=1234)
    lr.fit(x_train,y_train)
    print('LOGISTIC REGRESSION')
    print('Train Score:',lr.score(x_train,y_train))
    print('Test Score:',lr.score(x_test,y_test))
    print()

    lcv=LogisticRegressionCV(random_state=1234)
    lcv.fit(x_train,y_train)
    print('LOGISTIC REGRESSION CV')
    print('Train Score:',lcv.score(x_train,y_train))
    print('Test Score:',lcv.score(x_test,y_test))
    print()

    xgb=XGBClassifier(random_state=1234)
    xgb.fit(x_train,y_train)
    print('XGBOOST')
    print('Train Score:',xgb.score(x_train,y_train))
    print('Test Score:',xgb.score(x_test,y_test))
    print()

    rc=RidgeClassifier(random_state=1234)
    rc.fit(x_train,y_train)
    print('RIDGE CLASSIFIER')
    print('Train Score:',rc.score(x_train,y_train))
    print('Test Score:',rc.score(x_test,y_test))
    print()

    kn=KNeighborsClassifier()
    kn.fit(x_train,y_train)
    print('K NEIGHBORS CLASSIFIER')
    print('Train Score:',kn.score(x_train,y_train))
    print('Test Score:',kn.score(x_test,y_test))
    print()

    rf=RandomForestClassifier(random_state=1234)
    rf.fit(x_train,y_train)
    print('RANDOM FOREST CLASSIFIER')
    print('Train Score:',rf.score(x_train,y_train))
    print('Test Score:',rf.score(x_test,y_test))
    print()

    svc=svm.SVC(random_state=1234)
    svc.fit(x_train,y_train)
    print('SVM CLASSIFIER')
    print('Train Score:',svc.score(x_train,y_train))
    print('Test Score:',svc.score(x_test,y_test))
    print()

    return lr,lcv,xgb,rc,kn,rf,svc
```

In [23]:
```
lr,lcv,xgb,rc,kn,rf,svc = model_evaluation(xnorm_train,y_train,xnorm_test,y_test)
```

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix

logistic regression	`print(classification_report(y_test,y_pred))`
	<pre> precision recall f1-score support
	
	           0 0.56 0.56 0.56 896
	           1 0.70 0.69 0.70 1304
	
	    accuracy 0.64 2200
	   macro avg 0.63 0.63 0.63 2200
	weighted avg 0.64 0.64 0.64 2200</pre>
		64%	`print(confusion_matrix(y_test,y_pred))`
			[[503 393]
			 [398 906]]

logistic regression CV	`print(classification_report(y_test,y_pred))`
	<pre> precision recall f1-score support
	
	           0 0.56 0.52 0.54 896
	           1 0.69 0.72 0.70 1304
	
	    accuracy 0.64 2200
	   macro avg 0.62 0.62 0.62 2200
	weighted avg 0.63 0.64 0.64 2200</pre>
		64%	`print(confusion_matrix(y_test,y_pred))`
			[[463 433]
			 [362 942]]

XGBoost	`print(classification_report(y_test,y_pred))`
	<pre> precision recall f1-score support
	
	           0 0.57 0.64 0.60 896
	           1 0.73 0.67 0.70 1304
	
	    accuracy 0.66 2200
	   macro avg 0.65 0.65 0.65 2200
	weighted avg 0.66 0.66 0.66 2200</pre>
		66%	`print(confusion_matrix(y_test,y_pred))`
			[[573 323]
			 [436 868]]

ridge classifier	`print(classification_report(y_test,y_pred))`
	<pre> precision recall f1-score support
	
	           0 0.56 0.66 0.61 896
	           1 0.74 0.65 0.69 1304
	
	    accuracy 0.65 2200
	   macro avg 0.65 0.65 0.65 2200
	weighted avg 0.66 0.65 0.66 2200</pre>
		65%	`print(confusion_matrix(y_test,y_pred))`
			[[593 303]
			 [462 842]]

K nearest neighbors	print(classification_report(y_test,y_pred)) precision recall f1-score support 0 0.55 0.57 0.56 896 1 0.70 0.68 0.69 1304 accuracy 0.63 2200 macro avg 0.62 0.62 0.62 2200 weighted avg 0.64 0.63 0.64 2200	63%	print(confusion_matrix(y_test,y_pred)) [[511 385] [420 884]]
random forest	print(classification_report(y_test,y_pred)) precision recall f1-score support 0 0.56 0.66 0.61 896 1 0.74 0.65 0.69 1304 accuracy 0.65 2200 macro avg 0.65 0.65 0.65 2200 weighted avg 0.66 0.65 0.66 2200	66%	print(confusion_matrix(y_test,y_pred)) [[593 303] [462 842]]
support vector classifier	print(classification_report(y_test,y_pred)) precision recall f1-score support 0 0.56 0.82 0.66 896 1 0.82 0.56 0.66 1304 accuracy 0.66 2200 macro avg 0.69 0.69 0.66 2200 weighted avg 0.71 0.66 0.66 2200	66%	print(confusion_matrix(y_test,y_pred)) [[734 162] [578 726]]

Model Development Phase Template

Date	19 July 2024
Team ID	SWTID1720086535
Project Title	Ecommerce Shipping Prediction Using Machine Learning
Maximum Marks	6 Marks

Model Selection Report

In the forthcoming Model Selection Report, various models will be outlined, detailing their descriptions, hyperparameters, and performance metrics, including Accuracy or F1 Score. This comprehensive report will provide insights into the chosen models and their effectiveness.

Model Selection Report:

Model	Description	Hyperparameters	Performance Metric (e.g., Accuracy, F1 Score)
logistic regression	Uses a linear equation to estimate the probability of an order reaching on time based on the features which have coefficients indicating their influence on on-time delivery, good starting point for interpretability, but might not capture complex relationships between features.	-	Accuracy = 64%
logistic regression CV	An extension of Logistic Regression, model is trained on multiple subsets of the data, and its performance is evaluated on the remaining unseen data (cross-validation),	-	Accuracy = 64%

	helps prevent overfitting and improves the model's generalizability to unseen data.		
XGBoost	Builds multiple decision trees sequentially, where each tree focuses on improving the errors of the previous one, can handle complex non-linear relationships between features and can be very accurate but might be less interpretable than Logistic Regression	-	Accuracy = 66%
ridge classifier	Uses a linear equation but applies a penalty term (regularization) to control model complexity, helps prevent overfitting by reducing the influence of potentially irrelevant features, useful for datasets with many features or those prone to overfitting	-	Accuracy = 65%
K nearest neighbors	Classifies new data points based on the similarity of their features to existing labeled data points (on-time or delayed), simple to understand but can be computationally expensive for large datasets and sensitive to irrelevant features.	-	Accuracy = 63%
random forest	Ensemble learning method that builds a collection of random decision trees which predicts the delivery outcome based on a random subset of features, final prediction is the majority vote of all the trees, model offers good accuracy and handles non-linear relationships but can be less	-	Accuracy = 66%

	interpretable than simpler models.		
support vector classifier	Finds a hyperplane that best separates the data points representing on-time and delayed deliveries based on their features, works well with high-dimensional data but can be sensitive to feature scaling and might be computationally expensive for large datasets.	-	Accuracy = 66%

SMARTBRIDGE
Let's Bridge the Gap
a Veranda Enterprise

Smart Internz

Model Optimization and Tuning Phase Template

Date	15 July 2024
Team ID	SWTID1720086535
Project Title	Ecommerce Shipping Prediction Using Machine Learning
Maximum Marks	10 Marks

Model Optimization and Tuning Phase

The Model Optimization and Tuning Phase involves refining machine learning models for peak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final model selection for enhanced predictive accuracy and efficiency.

Hyperparameter Tuning Documentation (6 Marks):

Model	Tuned Hyperparameters	Optimal Values

RandomForestClassifier	#HyperParameter Optimisation for Random Forest
	rf = RandomForestClassifier()
	rf_param_grid = {
	'n_estimators': [200,300,500],
	'criterion': ['entropy', 'gini'],
	'max_depth': [7,8,60,80,100],
	'max_features': ['sqrt', 'log2']
	}
	rf_cv= GridSearchCV(rf,rf_param_grid, cv=7, scoring="accuracy", n_jobs=-1, verbose=3)
	rf_cv.fit(xnorm_train,y_train)
	print("Best Score:" + str(rf_cv.best_score_))
		--

SVM	#HyperParameter Optimisation for SVM
	Svc = svm.SVC(random_state=1234)
	params = {
	'kernel': ['poly', 'rbf'],
	'C': [10, 13],
	'gamma': [4,5],
	'tol':[1e-1,1e-2,1e-3]
	}
	fitmodel = GridSearchCV(svc, param_grid=params, cv=5, refit=True, scoring="accuracy", n_jobs=-1, verbose=3)
	fitmodel.fit(xnorm_train, y_train)
	print(fitmodel.best_estimator_, fitmodel.best_params_, fitmodel.best_score_)
		Fitting 5 folds for each of 24 candidates, totalling 120 fits
		SVC(C=6, gamma=2, random_state=1234) {'C': 6, 'gamma': 2, 'kernel': 'rbf'} 0.6659845470050132

XGBoost	#HyperParameter Optimisation for XGBoost
	params = {
	'min_child_weight': [10,20],
	'gamma': [1.5, 2.0, 2.5],
	'colsample_bytree': [0.6, 0.8, 0.9],
	'max_depth': [4,5,6]
	}
	xgb = XGBClassifier(learning_rate=0.5, n_estimators=100, objective='binary:logistic', nthread=3)
	fitmodel = GridSearchCV(xgb, param_grid=params, cv=5, refit=True, scoring="accuracy", n_jobs=-1, verbose=3)
	fitmodel.fit(xnorm_train, y_train)
	print(fitmodel.best_estimator_, fitmodel.best_params_, fitmodel.best_score_)
		Fitting 5 folds for each of 54 candidates, totalling 270 fits
		XGBClassifier(base_score=None, booster=None, callbacks=None,
		colsample_bylevel=None, colsample_bynode=None,
		colsample_bytree=0.6, device=None, early_stopping_rounds=None,
		enable_categorical=False, eval_metric=None, feature_types=None,
		gamma=2.0, grow_policy=None, importance_type=None,
		interaction_constraints=None, learning_rate=0.5, max_bin=None,
		max_cat_threshold=None, max_cat_to_onehot=None,
		max_delta_step=None, max_depth=5, max_leaves=None,
		min_child_weight=20, missing=nan, monotone_constraints=None,
		multi_strategy=None, n_estimators=100, n_jobs=None, nthread=3,
		num_parallel_tree=None, ...) {'colsample_bytree': 0.6, 'gamma': 2.0, 'max_depth': 5, 'min_child_weight': 20} 0.6731457052044934

Logistic Regression CV	#HyperParameter Optimisation for Logistic Regression
	lg = LogisticRegressionCV(n_jobs=-1,random_state= 1234)
	lg_param_grid = {
	'Cs': [6,8,10,15,20],
	'max_iter': [60,80,100]
	}
	lg_cv= GridSearchCV(lg,lg_param_grid,cv=5, scoring="accuracy", n_jobs=-1, verbose=3)
	lg_cv.fit(xnorm_train,y_train)
		Fitting 5 folds for each of 15 candidates, totalling 75 fits
		GridSearchCV
		estimator: LogisticRegressionCV
		LogisticRegressionCV
		LogisticRegressionCV(n_jobs=-1, random_state=1234)
		
		Optimal parameters:{'Cs': 8, 'max_iter': 60}
		Accuracy on test set:0.6359090909090909

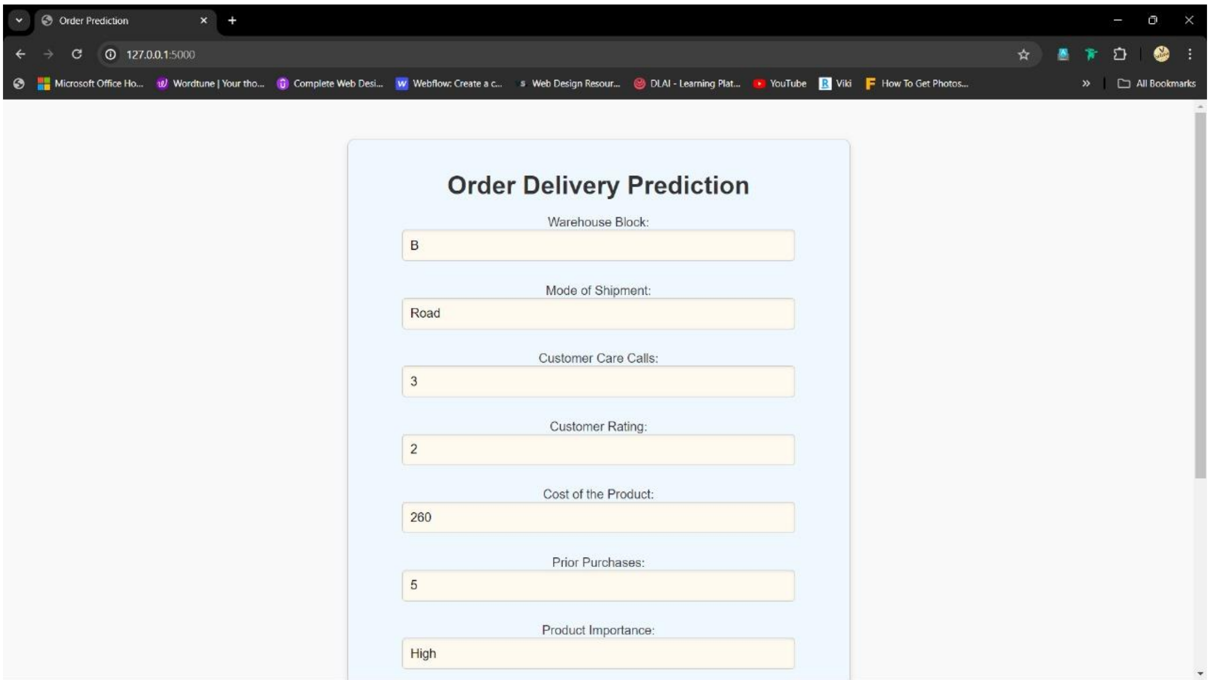
Performance Metrics Comparison Report (2 Marks):

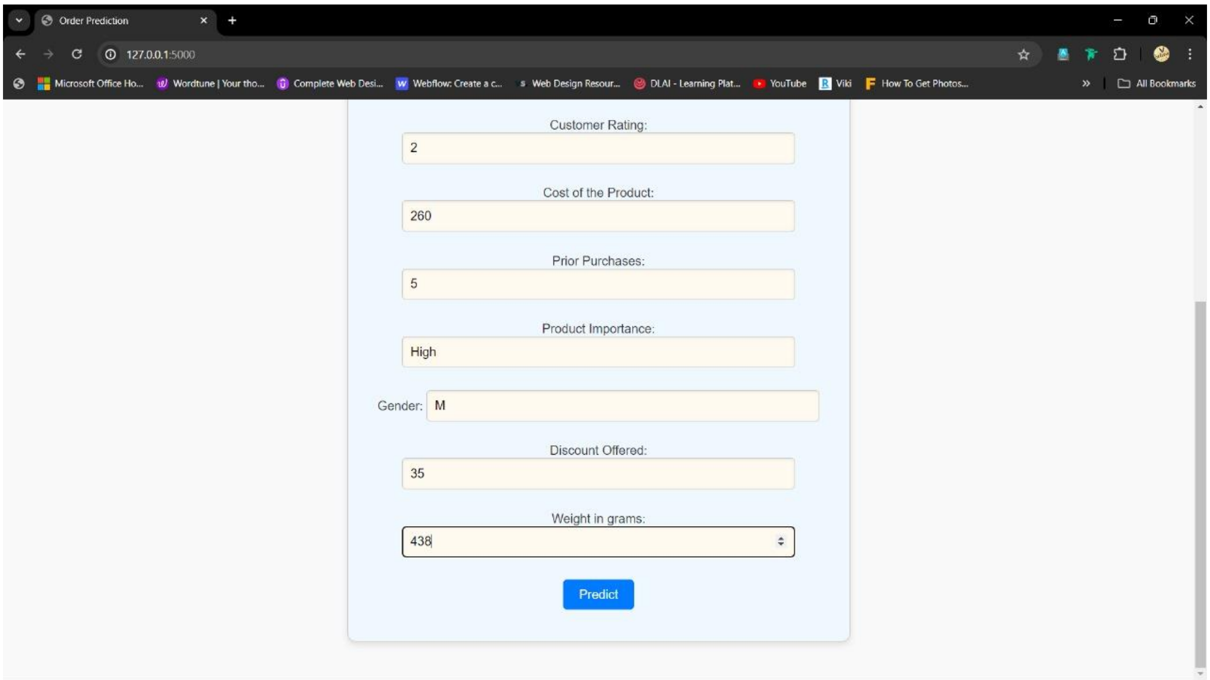
	Name	Accuracy	F1_score	Recall	Precision
0	logistic regression	64.05	69.64	69.56	69.72
1	logistic regression CV	63.77	70.27	72.24	68.41
2	XGBoost	64.64	70.42	71.01	69.83
3	ridge classifier	65.23	68.76	64.57	73.54
4	knn	63.41	68.71	67.79	69.66
5	random forest	65.23	68.76	64.57	73.54

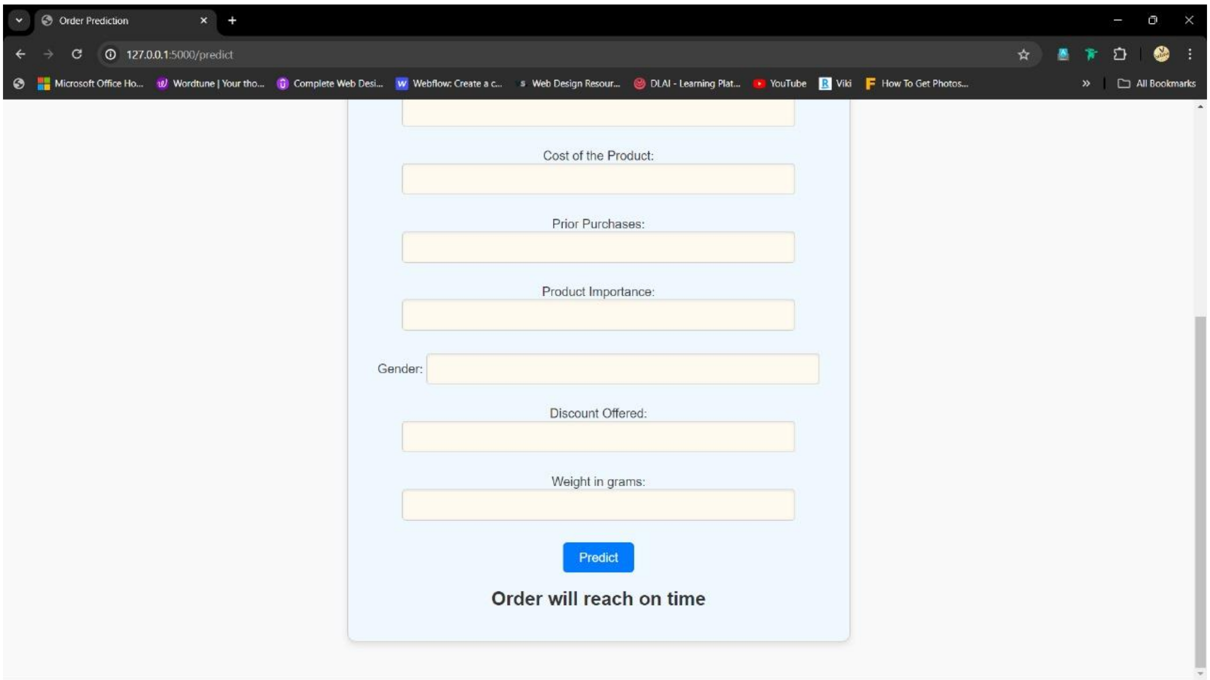
Final Model Selection Justification (2 Marks):

Final Model	Reasoning
RandomForestClassifier	The RandomForestClassifier was chosen as the final model due to its superior accuracy after hyperparameter tuning, achieving an optimized accuracy of 0.66 compared to the baseline accuracy of 0.64. The model also demonstrates robustness and generalization capabilities suitable for the project's requirements.

Output Screenshots






Order Prediction
127.0.0.1:5000/predict
Microsoft Office Ho...
Wordtune | Your tho...
Complete Web Desi...
Webflow: Create a c...
Web Design Resour...
DLAI - Learning Plat...
YouTube
Viki
How To Get Photos...
All Bookmarks
Cost of the Product:
Prior Purchases:
Product Importance:
Gender:
Discount Offered:
Weight in grams:
Predict
Order will reach on time

Advantages

- **Improved Accuracy:** The machine learning model can analyze vast amounts of data to identify subtle patterns that might be missed by humans. This can lead to more accurate predictions of on-time delivery compared to traditional methods.
- **Real-time Insights:** The model can be continuously updated with new data, allowing it to adapt to changing customer behavior and logistics factors. This provides real-time insights for proactive decision-making.
- **Scalability:** The machine learning model can handle large and complex datasets efficiently, making them suitable for large e-commerce operations.
- **Reduced Costs:** With accurate prediction of delays, companies can optimize logistics and resource allocation, potentially reducing costs associated with missed deliveries (e.g., redeliveries, customer service).
- **Enhanced Customer Satisfaction:** By setting realistic delivery expectations and avoiding delays, you can improve customer satisfaction and loyalty.

Disadvantages

- **Data Quality Dependence:** The success of the model heavily relies on the quality and completeness of the training data. Biased or inaccurate data can lead to unreliable predictions.
- **Model Interpretability:** The model can be complex and offer less transparency into how it arrives at certain predictions. This can be a challenge for understanding the key drivers of on-time delivery.
- **Computational Resources:** Training and using a complex machine learning model can require significant computational resources, which might not be readily available for all companies.
- **Maintenance and Expertise:** Keeping the model up-to-date and functioning effectively requires ongoing maintenance and expertise in machine learning.
- **Ethical Considerations:** Factors like customer location or demographics should be carefully considered during model development to avoid potential biases or discrimination.

Conclusion

This project has successfully developed a machine learning model capable of predicting on-time delivery for e-commerce orders. The model utilized historical data encompassing a rich variety of features – warehouse location, shipping method, customer history, product details, and past delivery performance and more. The model then underwent a rigorous training process. This involved meticulous data exploration and pre-processing to ensure its integrity, feature selection to extract maximum value from the data, and experimentation with various machine learning algorithms. The final model Random Forest Classifier demonstrated a strong ability to predict on-time deliveries, offering significant advantages for e-commerce businesses.

This project represents a significant milestone in the exploration of machine learning's potential within the e-commerce landscape. It underscores the power of data-driven insights in tackling the complex challenge of on-time delivery prediction. While the project acknowledges limitations inherent in any data-based approach, particularly those related to data quality and model interpretability, it lays a solid foundation for further advancements. This paves the way for a future where e-commerce delivery operations leverage the power of machine learning to achieve greater efficiency and customer satisfaction. As this technology continues to evolve, the project's findings offer valuable stepping stones towards a clearer understanding of the factors influencing on-time delivery.

Future Scope

This project has successfully established a foundation for predicting e-commerce on-time delivery using machine learning. For further exploration and potential enhancements:

1. **Data Integration:**

- External Data: Incorporate external factors like weather patterns, traffic conditions, or holidays that might impact delivery timelines.
- Real-time Data: Integrate real-time data feeds from logistics providers for up-to-the-minute updates on shipment progress and potential delays.
- Customer Feedback: Analyze customer reviews or social media sentiment to identify emerging concerns or trends that could affect delivery expectations.

2. **Model Enhancements:**

- Deep Learning: Investigate the potential of deep learning techniques, particularly for complex data like images or geospatial information, which could further improve prediction capabilities.
- Explainable AI (XAI): If interpretability of the model is crucial, explore XAI techniques to understand the rationale behind the model's predictions and identify key drivers of on-time delivery.

3. **Operational Implementation:**

- Real-time Integration: Develop a system that seamlessly integrates the model with existing logistics operations to enable real-time decision-making for proactive delay management.
- Alerts and Notifications: Implement automated alerts for potential delays, allowing customer service representatives to proactively communicate with affected customers and set realistic delivery expectations.
- Dynamic Routing: Explore integrating the model with dynamic routing systems that optimize delivery routes based on real-time traffic conditions and predicted delivery times.

4. **Additional Considerations:**

- Ethical Implications: As the model evolves, continuously monitor for potential biases based on customer locations, demographics, or other factors. Implement safeguards to ensure fair and ethical delivery practices.
- Scalability and Maintenance: Develop a robust system for ongoing model maintenance and updating with new data to ensure its continued effectiveness as the e-commerce landscape and delivery environment evolve.

By pursuing these future directions, this project can pave the way for a highly accurate and adaptable e-commerce delivery prediction system. This will ultimately lead to improved customer satisfaction, operational efficiency, and cost reduction for e-commerce businesses.

Appendix

GitHub Link: https://github.com/SN0212/E-commerce-Shipping-Prediction-Using-Machine-Learning.git

Project Demo Link:
https://drive.google.com/file/d/10eWkyR5qzZSCsS8_Ryhkm39VEaUeOd-S/view?usp=sharing