

Project Report

1. INTRODUCTION

1.1 Project Overview

The eCommerce shipping prediction project revolves around leveraging machine learning techniques to forecast the estimated shipping time for orders placed by customers. It begins with the collection of historical shipping data encompassing various order details such as destination, origin, weight, dimensions, chosen shipping method, carrier information, and actual delivery times. This comprehensive dataset forms the foundation for training and validating machine learning models.

1.2 Purpose

The eCommerce shipping prediction project utilizing machine learning techniques aims to revolutionize the accuracy and efficiency of order delivery timelines within online retail operations. Its foremost objective is to significantly enhance the customer experience by providing precise estimated shipping and delivery times. By harnessing historical shipping data encompassing various parameters like destination, shipping method, carrier, and delivery duration, the project seeks to create machine learning models capable of predicting shipping times with high accuracy. This accuracy translates into better managing customer expectations, reducing uncertainty around delivery dates, and ultimately bolstering overall customer satisfaction.

An integral purpose of this project is to optimize logistics and operational workflows within eCommerce businesses. Accurate shipping time predictions enable better planning and resource allocation, facilitating streamlined warehouse operations, inventory management, and route optimization. This optimization not only helps in reducing shipping delays but also assists in controlling associated costs, ultimately contributing to improved operational efficiency and resource utilization.

Furthermore, this endeavor focuses on harnessing the power of data-driven decision-making. By analyzing historical shipping data, identifying patterns, and understanding the factors influencing shipping times, the project aids in making informed decisions. It provides valuable insights that can guide inventory management strategies, warehouse distribution, and the selection of the most efficient shipping methods, thereby enhancing the overall business decision-making process.

2. LITERATURE SURVEY

2.1 Existing problem

In the realm of eCommerce shipping prediction, several persistent challenges and issues continue to pose hurdles, despite the advancements in machine learning and data analytics. One prevalent problem revolves around the variability and complexity of shipping data. The diverse range of factors influencing shipping times—such as geographical distance, shipping method, carrier efficiency, seasonal fluctuations, and unforeseen external factors—contributes to a highly dynamic and multifaceted dataset. Managing and effectively utilizing this varied data while addressing its heterogeneity and inconsistencies remain a significant challenge in developing accurate predictive models.

Moreover, the scarcity of standardized data across eCommerce platforms presents a notable obstacle. Variations in data formats, quality, and completeness among different eCommerce systems and carriers make it challenging to build robust, universally applicable shipping time prediction models. Integrating and harmonizing disparate data sources to create a cohesive dataset suitable for model training and generalization poses a considerable challenge.

2.2 References

1. Li, Z., Chen, S., & Sun, D. (2020). Machine Learning Methods for Predicting Delivery Time of Online Shopping. *IEEE Access*, 8, 165101-165110.
2. Liu, W., Jiang, L., & Shang, J. (2019). Predicting Delivery Time in E-commerce Using Machine Learning. *2019 International Conference on Intelligent Informatics and BioMedical Sciences (ICIIBMS)*.
3. Chen, Y., Lu, C. T., & Wang, Y. (2018). Predicting Delivery Time for E-commerce Orders. *2018 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*.
4. Verma, S., & Choudhary, P. (2021). Time Series Forecasting Techniques for Predicting Delivery Times in E-commerce. *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*.

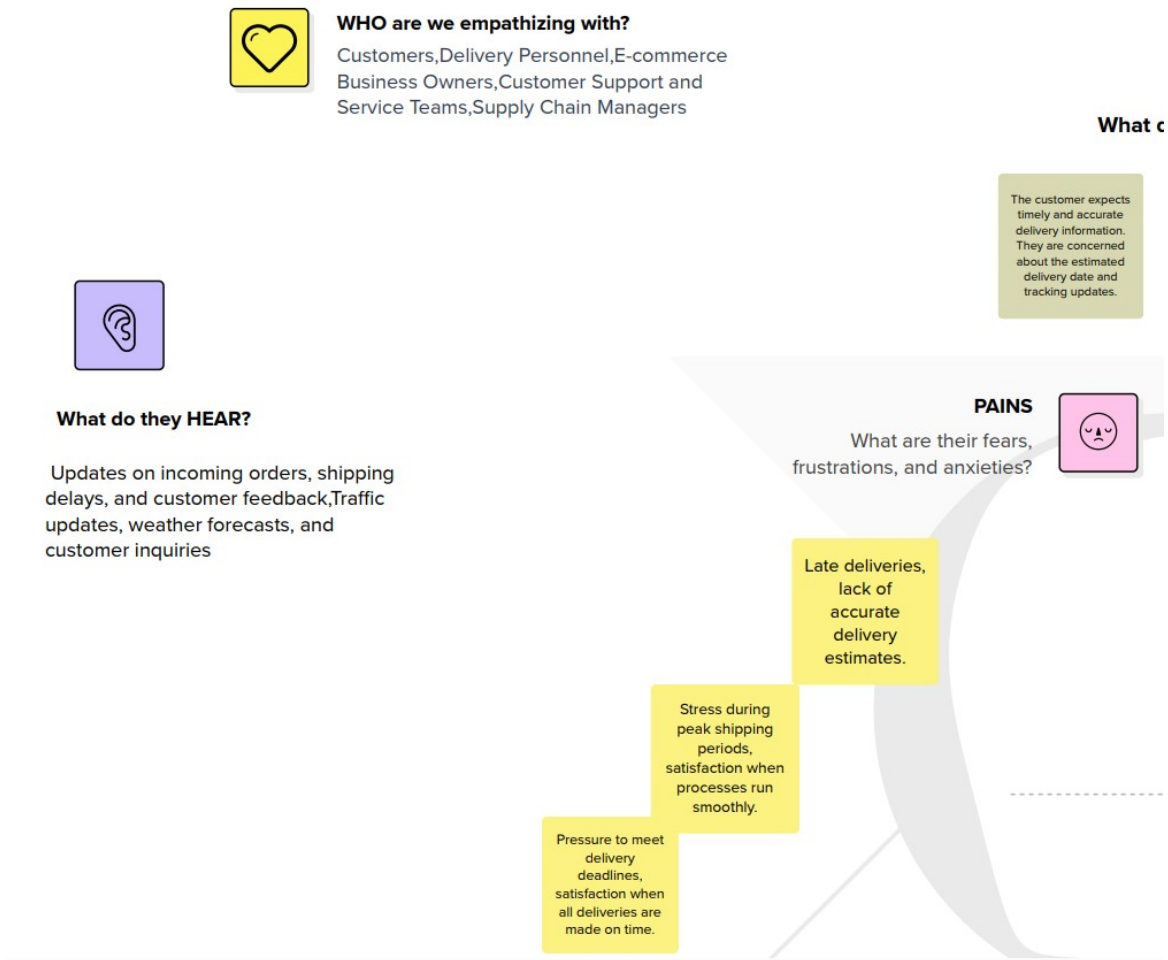
2.3 Problem Statement Definition

The challenge in eCommerce shipping lies in accurately predicting delivery times for orders placed on online platforms. Current methodologies often struggle to provide reliable estimated shipping times, leading to customer dissatisfaction, operational inefficiencies, and missed expectations. There is a need to develop a robust predictive system that leverages machine learning techniques to analyze historical shipping data, account for various influencing factors, and generate precise shipping time estimations in real-time. This system

should optimize logistics operations, enhance customer satisfaction, and empower businesses to make informed decisions regarding inventory management, resource allocation, and delivery planning. The overarching goal is to create a scalable and accurate predictive model that improves the overall eCommerce shipping experience for both businesses and customers.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



What do they THINK and FEEL?

The customer expects timely and accurate delivery information. They are concerned about the estimated delivery date and tracking updates.

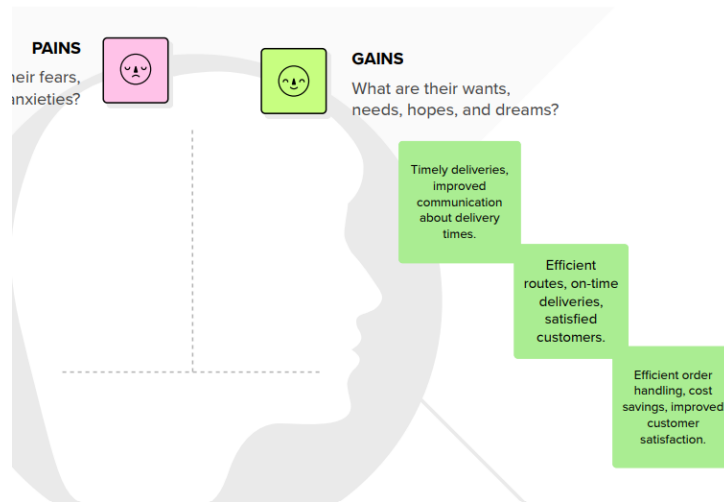
They might feel anxious if they don't receive updates or if the package is delayed. They may also feel relief when they see that their package is on its way.

What do they need to DO?

Receive orders on time for a seamless shopping experience.

Avoid the stress of late or uncertain deliveries.

Efficiently manage inventory to meet customer demand.



What do they SEE?

online ordering tracking interface


Warehouse layout

Delivery notifications

What are they watching and reading?

3.2 Ideation & Brainstorming

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

- 🕒 10 minutes to prepare
- 👥 1 hour to collaborate
- 👤 2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

🕒 10 minutes

A

Team gathering

Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B

Set the goal

Think about the problem you'll be focusing on solving in the brainstorming session.

C

Learn how to use the facilitation tools

Use the Facilitation Superpowers to run a happy and productive session.

[Open article](#) →

1 Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

🕒 5 minutes

Problem Statement

"Increasing customer satisfaction and optimizing operational efficiency by developing a machine learning model to predict accurate shipping times for e-commerce orders, addressing the challenges of unpredictability in delivery logistics."

Challenges

- Data Quality and Consistency
- Complexity of Supply Chain
- International Shipping Challenges
- Real-time Updates and Integration

Step-2: Brainstorm, Idea Listing and Grouping

2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

🕒 10 minutes

TIP
You can select a sticky note and hit the 'delete' button to erase it from your board.

Adharsh kumar Reddy

Use historical order and delivery data to identify patterns and trends, informing the model's predictions

Incorporate weather data to account for external factors affecting delivery times, like storms or traffic disruptions

Consider the geographical location of customers and warehouses to optimize routing and predict delivery times more accurately

Integrate real-time tracking information to adjust predictions dynamically based on the current status of the delivery

Yashwanth

Include customer-specific preferences and time constraints to tailor predictions for individual needs

Experiment with various machine learning algorithms (e.g., regression, neural networks) to find the most accurate model for your dataset

Leverage predictive analytics to foresee potential delays and proactively manage logistics

Work closely with logistics partners to gather insights and enhance the model with industry-specific knowledge

Rishiteja

Develop or integrate routing algorithms that consider traffic patterns, road closures, and other location-specific challenges

Predict maintenance needs for delivery vehicles, reducing the likelihood of unexpected delays due to breakdowns

Implement dynamic time windows for delivery predictions, adjusting estimates based on real-time conditions.

Build a system for automated, real-time communication with customers, keeping them informed about potential delays and reasons behind them.

Teja Sri

Establish a feedback loop where customer feedback on delivery experiences is used to continually improve and retrain the model.

Ensure the machine learning model's predictions are explainable, providing transparency to users and building trust in the system.

Explore blockchain technology to create a transparent and immutable record of the entire supply chain, enhancing trust and traceability.

Integrate predictive inventory management to ensure that products are readily available, reducing delays caused by stockouts.

3

Group Ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

🕒 20 minutes

TIP
Add customer labels to sticky notes to make it easier to find, organize, organize and categorize important ideas as they're added to your board.

Consider the geographical location of customers and warehouses to optimize routing and predict delivery times more accurately.

Predict maintenance needs for delivery vehicles, reducing the likelihood of unexpected delays due to breakdowns.

Develop or integrate routing algorithms that consider traffic patterns, road closures, and other location-specific challenges.

Build a system for automated, real-time communication with customers, keeping them informed about potential delays and reasons behind them.

Include customer-specific preferences and time constraints to tailor predictions for individual needs.

Explore blockchain technology to create a transparent and immutable record of the entire supply chain, enhancing trust and traceability.

Step-3: Idea Prioritization

4

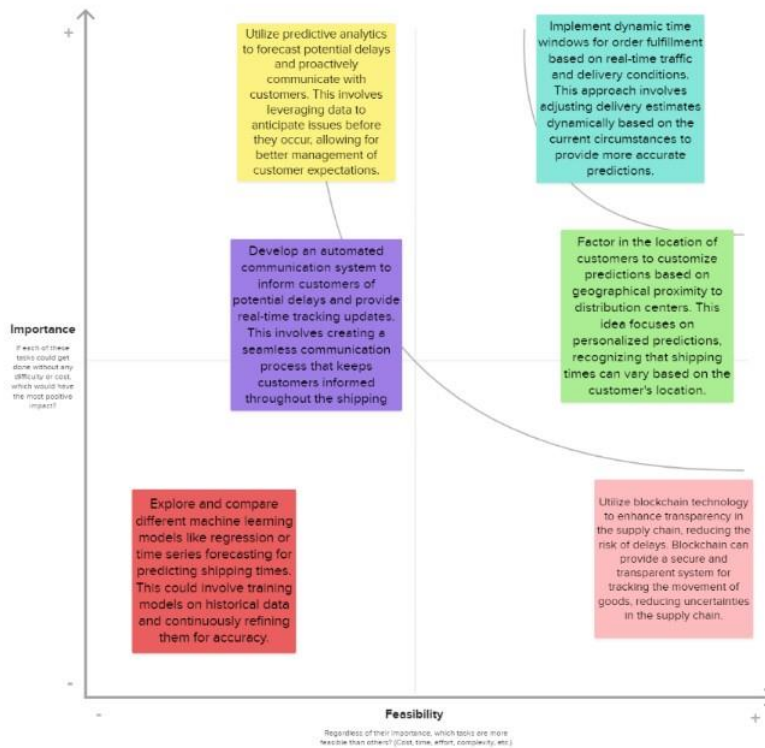
Prioritize

Your team should all be on the same page about what's important, moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

TP

Participants can use their cursor to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the H key on the keyboard.



→

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

- Share the mural**
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.
- Export the mural**
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

- Strategy blueprint**
Define the components of a new idea or strategy.
[Open the template →](#)
- Customer experience journey map**
Understand customer needs, motivations, and obstacles for an experience.
[Open the template →](#)
- Strengths, weaknesses, opportunities & threats**
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
[Open the template →](#)

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

- Prediction Accuracy:** The system should generate accurate estimated shipping times for orders based on historical data, considering various factors like distance, shipping method, carrier, and other relevant parameters.
- Integration with eCommerce Platform:** Seamlessly integrate the prediction system into the existing eCommerce platform, allowing for easy access and utilization of shipping time estimations.
- User-Friendly Interface:** Provide a user interface accessible to logistics managers, customer service representatives, or other authorized personnel to view predicted shipping times for orders.
- Feedback Loop:** Implement a mechanism to collect feedback on predicted versus actual shipping times, enabling continuous model improvement based on real-world performance.
- Alerts and Notifications:** Provide alerts or notifications to stakeholders in case of prediction failures, system downtime, or significant deviations in estimated shipping times.

4.2 Non-Functional requirements

1. **Response Time:** The system should provide predictions within a specified time frame (e.g., seconds) to ensure real-time or near-real-time estimations.
2. **Scalability:** Capability to scale horizontally or vertically to handle increased data volume or user load without compromising performance.
3. **Documentation:** Provide comprehensive documentation, including system architecture, data flow diagrams, and user manuals for system maintenance and troubleshooting.
4. **Data Encryption:** Utilize encryption techniques to protect sensitive shipping and customer data during transmission and storage.

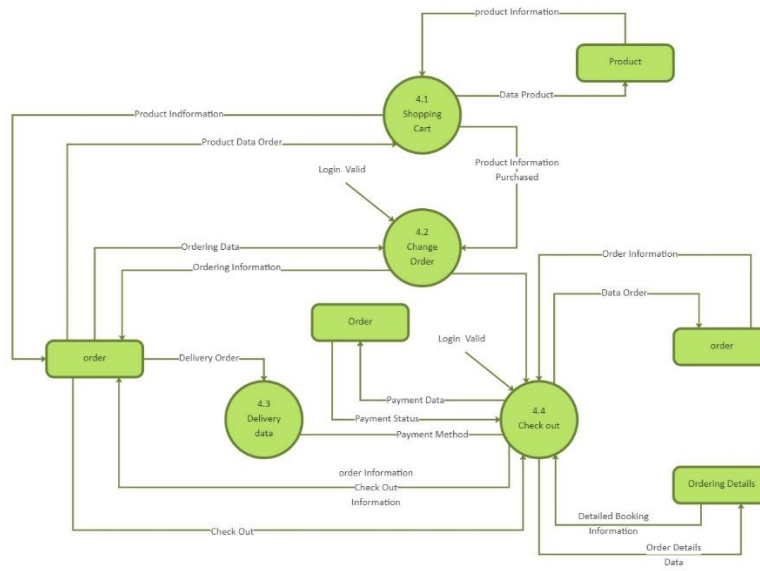
5. PROJECT DESIGN

5.1 Data Flow Diagrams & User Stories

Data Flow Diagrams are graphical representations that illustrate the flow of data within a system. In the context of an eCommerce shipping prediction system, DFDs serve as crucial visual tools for depicting how data moves through various processes and interactions within the system. These diagrams typically consist of entities (sources or destinations of data), processes (actions or transformations), data stores (where data is held), and data flows (the movement of data between entities, processes, and stores).

For an eCommerce shipping prediction system, a DFD might begin by illustrating the collection of data from multiple sources such as order databases, historical shipping records, and external data feeds. It would then showcase processes like data preprocessing, feature extraction, and model training, highlighting how the data is transformed and manipulated to generate shipping time predictions. Finally, the DFD could depict the delivery of predictions back to the eCommerce platform or user interfaces for accessibility and utilization.

Dataflow Diagram -



User Stories –

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Project Shipping Time	USN-1	AS a customer I want to know the estimated shipping time for my order	The system accurately predicts the shipping time based on the order details	High	Sprint-1
Seller	Update shipping Information	USN-2	AS a Seller I want to update the shipping information for a project	The system allows the update the shipping information and uses this information for future predictions	Medium	Sprint-2
System Admin	Monitor Prediction Accuracy	USN-3	AS a systemadmin I want to monitor the accuracy of the shipping time predictions	The system provides a dashboard showing the accuracy	Low	Sprint-3
Customer	Track order	USN-4	AS a Customer I want to track this status of my order	The system provides realtime updates on the status of the order	High	Sprint-2
Seller	View sales statistics	USN-5	As a seller I want to view statistics on my sales	The System provides a dashboard showing sales statistics	Medium	Sprint-4
System Admin	Manage Users	USN-6	As a system admin I want to manage user accounts	The system allows the admin to add remove or modify user accounts	Low	Sprint-5

Customer	Provide FeedBack	USN-7	As a customer I want to provide feedback on the shipping service.	The ystem allows the customer to provide feedback and uses this feedback to improve the prediction model	Medium	Sprint-4
Seller	View feedback	USN-8	As a seller I want to view customer feedback on the shipping service	The system provies a dashboard showing customer feedback	Low	Sprint-6

5.2 Solution Architecture

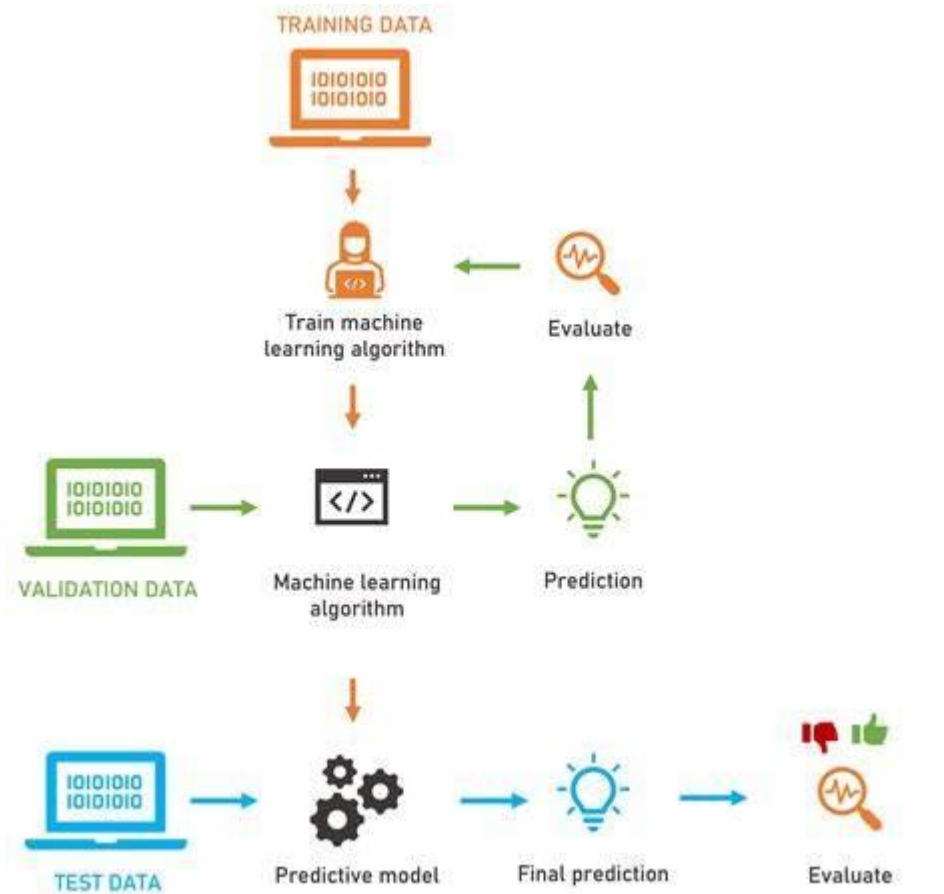
Steps involved:

- User interacts with the UI to enter the input.
- Entered input is analysed by the model which is integrated.
- Once model analyses the input the prediction is showcased on the UI

To accomplish this, we have to complete all the activities listed below,

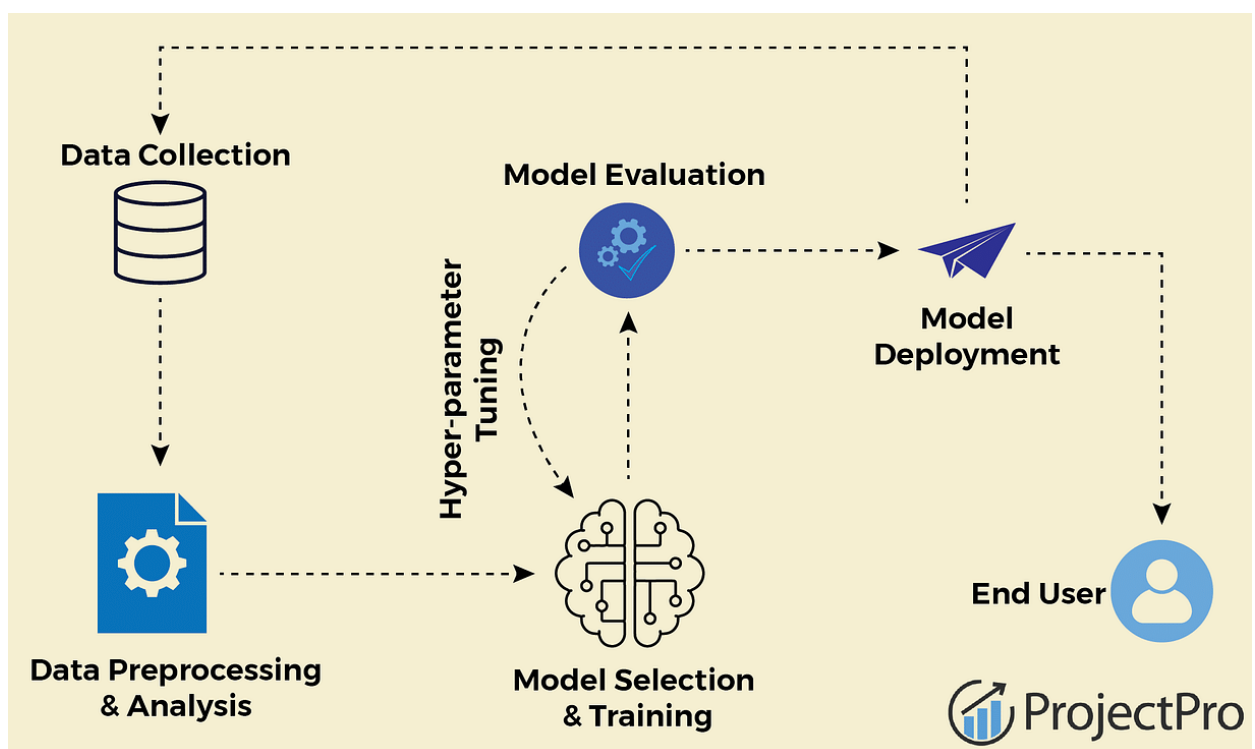
- Define Problem / Problem Understanding
 - 0 Specify the business problem
 - o Business requirements
 - o Literature Survey
 - o Social or Business Impact.
- Data Collection & Preparation
 - o Collect the dataset
 - o Data Preparation
- Exploratory Data Analysis
 - 0 Descriptive statistical
 - o Visual Analysis
- Model Building
 - 0 Training the model in multiple algorithms
 - o Testing the model
- Performance Testing & Hyperparameter Tuning
 - 0 Testing model with multiple evaluation metrics
 - o Comparing model accuracy before & after applying hyperparameter tuning
- Model Deployment
 - 0 Save the best model
 - o Integrate with Web Framework
- Project Demonstration & Documentation
 - 0 Record explanation Video for project end to end solution
 - o Project Documentation-Step by step project development procedure

Solution Architecture Diagram:



6. PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture



6.2 Sprint Planning & Estimation

Sprint	Functional Requirement	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Project Shipping Time	AS a customer I want to know the estimated shipping time for my order	1	High	Tejasri, Adharsh
Sprint-2	Update shipping Information	AS a Seller I want to update the shipping information for a project	2	Medium	Tejasri, Rishiteja
Sprint-3	Monitor Prediction Accuracy	AS a systemadmin I want to monitor the accuracy of the shipping time predictions	2	Low	Yashwanth, Adharsh
Sprint-2	Track order	AS a Customer I want to track this status of my order	3	High	Rishiteja Adharsh
Sprint-4	View sales statistics	As a seller I want to view statistics on my sales	4	Medium	Yashwanth Rishiteja
Sprint-5	Manage Users	As a system admin I want to manage user accounts	6	Low	Tejasri
Sprint-4	Provide FeedBack	As a customer I want to provide feedback on the shipping service	1	Medium	Yashwanth, Tejasri
Sprint-6	View feedback	As a seller I want to view customer feedback on the shipping service	1	Low	Yashwanth, Adharsh

6.3 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Sprint Release Date (Actual)
Sprint-1	3	8 Days	2 Nov 2023	9 Nov 2023	9 Nov 2023
Sprint-2	5	3 Days	7 Nov 2023	9 Nov 2023	9 Nov 2023

Sprint-3	10	3 Days	7 Nov 2023	9 Nov 2023	9 Nov 2023
Sprint-4	1	8 days	8 Nov 2023	15 Nov 2023	15 Nov 2023
Sprint-5	1	2 days	9 Nov 2023	10 Nov 2023	10 Nov 2023
Sprint-6	3	4 days	10 Nov 2023	13 Nov 2023	13 Nov 2023

7. CODING & SOLUTIONING

7.1. Features

- **Distance/Route:** The geographical distance between the sender and receiver can significantly impact shipping times. Incorporating route information or distance calculations could be helpful.
- **Shipping Method:** Different shipping methods (e.g., standard, express) have varying delivery times. Encoding these methods as features can enhance prediction accuracy.
- **Package Weight and Size:** Heavier or larger packages might take longer to deliver due to handling and transportation constraints.
- **Origin and Destination Locations:** Specific regions or countries may have different delivery timelines due to logistical differences.
- **Historical Shipping Data:** Past shipping records with timestamps can provide valuable insights into average delivery times based on similar parameters.
- **Carrier Information:** Different carriers might have different delivery times and efficiencies. Incorporating this information could be beneficial.
- **Weather Conditions:** Extreme weather can delay shipments. Integrating weather data relevant to the shipping route or destination can improve predictions.
- **Day of the Week/Seasonality:** Delivery times can vary based on the day of the week or holiday seasons. Encoding this temporal information can be valuable.
- **Traffic Conditions (for local delivery):** In urban areas, traffic congestion can affect local deliveries.

8. PERFORMANCE TESTING

8.1 Performance Metrics

1. Accuracy:- We have got training and testing accuracy as follows:

▼ Decision Tree Classifier

```
✓ [71] from sklearn.tree import DecisionTreeClassifier  
Ds dtree = DecisionTreeClassifier(random_state=0)  
dtree.fit(X_train, y_train)
```

```
• DecisionTreeClassifier  
DecisionTreeClassifier(random_state=0)
```

```
✓ [72] y_pred = dtree.predict(X_test)  
Ds print("Accuracy Score :", round(accuracy_score(y_test, y_pred)*100 ,2), "%")
```

```
⇒ Accuracy Score : 77.64 %
```

```
✓ [73] from sklearn.metrics import accuracy_score, f1_score, precision_score, recall_score  
Ds print('F-1 Score : ',(f1_score(y_test, y_pred)))  
print('Precision Score : ',(precision_score(y_test, y_pred)))  
print('Recall Score : ',(recall_score(y_test, y_pred)))
```

```
F-1 Score : 0.7717303005686433  
Precision Score : 0.818260120585702  
Recall Score : 0.7302075326671791
```

2. Classification report:-

▼ Random Forest Classifier

```
✓ [74] from sklearn.ensemble import RandomForestClassifier  
1s rfc = RandomForestClassifier(random_state=0)  
rfc.fit(X_train, y_train)
```

```
▼ RandomForestClassifier  
RandomForestClassifier(random_state=0)
```

```
✓ [75] y_pred = rfc.predict(X_test)  
0s print("Accuracy Score :", round(accuracy_score(y_test, y_pred)*100 ,2), "%")
```

Accuracy Score : 74.81 %

```
✓ [76] from sklearn.metrics import accuracy_score, f1_score, precision_score, recall_score  
0s print('F-1 Score : ',(f1_score(y_test, y_pred)))  
print('Precision Score : ',(precision_score(y_test, y_pred)))  
print('Recall Score : ',(recall_score(y_test, y_pred)))
```

```
📄 F-1 Score : 0.7097661623108666  
Precision Score : 0.8795454545454545  
Recall Score : 0.5949269792467333
```

9. RESULTS

9.1 Output Screenshots –

```
✓ [81] from sklearn.tree import DecisionTreeClassifier  
0s from sklearn.model_selection import train_test_split  
from sklearn.metrics import accuracy_score  
import joblib  
  
# Your code for splitting the data and training the Decision Tree Classifier  
X = df_upsampled.drop('Reached.on.Time_Y.N', axis=1)  
y = df_upsampled['Reached.on.Time_Y.N']  
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)  
  
dtree = DecisionTreeClassifier(random_state=0)  
dtree.fit(X_train, y_train)  
y_pred = dtree.predict(X_test)  
  
# Save the best model (Decision Tree Classifier)  
joblib.dump(dtree, 'best_model_decision_tree.pkl')  
  
['best_model_decision_tree.pkl']
```

```
✓ [82] loaded_model = joblib.load('best_model_decision_tree.pkl')  
0s # Use the loaded_model for predictions or evaluation
```

10. ADVANTAGES & DISADVANTAGES

Advantages:

- 1. Improved Customer Experience:** Accurate shipping time predictions help manage customer expectations, leading to higher satisfaction and trust in the eCommerce platform.
- 2. Optimized Logistics Operations:** Efficient predictions enable better resource allocation, route planning, and inventory management, reducing shipping delays and operational costs.
- 3. Enhanced Decision-Making:** Data-driven insights from predictive models aid in making informed decisions regarding inventory stocking, warehouse management, and shipping methods.
- 4. Competitive Edge:** Providing reliable estimated delivery times can serve as a competitive advantage, attracting and retaining customers in a highly competitive eCommerce market.
- 5. Continuous Improvement:** Feedback mechanisms allow for iterative improvements, ensuring the system adapts to changing shipping dynamics and customer needs.
- 6. Scalability:** Scalable architecture enables the system to handle increasing volumes of data and prediction requests, supporting business growth.

Disadvantages:

- 1. Data Complexity and Quality:** Managing diverse and complex shipping data sources can pose challenges in data preprocessing and ensuring data quality for accurate predictions.
- 2. Model Interpretability:** Complex machine learning models might lack interpretability, making it difficult to explain predictions, potentially impacting trust among users and stakeholders.
- 3. Overfitting or Underfitting:** Models might suffer from overfitting (fitting too closely to training data) or underfitting (oversimplifying), affecting prediction accuracy.

4. Dependency on Historical Data: Models heavily rely on historical data patterns, and unexpected shifts or disruptions not present in the training data might affect prediction accuracy.

5. System Integration Challenges: Integrating the prediction system with existing eCommerce platforms or third-party systems could pose technical challenges and compatibility issues.

6. Resource Intensiveness: Developing and maintaining machine learning models requires skilled personnel, computational resources, and continuous updates, which can be resource-intensive.

11. CONCLUSION

In conclusion, an eCommerce shipping prediction system leveraging machine learning presents a transformative solution to enhance the efficiency, accuracy, and customer satisfaction within the logistics domain of online retail. The system's advantages lie in its ability to provide accurate estimated shipping times, optimizing logistics operations, and empowering data-driven decision-making. By leveraging historical shipping data and predictive algorithms, this system facilitates improved customer experiences, operational excellence, and a competitive edge in the dynamic eCommerce landscape.

However, the implementation of such a system comes with its set of challenges. These challenges include managing complex and diverse data sources, ensuring model interpretability, addressing issues of overfitting or underfitting, and seamlessly integrating the prediction system within existing eCommerce platforms. Overcoming these hurdles requires meticulous data preprocessing, continuous model refinement, robust security measures, and effective collaboration among stakeholders and technical teams.

Despite these challenges, the benefits offered by an eCommerce shipping prediction system are substantial. It empowers businesses to optimize logistics, allocate resources effectively, and meet customer expectations with accurate estimated delivery times. The iterative nature of the system, incorporating feedback mechanisms and continuous improvements, allows for adaptation to evolving trends, thereby ensuring relevance and accuracy over time.

In essence, while there are complexities and obstacles to navigate, the advantages offered by an eCommerce shipping prediction system are significant. With strategic planning, technological innovation, and a commitment to refining models and processes, businesses can harness the transformative potential of these systems to revolutionize eCommerce logistics, improve customer satisfaction, and gain a competitive advantage in the market.

12. FUTURE SCOPE

The future prospects for an eCommerce shipping prediction system using machine learning are poised for significant advancements across various dimensions. Emerging developments in predictive algorithms are expected to elevate accuracy by integrating more intricate machine learning models, potentially delving into complex deep learning techniques. This evolution aims to refine predictive capabilities, enabling these systems to adapt dynamically to real-time alterations in demand, customer preferences, or unforeseen disruptions. Moreover, the integration of Internet of Things (IoT) devices and sensor data offers a promising avenue, allowing for the incorporation of live data streams into predictions, enhancing precision through real-time tracking and environmental considerations such as weather or traffic conditions. The upcoming emphasis on AI transparency aims to address interpretability concerns, ensuring models are more explainable, fostering trust, and aligning with ethical guidelines. Blockchain technology is also poised to play a pivotal role, ensuring secure and transparent supply chain management, while AI-powered customer service tools could transform communication by offering real-time shipping updates and proactive solutions. Furthermore, the future landscape anticipates a shift toward green logistics, emphasizing sustainable practices by optimizing delivery routes and methodologies to minimize environmental impact. These impending advancements herald a transformative era, propelling eCommerce shipping prediction systems towards greater accuracy, adaptability, sustainability, and customer-centric innovation.

13. APPENDIX

GitHub & Project Demo Link –

<https://github.com/smartinternz02/SI-GuidedProject-611628-1698727967>

https://drive.google.com/file/d/1kSjG5v6zqQYUbu_F6wKPJNfrR9fd2tv1/view?usp=sharing

Source Codes -

HTML –

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta
name="viewport"
content="width=device-
width, initial-scale=1.0">
  <title>Ecommerce
Shipping
Prediction</title>
  <style>
    body {
      font-family: Arial,
sans-serif;
      margin: 20px;
      background-
image:
url('https://www.electrich
ybridvehicletechnology.c
om/wp-
```

```
content/uploads/2020/10/
RPV-Rivian-Sept2020-
AEM04487-
cropped.jpg');
    background-size:
cover;
    }
    form {
        display: grid;
        grid-template-
columns: repeat(2, 1fr);
        gap: 20px;
        max-width:
800px;
        margin: auto;
        padding: 20px;
        background-color:
rgba(255, 255, 255, 0.9);
        border-radius:
10px;
        box-shadow: 0 0
10px rgba(0, 0, 0, 0.1);
    }
    label {
        display: block;
        margin-bottom:
5px;
        margin-top: 15px;
        font-weight: bold;
        color: darkblue;
    }
    input, select {
        display: block;
        margin-bottom:
8px;
        margin-top: 8px;
        width: 100%;
        padding: 8px;
    }
    button {
        grid-column: span
2;
        padding: 10px;
        background-color:
#4CAF50;
        color: white;
        border: none;
        margin-top: 15px;
        cursor: pointer;
    }
    #predictionResult {
        margin-top: 20px;
        font-weight: bold;
```

```

        grid-column: span
2;
        background-color:
rgba(255, 255, 255, 0.5);
        text-align:center;
        padding: 10px;
    }
    /* Styling for the two
columns */
    .column {
        display: flex;
        flex-direction:
column;
        margin:15px;
    }
    /* Styling for radio
buttons */
    .radio-group {
        margin-top:10px;
        display: flex;
        gap: 5px;
    }
    .radio-group label {
        color: black;
        font-weight:
normal;
    }
</style>
</head>
<body>
    <h1 style="color: #333;
text-align:
center">Ecommerce
Shipping Prediction</h1>
    <form
id="predictionForm"
method="POST"
action="/predict">

        <div
class="column">

            <!-- Form fields --
>

            <label
for="customerCareCalls"
>Customer Care
Calls:</label>

            <input
type="number"
id="customerCareCalls"
name="Customer_care_c
alls">

```

```
<label
for="costOfProduct">Cost
of the Product:</label>
```

```
<input
type="number" step="1"
id="costOfProduct"
name="Cost_of_the_prod
uct">
```

```
<label
for="priorPurchases">Pri
or Purchases:</label>
```

```
<input
type="number"
id="priorPurchases"
name="Prior_purchases"
>
```

```
<label
for="discountOffered">D
iscount Offered:</label>
```

```
<input
type="number" step="1"
id="discountOffered"
name="Discount_offered"
">
```

```
<label
for="weightInGrams">W
eight in Grams:</label>
```

```
<input
type="number" step="1"
id="weightInGrams"
name="Weight_in_grams"
">
```

```
</div>
```

```
<div
class="column">
```

```
<!-- Radio button
fields -->
```

```
<label
for="warehouseBlock">
Warehouse
Block:</label>
```

```
<div class="radio-
group">
```

```
<input
type="radio"
id="blockD"
name="warehouse_block"
" value="D">
```

```
<label
for="blockD">D</label>
```

```
<input
type="radio" id="blockF"
```

```

name="warehouse_block
" value="F">

    <label
for="blockF">F</label>

    <input
type="radio"
id="blockA"
name="warehouse_block
" value="A">

    <label
for="blockA">A</label>

    <input
type="radio" id="blockB"
name="warehouse_block
" value="B">

    <label
for="blockB">B</label>

    <input
type="radio" id="blockC"
name="warehouse_block
" value="C">

    <label
for="blockC">C</label>

</div>


    <label
for="modeOfShipment">
Mode of
Shipment:</label>

    <div class="radio-
group">

        <input
type="radio" id="flight"
name="Mode_of_shipme
nt" value="Flight">

        <label
for="flight">Flight</label
>

        <input
type="radio" id="ship"
name="Mode_of_shipme
nt" value="Ship">

        <label
for="ship">Ship</label>

        <input
type="radio" id="road"
name="Mode_of_shipme
nt" value="Road">

        <label
for="road">Road</label>

    </div>


    <label
for="productImportance"
>Product
Importance:</label>

```

```
<div class="radio-
group">
    <input
type="radio" id="low"
name="Product_importan
ce" value="low">
    <label
for="low">Low</label>
    <input
type="radio"
id="medium"
name="Product_importan
ce" value="medium">
    <label
for="medium">Medium<
/label>
    <input
type="radio" id="high"
name="Product_importan
ce" value="high">
    <label
for="high">High</label>
</div>
```

```
<label
for="gender">Gender:</l
abel>
<div class="radio-
group">
    <input
type="radio" id="female"
name="Gender"
value="Female">
    <label
for="female">Female</la
bel>
    <input
type="radio" id="male"
name="Gender"
value="Male">
    <label
for="male">Male</label>
</div>
```

```
<label
for="customerRating">C
ustomer Rating:</label>
<div class="radio-
group">
    <input
type="radio" id="rating1"
name="Customer_rating"
value="1">
    <label
for="rating1">1</label>
    <input
type="radio" id="rating2"
```

```

name="Customer_rating"
value="2">

    <label
for="rating2">2</label>

    <input
type="radio" id="rating3"
name="Customer_rating"
value="3">

    <label
for="rating3">3</label>

    <input
type="radio" id="rating4"
name="Customer_rating"
value="4">

    <label
for="rating4">4</label>

    <input
type="radio" id="rating5"
name="Customer_rating"
value="5">

    <label
for="rating5">5</label>
</div>

</div>

<button
type="submit">Predict</
button>

</form>

<!-- Display prediction
result -->

<div
id="predictionResult"></
div>

<script>

document.getElementByI
d('predictionForm').addE
ventListener('submit',
function(e) {

e.preventDefault(); //
Prevent the default form
submission

    // Get form data
    const formData =
new FormData(this);

    // Make a POST
request to the server
    fetch('/predict', {
        method:
'POST',

```



```

        body: formData
    })
    .then(response =>
response.text())
    .then(data => {
        // Display the
prediction result

document.getElementById('predictionResult').innerHTML = data;
    })
    .catch(error => {

console.error('Error:',
error);
    });
    });
</script>
</body>
</html>

```

Python(Flask application)-

```

import pickle
from flask import Flask, request, render_template

app = Flask(__name__)

# Load the model
model = pickle.load(open("best_model_decision_tree.pkl", "rb"))

# Define the mappings(For preprocessing)
warehouse_block_mapping = {'D': 0, 'F': 1, 'A': 2, 'B': 3, 'C': 4}
shipment_mapping = {'Flight': 0, 'Ship': 1, 'Road': 2}
product_importance_mapping = {'low': 0, 'medium': 1, 'high': 2}
gender_mapping = {'Female': 0, 'Male': 1}

@app.route('/')
def input():
    return render_template('index2.html')

@app.route('/predict', methods=['POST'])
def predict():
    try:
        warehouse_block = request.form["warehouse_block"]
        mode_of_shipment = request.form["Mode_of_shipment"]
        customer_care_calls = int(request.form["Customer_care_calls"])
        customer_rating = int(request.form["Customer_rating"])
        cost_of_the_product = float(request.form["Cost_of_the_product"])
        prior_purchases = int(request.form["Prior_purchases"])
        product_importance = request.form["Product_importance"]
        gender = request.form["Gender"]
        discount_offered = float(request.form["Discount_offered"])

```

```

weight_in_gms = float(request.form["Weight_in_grams"])

# Apply mappings(Preprocessing)
warehouse_block = warehouse_block_mapping.get(warehouse_block, warehouse_block)
mode_of_shipment = shipment_mapping.get(mode_of_shipment, mode_of_shipment)
product_importance = product_importance_mapping.get(product_importance, product_importance)
gender = gender_mapping.get(gender, gender)

preds = [[warehouse_block, mode_of_shipment, customer_care_calls, customer_rating,
           cost_of_the_product, prior_purchases, product_importance, gender,
           discount_offered, weight_in_gms]]
print("Form Data:", request.form)
prob = model.predict(preds)
reach = prob[0]
if reach == 1:
    return "Hurray! your product will reach in time."
else:
    return "Sorry! your product will take more time than expexcted to get delivered"

except Exception as e:
    error_message = f"An error occurred: {str(e)}"
    return error_message

if __name__ == '__main__':
    app.run(debug=True, port=4000)

```