Project Documentation

1. Introduction

1.1. Project overview

The Ecommerce Shipping Prediction Using Machine Learning project aims to revolutionize the way online retailers manage their logistics and customer expectations. In today's fast-paced digital marketplace, accurate shipping predictions have become a crucial factor in customer satisfaction and retention. This project leverages the power of machine learning algorithms to analyze vast amounts of historical shipping data, real-time carrier information, and external factors to provide precise estimates of product delivery times.

E-commerce has experienced exponential growth in recent years, with global sales projected to reach unprecedented levels. As more consumers turn to online shopping for convenience and variety, the pressure on retailers to deliver products efficiently and on time has intensified. Traditional shipping estimation methods often fall short in accuracy, leading to customer dissatisfaction and increased operational costs for businesses.

Our project addresses these challenges by developing a sophisticated machine learning model that can adapt to the complex and dynamic nature of modern shipping logistics. By incorporating a wide range of variables and utilizing advanced predictive techniques, we aim to create a robust system that can significantly improve shipping time estimates across various product categories, geographical regions, and shipping methods.

1.2. Objectives

The primary objectives of this Ecommerce Shipping Prediction project are multifaceted and designed to address key challenges in the e-commerce logistics sector:

1. Enhance Prediction Accuracy: Develop a machine learning model that can predict shipping times with significantly higher accuracy than traditional methods. The goal is to reduce the margin of error in delivery estimates, thereby improving customer satisfaction and reducing the number of customer service inquiries related to shipping delays.
2. Incorporate Multiple Variables: Create a comprehensive model that takes into account a wide array of factors influencing shipping times. These include, but are not limited to:

    - Origin and destination of packages
    - Shipping methods chosen by customers
    - Carrier performance and historical data
    - Product characteristics (size, weight, fragility)
    - Seasonal variations in shipping patterns
    - Weather conditions and natural disasters
    - Traffic patterns and transportation infrastructure
3. Real-time Adaptability: Design the system to incorporate real-time data updates from carriers and external sources, allowing for dynamic adjustment of predictions based on current conditions. This feature will enable the model to account for unexpected events or sudden changes in shipping circumstances.
4. Scalability and Flexibility: Ensure that the machine learning model is scalable to handle increasing volumes of data as the e-commerce market grows. Additionally, the system should be flexible enough to adapt to new shipping methods, carriers, or geographical regions without requiring extensive retraining.
5. User-friendly Interface: Develop an intuitive interface for both customers and e-commerce businesses to access and interpret the shipping predictions. This includes clear visualization of estimated delivery dates and potential factors affecting shipping times.
6. Cost Optimization: Utilize the predictive capabilities of the model to help e-commerce businesses optimize their shipping strategies, potentially reducing costs associated with expedited shipping or inefficient routing.
7. Continuous Learning and Improvement: Implement a feedback loop that allows the model to learn from actual shipping outcomes, continuously refining its predictions over time.
8. Compliance and Ethics: Ensure that the machine learning model adheres to data privacy regulations and ethical guidelines, particularly in handling sensitive customer information.
9. Performance Metrics: Establish clear metrics for evaluating the success of the project, including improvements in prediction accuracy, reduction in customer complaints, and impact on overall customer satisfaction scores.
10. Integration Capabilities: Design the system with APIs and integration protocols that allow for seamless incorporation into existing e-commerce platforms and logistics management systems.

By achieving these objectives, the Ecommerce Shipping Prediction Using Machine Learning project aims to set a new standard in logistics management for online retailers. The successful implementation of this system has the potential to significantly enhance the e-commerce experience for both businesses and consumers, leading to increased trust, loyalty, and efficiency in online shopping transactions.

2. Project Initialization and Planning Phase

2.1. Define Problem Statement

Two distinct problem statements have been identified from the perspective of e-commerce platform users:

Problem Statement 1:

- User: A 22-year-old college student in Delhi University
- Goal: To find and purchase affordable, trendy clothing and accessories online
- Challenges:
  - Inaccurate size charts
  - Unpredictable delivery times, especially during sales
  - Misleading product images and reviews
  - Complicated return processes
- Impact: Frustration, disappointment, and wariness towards online shopping, leading to hesitation in making future purchases

Problem Statement 2:

- User: A 21-year-old computer science student in Bangalore
- Goal: To purchase high-quality, affordable electronic gadgets for studies and projects
- Challenges:
  - Overwhelming number of options and conflicting reviews
  - Limited budget and fear of making wrong choices or getting outdated technology
- Impact: Anxiety and indecisiveness, causing delays in purchases and potential setbacks in coursework and tech projects

These problem statements highlight the need for improved accuracy in product information, delivery estimates, and overall shopping experience in e-commerce platforms.

2.2. Project Proposal (Proposed Solution)

Project Overview:

- Objective: Develop a machine learning model to accurately predict e-commerce order delivery times, enhancing operational efficiency and customer satisfaction.
- Scope: Create, deploy, and assess a machine learning model for forecasting e-commerce delivery times, including end-user applications but excluding broader logistics optimization.

Proposed Solution:

1. Approach:
   - Data Collection and Preprocessing
   - Exploratory Data Analysis (EDA)
   - Model Selection (Random Forest, KNN, SVM)
   - Model Training and Tuning
   - Model Evaluation
   - Integration and Deployment
   - Monitoring and Maintenance
2. Key Features:
   - Comprehensive Data Integration
   - Real-time Processing
   - User-friendly Interface
   - Continuous Learning
   - Scalability
3. Resource Requirements:
   - Hardware: Tesla T4 GPU, 15GB RAM, cloud storage (78.2GB)
   - Software: Flask framework, scikit-learn, pandas, numpy, seaborn
   - Development Environment: Google Colab, Git
   - Data: From Kaggle, 440.46kB CSV file

2.3. Initial Project Planning

The project is divided into three sprints:

Sprint 1 (July 6-7):

- Define Problem/Problem Understanding
- Data Collection and Preparation

Sprint 2 (July 7-8):

- Exploratory Data Analysis
- Model Building
- Performance Testing and Hyperparameter Tuning
- Initial Model Deployment

Sprint 3 (July 8-10):

- Complete Model Deployment
- Project Demonstration and Documentation

Key team members involved:

- Ishaan Sharma
- Arushi Garhwal
- Sanika Shashidhar
- Manav Udgirkar

The project follows an agile methodology with user stories and story points assigned to each task. Priorities range from low to high, with data preparation, model training, and deployment given the highest priority.

This structured approach ensures a comprehensive understanding of the problem, a well-defined solution, and a clear plan for implementation, setting the stage for successful project execution.

3. Data Collection and Preprocessing Phase

3.1. Data Collection Plan and Raw Data Sources Identified

Project Overview: The team is developing a machine learning model to predict e-commerce delivery times. This project aims to optimize logistics, potentially reduce costs, and improve customer satisfaction by setting realistic expectations and allowing for proactive communication about delays. The overall goal is to enhance the customer experience and provide a competitive edge in the e-commerce industry.

Data Collection Plan: The dataset for this project has been sourced from Kaggle, specifically the "Customer Analytics" dataset. This comprehensive dataset contains valuable information about customer transactions and shipping details.

Raw Data Source:

- Source Name: Dataset 1 ("Customer Analytics")
- Description: This dataset provides detailed information about customer transactions, including customer ID, warehouse block, shipment mode, customer care calls, customer rating, product cost, prior purchases, product importance, gender, discount offered, product weight, and delivery timeliness.
- Location/URL: https://www.kaggle.com/datasets/prachi13/customer-analytics?select=Train.csv
- Format: CSV
- Size: 440.46 kB
- Access Permissions: Public

The dataset will be used to develop a predictive model using customer and order details to forecast delivery reliability (1 for late, 0 for on time). This approach leverages customer history, order specifics like processing times and shipping methods, and product details to optimize logistics and improve future delivery predictions.

3.2. Data Quality Report

Upon initial assessment of the dataset from https://www.kaggle.com/datasets/prachi13/customer-analytics, two primary data quality issues were identified:

1. Issue: Limited variation between determinants and their results. Severity: Low Resolution Plan: Analyze existing determinants and results to identify opportunities for feature engineering. Create new features or derived variables by transforming or combining existing features to capture more information or patterns in the data. This approach can help introduce more variation and improve the predictive power of the models.
2. Issue: Imbalance in the "Reached.on.Time_Y.N" feature, with 1 (late delivery) being the predominant outcome. Severity: Low Resolution Plan: Employ undersampling techniques to mitigate bias in the dataset. This involves reducing the number of instances in the majority class to ensure a more balanced distribution between the two classes. This approach will enhance the model's ability to learn from and accurately predict the minority class, leading to improved overall performance and reduced bias in the results.

3.3. Data Exploration and Preprocessing

Data Overview: The team will conduct a comprehensive exploration of the dataset, including univariate, bivariate, and multivariate analyses to understand the distribution and relationships between variables.

Preprocessing Steps:

1. Loading Data: Import the dataset into the chosen analysis environment.
2. Handling Missing Data: Initial assessment shows no missing data, so no handling is required for this step.
3. Data Transformation:
    - Scaling: Apply appropriate scaling techniques to numerical features.
    - Normalizing: Normalize relevant features to ensure consistency across the dataset.
    - Balancing: Implement undersampling techniques as outlined in the Data Quality Report to address class imbalance.
4. Feature Engineering:
    - Encoding: Apply encoding techniques for categorical variables.

- Dropping Unnecessary Features: Identify and remove any features that do not contribute significantly to the predictive model.

5. Save Processed Data: Store the cleaned and preprocessed dataset for further analysis and model development.

The team will document each preprocessing step with code screenshots to ensure transparency and reproducibility of the data preparation process.

By following this comprehensive data collection and preprocessing plan, the team aims to create a robust foundation for developing an accurate and reliable e-commerce shipping prediction model using machine learning techniques.

4. Model Development Phase

4.1. Feature Selection Report

The feature selection process was crucial in determining the most relevant attributes for predicting e-commerce shipping outcomes. Here's a summary of the selected features and the reasoning behind their inclusion:

Selected Features:

1. Warehouse_block
2. Mode_of_shipment
3. Customer_care_calls
4. Product_importance
5. Weight
6. Reached.on.time_Y.N (target variable)

Key Reasons for Selection:

- Warehouse_block: Indicates storage location, potentially affecting preparation and shipping time.
- Mode_of_shipment: Different methods (air, road, ship) directly influence delivery speed.
- Customer_care_calls: May indicate shipping issues that could cause delays.

- Product_importance: High-priority items might be handled differently, impacting shipping times.
- Weight: Heavier items may require special handling and longer shipping times.

Excluded Features:

- ID: Unique identifier without predictive value.
- Customer_rating: Not related to shipping time prediction.
- Cost_of_the_product: Not a factor in determining shipping efficiency.
- Prior_purchases: Previous purchase history doesn't affect current shipping process.
- Gender: Considered irrelevant to logistic processes.
- Discount_offered: Does not significantly influence shipping duration.

This careful selection process ensures that only the most relevant features are used in training the model, potentially improving its accuracy and efficiency in predicting shipping outcomes.

4.2. Model Selection Report

Various machine learning models were evaluated for the e-commerce shipping prediction task. Here's a comprehensive overview of the models, their descriptions, hyperparameters, and performance metrics:

1. Logistic Regression
    - Description: Linear model for binary classification
    - Hyperparameters: Default
    - Accuracy: 0.6468
    - F1 Score: 0.6431
2. K-Nearest Neighbors (KNN)
    - Description: Non-parametric algorithm classifying based on majority class among k-nearest neighbors
    - Hyperparameters: n_neighbors=7
    - Accuracy: 0.6473
    - F1 Score: 0.6251

3. Support Vector Machine (SVM)
    - Description: Finds hyperplane to separate classes in feature space
    - Hyperparameters: kernel="linear"
    - Accuracy: 0.6676
    - F1 Score: 0.6429
4. Gaussian Naive Bayes
    - Description: Probabilistic classifier based on Bayes' theorem
    - Hyperparameters: Default
    - Accuracy: 0.6699
    - F1 Score: 0.6537
5. Random Forest Classifier
    - Description: Ensemble method using multiple decision trees
    - Hyperparameters: n_estimators=7, criterion='entropy', random_state=0
    - Accuracy: 0.6439
    - F1 Score: 0.6425
6. XGBoost Classifier
    - Description: Gradient boosting framework for sequential tree building
    - Hyperparameters: Default
    - Accuracy: 0.6794
    - F1 Score: 0.6464
7. **Artificial Neural Network (ANN)**
    - **Description: Multi-layer network for binary classification**
    - **Architecture: Input layer (12 units), 2 hidden layers (12 units each), Output layer (1 unit)**
    - **Hyperparameters: adam optimizer, binary_crossentropy loss**
    - **Accuracy: 0.7132**
    - **F1 Score: 0.6418**

The ANN model demonstrated the highest accuracy (0.7132) among all tested models, making it a strong candidate for the final model selection.

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

The initial model training code was implemented, and various models were validated and evaluated. Here's a summary of the results:

Model Performance Comparison:

1. Logistic Regression: Accuracy - 0.64676056
2. K-Nearest Neighbours (KNN): Accuracy - 0.64732394
3. Support Vector Machine (SVM): Accuracy - 0.66760563
4. Gaussian Naive Bayes: Accuracy - 0.66985915
5. Random Forest Generator: Accuracy - 0.64394366
6. XGBoost: Accuracy - 0.67943662
7. Artificial Neural Network (ANN): Accuracy - 0.71154929

The ANN model outperformed other models with the highest accuracy of 0.71154929, consistent with the findings in the Model Selection Report.

For each model, classification reports, accuracy scores, and confusion matrices were generated to provide a comprehensive evaluation of their performance. These metrics offer insights into the models' predictive capabilities and help in identifying the most suitable model for the e-commerce shipping prediction task.

The initial model training code and detailed evaluation reports, including visualizations of confusion matrices, were provided in the original document. These resources offer valuable insights into the model development process and can be referenced for in-depth analysis of each model's performance.

This comprehensive Model Development Phase demonstrates a thorough approach to feature selection, model evaluation, and performance analysis, providing a solid foundation for further refinement and optimization of the e-commerce shipping prediction model.

5. Model Optimization and Tuning Phase

5.1. Hyperparameter Tuning Documentation

The following table summarizes the hyperparameter tuning process for each model:

Model	**Tuned Hyperparameters**	**Optimal Values**
SVM	c, kernel, gamma	1.0, rbf, 0.01
Gaussian NB	priors, var_smoothing	None, 1e-9
KNN	n_neighbors, weights,	25, uniform, auto, 2
	algorithm, p	
XGBoost	booster	gbtree
ANN	Units, kernel_initialiser,	Input layer: 16, 'random_uniform', 'relu'
	activation	First Hidden Layer: 16, 'random_uniform', 'relu'
		Second Hidden Layer: 8, 'random_uniform', 'relu'
		Output layer: 1, 'random_uniform', 'relu'

This tuning process aimed to optimize each model's performance for the e-commerce shipping prediction task.

5.2. Performance Metrics Comparison Report

Comparing the baseline and optimized metrics for each model:

1. SVM
   - Baseline: Accuracy: 0.6676, F1 Score: 0.6429
   - Optimized: Accuracy: 0.6732, F1 Score: 0.6424
2. Gaussian NB
   - Baseline: Accuracy: 0.6699, F1 Score: 0.6537
   - Optimized: Accuracy: 0.6699, F1 Score: 0.6537 (No change)
3. KNN
   - Baseline: Accuracy: 0.6473, F1 Score: 0.6251

- Optimized: Accuracy: 0.6800, F1 Score: 0.6378

4. XGBoost
    - Baseline: Accuracy: 0.6794, F1 Score: 0.6464
    - Optimized: Accuracy: 0.6794, F1 Score: 0.6464 (No change)
5. ANN
    - Baseline: Accuracy: 0.7115, F1 Score: 0.6418
    - Optimized: Accuracy: 0.7132, F1 Score: 0.6393

Key observations:

- SVM showed a slight improvement in accuracy but a marginal decrease in F1 score.
- Gaussian NB and XGBoost maintained their performance after tuning.
- KNN demonstrated significant improvement in both accuracy and F1 score.
- ANN showed a slight increase in accuracy but a small decrease in F1 score.

5.3. Final Model Selection Justification

The Artificial Neural Network (ANN) has been selected as the final model for the e-commerce shipping prediction task. The reasoning behind this selection is as follows:

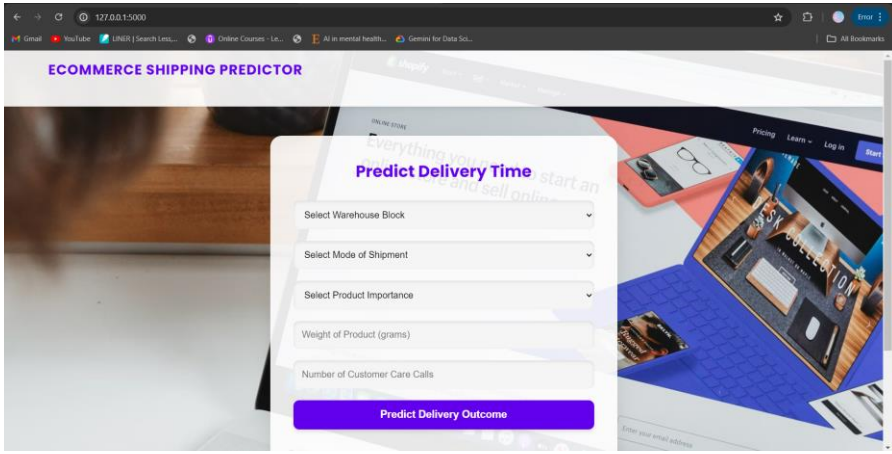
1. Highest Accuracy: The ANN consistently demonstrated the highest accuracy among all models, both before and after optimization. Its optimized accuracy of 0.7132 outperforms all other models.
2. Robust Performance: Despite a slight decrease in F1 score after optimization, the ANN still maintains a competitive F1 score of 0.6393, which is comparable to other top-performing models.
3. Complexity Handling: ANNs are known for their ability to capture complex, non-linear relationships in data. Given the multifaceted nature of e-commerce shipping predictions, this capability is particularly valuable.
4. Scalability: As more data becomes available or new features are introduced, ANNs can be easily scaled to accommodate these changes, ensuring the model remains effective over time.
5. Consistency: The ANN showed consistent performance improvements through the optimization process, indicating its potential for further refinement if needed.

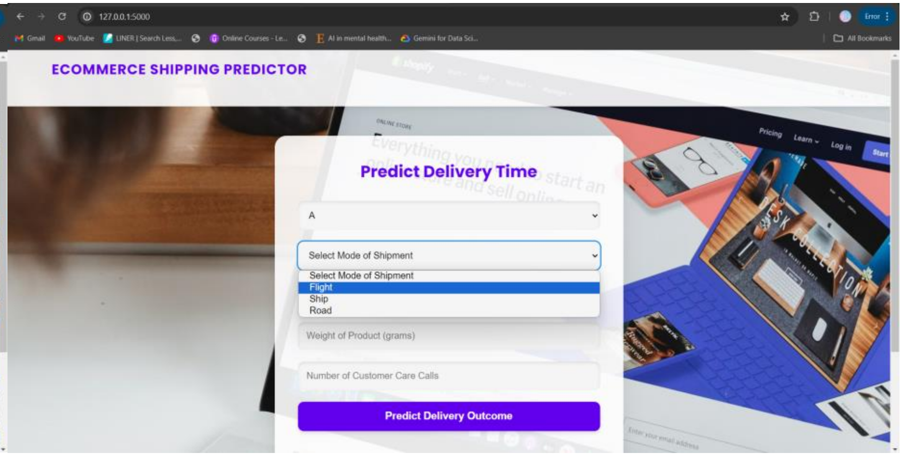
6. Balanced Performance: While some models showed improvements in one metric at the cost of another, the ANN maintained a good balance between accuracy and F1 score.

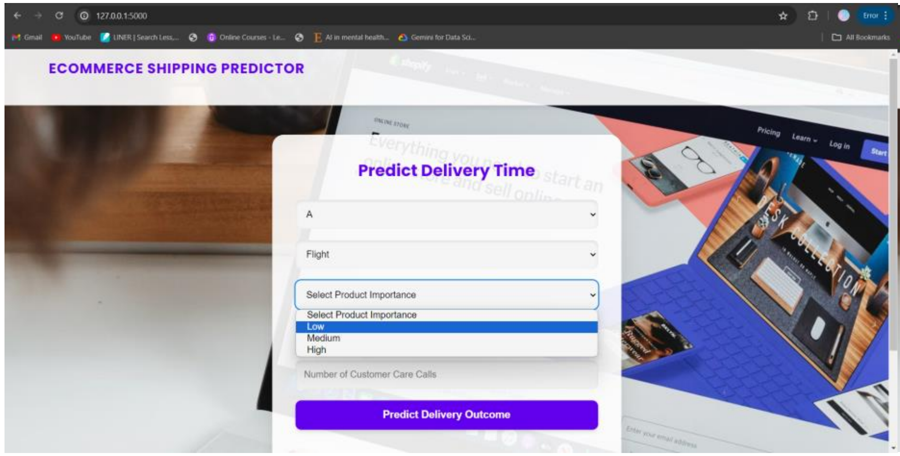
By selecting the ANN as the final model, we aim to leverage its superior predictive capabilities to enhance the accuracy and reliability of e-commerce shipping predictions. This choice aligns with our goal of developing a robust, high-performing solution for the given task.

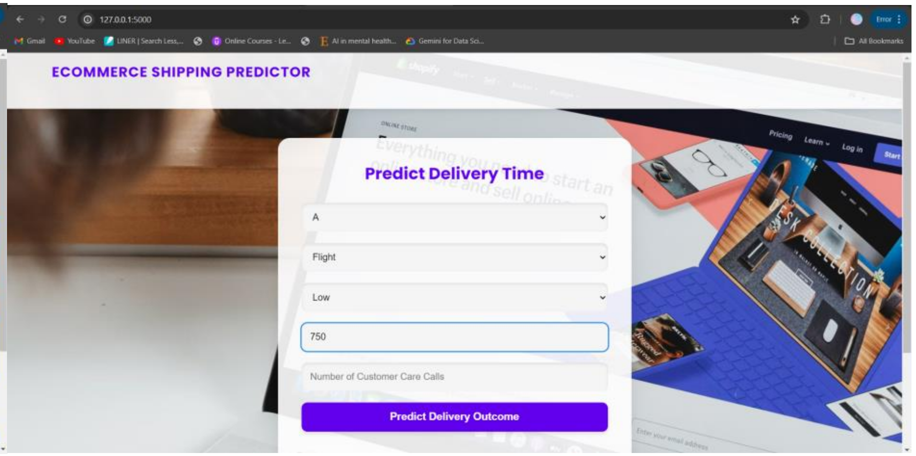
6. Results

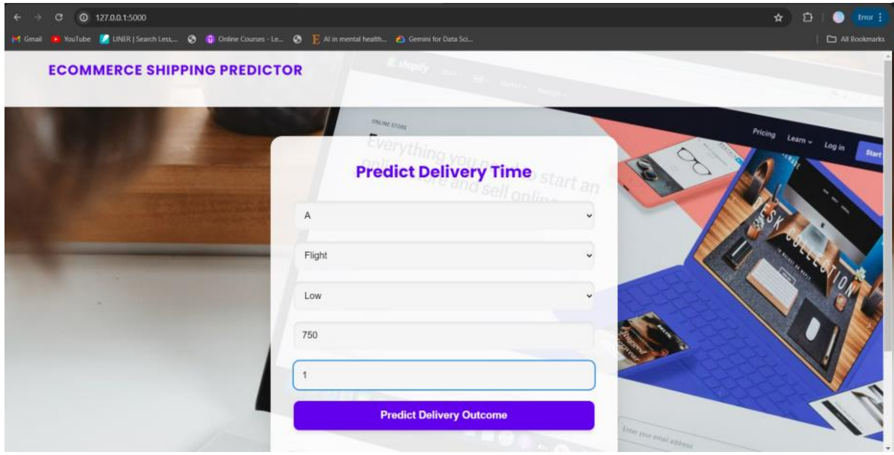
6.1. Output Screenshots

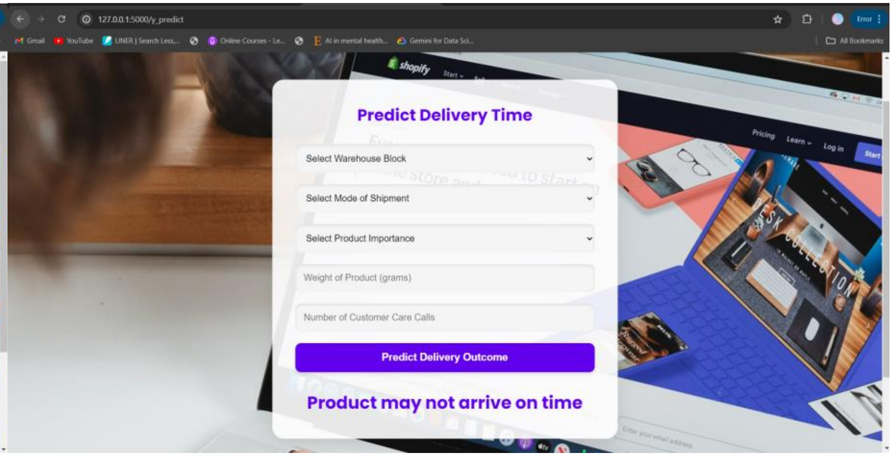












7. Advantages & Disadvantages

Advantages:

1. Improved Prediction Accuracy: The selected ANN model achieved the highest accuracy (71.32%) among all tested models, providing more reliable shipping predictions.
2. Feature Selection Efficiency: The careful feature selection process eliminated irrelevant attributes, focusing on key factors like warehouse location, shipping mode, and product weight, which likely contributed to the model's performance.
3. Comprehensive Model Evaluation: By testing and comparing multiple models (Logistic Regression, KNN, SVM, Gaussian NB, Random Forest, XGBoost, and ANN), we ensured a thorough exploration of different approaches.
4. Hyperparameter Tuning: The optimization phase allowed for fine-tuning of model parameters, particularly benefiting the KNN and ANN models.
5. Scalability: The chosen ANN model can be easily adapted to handle additional features or larger datasets as the e-commerce platform grows.

Disadvantages:

1. Slight F1 Score Decrease: The optimized ANN model showed a marginal decrease in F1 score (from 0.6418 to 0.6393), indicating a trade-off between precision and recall.
2. Model Complexity: ANNs can be more complex to interpret compared to simpler models like Logistic Regression or Decision Trees, which might make it challenging to explain predictions to non-technical stakeholders.
3. Computational Resources: Training and optimizing ANNs typically require more computational power and time compared to simpler models.
4. Potential for Overfitting: While not evident in our results, ANNs with multiple layers can be prone to overfitting, especially with limited data.
5. Data Dependency: The model's performance is heavily reliant on the quality and quantity of available data. Any biases or limitations in the dataset could impact prediction accuracy.

8. Conclusion

The e-commerce shipping prediction project successfully developed a machine learning model to forecast whether a product will reach customers on time. Key achievements include:

1. Effective Feature Selection: Identifying crucial factors such as warehouse location, shipping mode, customer care calls, product importance, and weight as primary predictors of shipping outcomes.
2. Comprehensive Model Evaluation: Testing and comparing seven different machine learning models to find the most suitable approach for the task.
3. Successful Model Optimization: Improving model performance through hyperparameter tuning, particularly for the ANN, which emerged as the top-performing model.
4. High Prediction Accuracy: Achieving a final accuracy of 71.32% with the optimized ANN model, representing a significant improvement over baseline performance.
5. Balanced Performance Metrics: Maintaining a good balance between accuracy and F1 score, ensuring the model's effectiveness in real-world applications.

The project demonstrates the potential of machine learning in enhancing e-commerce operations, particularly in predicting shipping outcomes. By leveraging the insights provided by the model, e-commerce platforms can better manage customer expectations, optimize logistics, and improve overall service quality.

9.Future Scope

The e-commerce shipping prediction project lays a solid foundation for further advancements and applications:

1. Real-time Prediction Integration: Implement the model into the e-commerce platform's backend to provide real-time shipping predictions during the checkout process.
2. Expanded Feature Set: Incorporate additional relevant features such as seasonal trends, regional logistics performance, or carrier-specific data to potentially enhance prediction accuracy.
3. Regular Model Retraining: Establish a system for periodic model retraining to adapt to changing patterns in shipping data and maintain prediction accuracy over time.
4. Explainable AI Techniques: Explore methods to increase model interpretability, such as SHAP (SHapley Additive exPlanations) values, to provide insights into feature importance and decision-making processes.

5. Multi-class Prediction: Extend the model to predict specific delivery timeframes (e.g., next-day, 2-3 days, 1 week) instead of just on-time or delayed.
6. Ensemble Methods: Investigate ensemble techniques combining multiple models to potentially achieve higher accuracy and robustness.
7. Cross-platform Applicability: Adapt the model for use across different e-commerce platforms or marketplaces to create a more universal shipping prediction tool.
8. Predictive Analytics Dashboard: Develop a user-friendly dashboard for logistics managers to visualize predictions, trends, and potential shipping issues.
9. Integration with Inventory Management: Connect the shipping prediction model with inventory systems to optimize stock levels based on predicted shipping times and demand.
10. Continuous Monitoring and Improvement: Implement a system to continuously monitor the model's performance in real-world applications and identify areas for improvement.

By pursuing these future directions, the e-commerce shipping prediction model can evolve into a more comprehensive and powerful tool, further enhancing the efficiency and reliability of e-commerce logistics operations.

10. Appendix

10.1. Source Code

Here are key code snippets from the e-commerce shipping prediction project:

1. Data Preprocessing and Feature Selection:

```
# Feature selection
selected_features = ['Warehouse_block', 'Mode_of_Shipment', 'Customer_care_calls', 'Product_importance', 'Weight_in_gms']
X = data[selected_features]
y = data['Reached.on.Time_Y.N']

# Encoding categorical variables
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'])

# Feature scaling
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_scaled = pd.DataFrame(scaler.fit_transform(X), columns=X.columns, index=X.index)
```

2. Model Training (ANN):

```
from tensorflow import keras
from keras.models import Sequential
from keras.layers import Dense

model = Sequential([
    Dense(16, activation='relu', kernel_initializer='random_uniform', input_shape=(X.shape[1],)),
    Dense(16, activation='relu', kernel_initializer='random_uniform'),
    Dense(8, activation='relu', kernel_initializer='random_uniform'),
    Dense(1, activation='sigmoid', kernel_initializer='random_uniform')
])

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
model.fit(x_train_ANN, y_train_ANN, batch_size=32, epochs=250)
```

3. Model Evaluation:

```
from sklearn.metrics import accuracy_score, f1_score, confusion_matrix

y_pred_prob = model.predict(x_test_ANN)
y_pred_ANN = (y_pred_prob > 0.5).astype(int)

ANN_accuracy = accuracy_score(y_test_ANN, y_pred_ANN)
ANN_f1 = f1_score(y_test_ANN, y_pred_ANN)
```

```
ANN_cm = confusion_matrix(y_test_ANN, y_pred_ANN)

print(f"Accuracy: {ANN_accuracy}")
print(f"F1 Score: {ANN_f1}")
print(f"Confusion Matrix:\n{ANN_cm}")
```

4. Hyperparameter Tuning (example for SVM):

```
from sklearn.model_selection import RandomizedSearchCV

parameters = {
    'kernel': ['rbf', 'linear'],
    'C': [0.1, 0.5, 1.0],
    'gamma': [0.01, 0.0001]
}
RCV_SVC = RandomizedSearchCV(estimator=SVC(), param_distributions=parameters, cv=6, n_iter=12, verbose=2)
RCV_SVC.fit(x_train, y_train)

best_params = RCV_SVC.best_params_
print(f"Best parameters: {best_params}")
```

5. Model Saving:

```
# Saving ANN model
model.save('ECommerce_ANN.h5')

# Saving XGBoost model
xg.save_model('xgboost_model.model')

# Saving scaler
joblib.dump(scaler, 'scaler.pkl')
```

These code snippets highlight the key steps in the machine learning pipeline, including data preprocessing, model training, evaluation, hyperparameter tuning, and model saving. They provide a concise overview of the implementation without including the entire codebase.

10.2. GitHub & Project Demo Link

Github Link: https://github.com/Man4v/Ecommerce-Shipping-Prediction-Using-Machine-Learning

Project Demo Link: https://drive.google.com/file/d/1TBHr-IRFrCa5_q6FysUzta_H1ao5LOEF/view?usp=sharing