HW3 Report

Ruijie Sun

(a)The problem statement

My problem is to predict whether an e-commerce shipment will arrive on time or not. This prediction is crucial for maintaining customer satisfaction and optimizing logistic operations.

(b) Data sources

The dataset used for this analysis, ‘e-commerce_dataset.csv’, contains various features related to e-commerce shipments. I got this dataset from Kaggle.

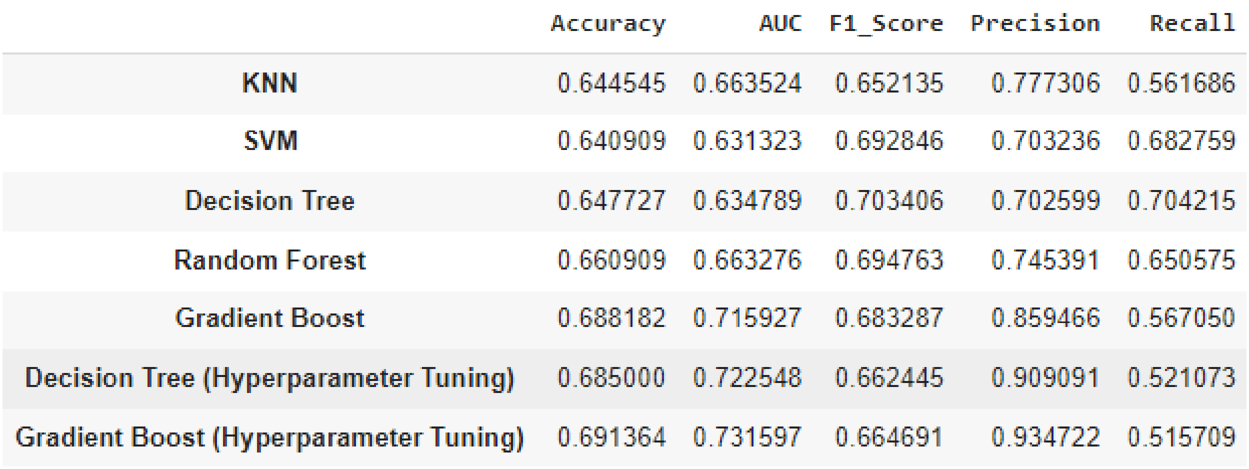
(c) Feature engineering and preprocessing done

Data Cleaning: No missing values were found in the dataset, so no imputation or dropping rows were required. The irrelevant ‘ID’ feature was dropped.

One-Hot Encoding: Categorical variables ('Warehouse_block', 'Mode_of_Shipment', 'Product_importance', 'Gender') were one-hot encoded. Since KNN, Decision Tree, Random Forest, and Gradient Boost models are not sensitive to multicollinearity, we don’t have drop first column of dummy variables for them. However, I used Linear kernel for SVM, which was sensitive to multicollinearity, so I should drop the first column of dummy variables.

Normalization: The first five numerical features were standardized using z-score standardization.

(d) Results from all the models in the form of a table



	Accuracy	AUC	F1_Score	Precision	Recall
KNN	0.644545	0.663524	0.652135	0.777306	0.561686
SVM	0.640909	0.631323	0.692846	0.703236	0.682759
Decision Tree	0.647727	0.634789	0.703406	0.702599	0.704215
Random Forest	0.660909	0.663276	0.694763	0.745391	0.650575
Gradient Boost	0.688182	0.715927	0.683287	0.859466	0.567050
Decision Tree (Hyperparameter Tuning)	0.685000	0.722548	0.662445	0.909091	0.521073
Gradient Boost (Hyperparameter Tuning)	0.691364	0.731597	0.664691	0.934722	0.515709

(e) Results from the hyperparameter search

I used Grid Search to do hyperparameter tuning.

Decision Tree:

```
max_depth_values = [3, 5, 10, None]
min_samples_split_values = [2, 5, 10]
```

```
Best Parameters for Decision Tree: {'max_depth': 5, 'min_samples_split': 2}
Best Accuracy for Decision Tree: 0.685
```

Accuracy is 0.685

AUC is 0.7225

F1 score is 0.6624

Precision is 0.9091

Recall is 0.5211

Gradient Boost:

```
n_estimators_values = [100, 200, 300]
learning_rate_values = [0.01, 0.1, 0.2]
```

```
Best Parameters for Gradient Boost: {'n_estimators': 300, 'learning_rate': 0.01}
Best Accuracy for Gradient Boost: 0.6914
```

Accuracy is 0.6914

AUC is 0.7316

F1 score is 0.6647

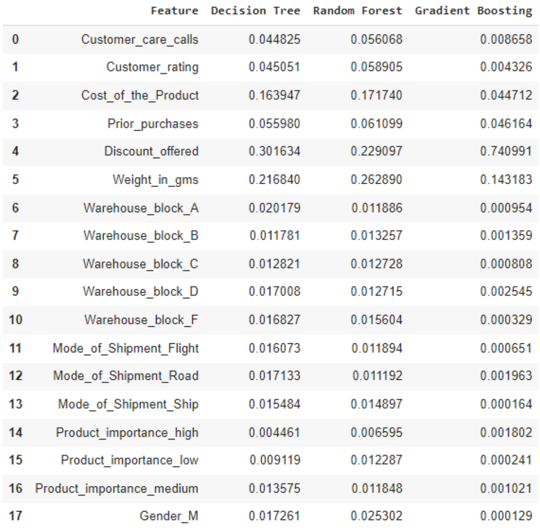
Precision is 0.9347

Recall is 0.5157

(f) Conclusions

Gradient Boosting: It showed the highest accuracy and AUC, indicating its effectiveness for this prediction task.

Feature Importance: Since KNN and SVM were not able to calculate feature importance, I just calculated the feature importance of Decision Tree, Random Forest, and Gradient Boost models. The 'Discount_offered' and 'Weight_in_gms' were significant predictors across multiple models, indicating their strong influence on shipment timeliness.



	Feature	Decision Tree	Random Forest	Gradient Boosting
0	Customer_care_calls	0.044825	0.056068	0.008658
1	Customer_rating	0.045051	0.058905	0.004326
2	Cost_of_the_Product	0.163947	0.171740	0.044712
3	Prior_purchases	0.055980	0.061099	0.046164
4	Discount_offered	0.301634	0.229097	0.740991
5	Weight_in_gms	0.216840	0.262890	0.143183
6	Warehouse_block_A	0.020179	0.011886	0.000954
7	Warehouse_block_B	0.011781	0.013257	0.001359
8	Warehouse_block_C	0.012821	0.012728	0.000808
9	Warehouse_block_D	0.017008	0.012715	0.002545
10	Warehouse_block_F	0.016827	0.015604	0.000329
11	Mode_of_Shipment_Flight	0.016073	0.011894	0.000651
12	Mode_of_Shipment_Road	0.017133	0.011192	0.001963
13	Mode_of_Shipment_Ship	0.015484	0.014897	0.000164
14	Product_importance_high	0.004461	0.006595	0.001802
15	Product_importance_low	0.009119	0.012287	0.000241
16	Product_importance_medium	0.013575	0.011848	0.001021
17	Gender_M	0.017261	0.025302	0.000129

Model Selection: The hyperparameter-tuned Gradient Boost model emerged as the most effective model for predicting on-time delivery in e-commerce with accuracy of 0.6914 and AUC of 0.7316. Therefore, I am able to answer my initial question with the models.

(g) Link to the GitHub repo