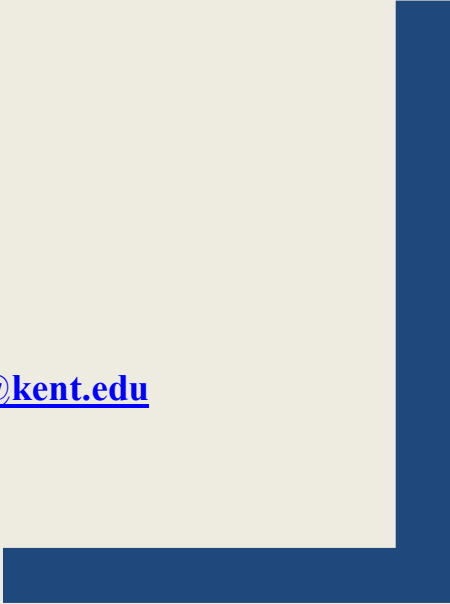




# **ON-TIME OR DELAYED SHIPMENT PREDICTION USING MACHINE LEARNING METHODS**

**By:**  
**Shefali Gupta – [sgupta22@kent.edu](mailto:sgupta22@kent.edu)**



## **Table of content**

<b>Abstract</b>	<b>2</b>
<b>1. Introduction</b>	<b>2</b>
<b>2. Relevant studies</b>	<b>3</b>
<b>3. Data Description</b>	<b>4</b>
<b>4. Methodology</b>	<b>5</b>
4.1. Data Preprocessing and Feature Selection	5
1.2. Methodology and Machine Learning Algorithm	11
<b>5. Results</b>	<b>14</b>
5.1. Selected Criteria: True Positive Rate (TPR), F1 Score, and Fowlkes-Mallows Index (FM)	23
5.2. A Comparative Visualization of Three Sets Based on IQR and Median Performance Metrics	24
5.3 Refining Model Configuration: The selection of Set 1	29
5.5 Comparative Analysis of Set 1, Set 2, and Set 3: Identifying Effective Models	37
Analysis:	37
<b>6. Distinctive Contributions</b>	<b>39</b>
<b>7. Conclusion</b>	<b>39</b>
<b>8. References</b>	<b>40</b>

## Abstract

This research explores the application of various machine-learning methods to predict the timely arrival of shipments. The prediction outcome is represented by a binary variable indicating whether the shipment arrived on time (0) or not (1). The employed techniques encompass decision tree, logistic regression, Gaussian Naive Bayes, efficient linear support vector machines (SVM), kernel Naive Bayes, logistic regression kernel, ensemble classifiers, and neural network classifiers. The study employs the design of experiments to assess the consistency of these machine learning models, with variations in test data sizes and a validation holdout of 10%, 15%, and 20%. The primary objective is to advance the field of shipment arrival prediction through the integration of diverse machine-learning approaches.

**Key Words:** Shipment Prediction, Supply Chain Management, Design of Experiments, Classification, Decision Trees, Coarse Tree, Gaussian Naive Bayes, RUSBoosted Trees, Neural Network

## 1. Introduction

The timely delivery of shipments is crucial in today's dynamic global marketplace. Ensuring prompt and reliable deliveries is vital for businesses across sectors, impacting customer satisfaction, operational efficiency, and economic growth. An illustrative study by Salari et al [1] endeavors to contribute to this discourse by delving into the realm of delivery time prediction for orders, with the overarching objective of enhancing customer satisfaction cost-effectively. Recently, Mariappan et al [2] underscored the critical significance of shipment prediction utilizing cutting-edge artificial intelligence and machine learning techniques amid the challenging backdrop of the Covid-19 pandemic. Our research aims to leverage various machine-learning techniques to develop robust models for shipment arrival prediction. By comparing existing models, incorporating real-time data, and considering domain-specific features, this study seeks to contribute novel insights and methodologies that can revolutionize supply chain operations, reduce costs, and enhance customer satisfaction, ultimately benefiting both the academic community and industry practitioners.

The successful implementation of advanced machine learning techniques in shipment arrival prediction can lead to significant economic benefits by reducing costly delays and optimizing resource allocation, resulting in improved cost-effectiveness and profitability for businesses and supply chain stakeholders. Moreover, enhancing welfare is achieved by minimizing delivery uncertainties, ensuring timely access to goods, and promoting customer satisfaction, thereby fostering a positive impact on consumers and society at large.

## 2. Relevant studies

The existing literature on supply chain management (SCM) has shown limited exploration of machine learning (ML) techniques for predictive purposes, specifically in the context of predicting late deliveries by suppliers. One of the key challenges highlighted in Steinberg, et al [3] study is the applicability of classification algorithms and the curse of dimensionality in low-volume-high-variety production settings within SCM. To address these challenges, the authors propose a novel regression-based prediction model that effectively predicts the severity of late deliveries. Fancello, et al [4] addressed a crucial objective of predicting timely arrivals in the context of ship deliveries. The study introduces two innovative algorithms: a dynamic learning predictive algorithm based on neural networks and an optimization algorithm for resource allocation. Salleh, et al [5] provide insights into the shipping industry and the factors that contribute to variations in transit time. The authors classify the significant factors affecting arrival punctuality into two main categories: port conditions (e.g., tidal window, weather conditions at the port, etc.) and vessel conditions (e.g., speed, reliability of crews).

In the domain of e-commerce logistics and fulfillment, where timely last-mile order delivery is of paramount importance, Kandula, et al [6] introduce a novel decision support framework designed to enhance delivery operations. The authors introduce a comprehensive framework that leverages machine learning models to predict the success of order deliveries within a delivery shift. Considering logistics and delivery operations, Duin, et al [7] address the utilization of historical delivery data and address intelligence to predict future delivery outcomes. By employing multiple linear regression techniques, the study identifies and forecasts improvement potential (rework) for various zip code areas. Failed delivery attempts remain a prevalent challenge in the e-commerce order fulfillment domain. To address this issue, Florio, et al [8] introduce the delivery problem (DP), which aims to minimize the expected number of unsuccessful deliveries by optimizing a set of routes. The COVID-19 pandemic has had a profound impact on supply chains and last-mile logistics, necessitating accurate prediction of delays to address time-related uncertainties. A study by Wani, et al [9] utilizes shipment data from an e-commerce organization. To improve delay prediction accuracy, an enhanced hybrid voting-based classification model is proposed, incorporating Trees and Ensemble techniques such as bagging and boosting, with consideration of shipping mode, scheduled shipment time, and order type. In last-mile distribution, logistics companies often rely on rough estimates of stop delivery times to plan their routes. However, inaccurate estimates can result in degraded service levels and failure to meet promised time windows. Hughes, et al [10] aim to explore the feasibility of machine learning techniques in predicting stop delivery times. Various machine learning approaches, including ensembles, are tested to predict the stop delivery time and classify whether the total stop delivery time will exceed a predefined threshold.

Supply chain plays a crucial role in the growth of the eCommerce industry, and there is increasing interest in supply chain data analytics and risk management. Lolla, et al [11] focuses on the application of predictive analytics in the eCommerce industry to predict the risks of late deliveries to online shopping customers. One of the other research articles aimed to mitigate the effects of variability in supply chains by improving the prediction of transit time for shipments, rather than eliminating it completely. The results obtained through Jonquais & Kreml's [12] machine

learning model demonstrate the validity of using this approach to predict the estimated time of arrival (ETA) for shipments. In the realm of transportation and planning, Tsolaki, et al [13] strongly emphasize the utilization of machine learning techniques for valuable predictions, such as determining the number of containers to be loaded on container vessels, and state predictions, such as predicting on-time and delayed events, across various variables. Moreover, Chun [14] introduces a model that tackles the challenges Amazon encounters in handling inbound shipments and enhancing inventory optimization. By analyzing historical error rates, the model uncovers the underlying variation in arrival signals to enhance the accuracy of estimated delivery dates. Lastly, in the context of supply chain management, shipment consolidation is commonly employed to reduce outbound shipment costs; however, it can lead to increased lead times. By Employing machine-learning techniques like linear regression and logistic regression, Alnahhal, et al [15] assess whether orders will be received in the next delivery week or not. To optimize delivery efficiency, forecasting is continuously evaluated after each shipment delivery, allowing for the possibility of delaying current arriving orders for specific customers until the following week or delivering to the customer immediately.

### 3. Data Description

The dataset used in this study was obtained from the supply chain unit and includes a total of 12 variables. The data used in this study is sourced from Kaggle [16], a reputable open-source platform that hosts a diverse collection of datasets contributed by the global data science community. One of these variables serves as the target, and the remaining 11 serve as features used to forecast the response variable. In this study, the response variable of interest is indicated as Reached. on.Time\_Y.N. It is a categorical variable with two possible responses: 0 or 1. A value of 0 in this dataset indicates that the specific shipment resulted in being delivered on time, whereas a value of 1 indicates that the shipment was not delivered on time. The dataset contains 11 feature variables that are used to forecast the target variable Reached. on.Time\_Y.N. These features collect useful information about various areas of the supply chain and are critical for prediction.

The dataset contains four categorical variables that provide qualitative information about various aspects of supply chain factors. Among these category traits are:

1. Warehouse block: This feature describes the warehouse block where the product is held and is labeled A, B, C, D, or F.
2. Mode of Shipment: The mode of shipping used for the delivery, which can be Flight, Ship, or Road.
3. Product importance: This attribute indicates the product's importance level, which might be low, medium, or high.
4. Gender: The gender of the consumer in the supply chain is labeled as F (Female) or M (Male).

The dataset has seven numerical features that provide quantitative insights. These numerical features are as follows:

1. Customer care calls: This feature represents the number of customer care calls made during the supply chain process.

2. Customer rating: The customer's rating of the product or service (1-5).
3. Cost of the product
4. Discount offered: The discount offered on the product in percentages.
5. Weight in gms: The weight of the product in grams.
6. Prior purchases
7. ID: This feature serves as a unique identifier for each record in the dataset.

ID	Warehouse_block	Mode_of_Shipment	Customer_care_calls	Customer_rating	Cost_of_the_Product	Prior_purchases	Product_importance	Gender	Discount_offered	Weight_in_gms	Reached.on.Time_Y.N
1	D	Flight	4	2	177	3	low	F	44	1233	1
2	F	Flight	4	5	216	2	low	M	59	3088	1
3	A	Flight	2	2	183	4	low	M	48	3374	1
4	B	Flight	3	3	176	4	medium	M	10	1177	1
5	C	Flight	2	2	184	3	medium	F	46	2484	1

Table 1: input dataset utilized in predictive models.

## 4. Methodology

For this study, two distinct software tools are employed: RStudio version 2023.06.1+524 (Build 492) and MATLAB version 2022. RStudio served as the primary tool for data cleaning and descriptive analysis, while MATLAB was utilized for predictive analysis, particularly leveraging machine learning models via the classification learner tool.

### 4.1. Data Preprocessing and Feature Selection

Data preprocessing plays a crucial role in the data analysis pipeline, transforming raw data into a clean and structured format for analysis and modeling. During data cleaning, missing values are identified and replaced to ensure data integrity, while outliers are carefully handled to avoid undue influence on the analysis. The dataset comprises 10,999 observations and 12 variables, with no null values requiring replacement. Outliers are detected using boxplot and quartile methods, revealing 3212 instances. However, considering domain knowledge, none are removed as they are deemed significant. In this analysis, normalization or scaling is omitted for two main reasons: 1) Normalization or scaling is not performed in this analysis to maintain the original scale of the variables, facilitating easy interpretation and communication of the results. Applying normalization or scaling could potentially alter the original meaning of the variables and complicate the understanding of the analysis. 2) Preservation of the original data distribution is crucial to avoid potential information loss or distortion.

For feature selection, multiple analyses are performed to select the most influential variables that contribute toward the target variable. The analyses include correlation tests, analysis of variance

(ANOVA), step-wise regression, chi-square test, principal component analysis (PCA), and random forest, all aimed at determining the most significant features. Initially, the dataset consists of 12 variables, one of which, "ID," is deemed irrelevant and subsequently removed. As a result, the dataset comprises 11 variables, which are then segregated into two distinct sets based on their type: categorical and numerical variables. The categorical set includes the variables warehouse block, mode of shipment, product importance, and gender. The numerical set, on the other hand, comprises customer care calls, customer rating, cost of the product, prior purchases, discount offered, and weight in grams.

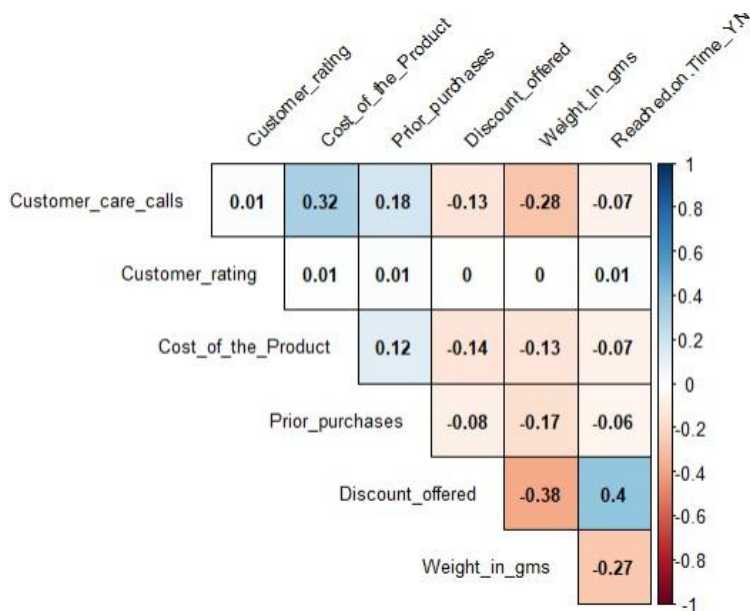


Figure 1: Correlation Plot

Correlation analysis (Figure 1) reveals valuable insights into the relationships among the variables. The results indicate a positive correlation between the variable "discount offered" and the response variable, "Reached\_on\_Time\_Y.N." Conversely, variables such as "cost of the product," "customer care calls," and "weight in grams" exhibit negative correlations with the response variable. This initial analysis provides a foundational understanding of the associations between the variables.

## Analysis of Variance Table

Response: Reached.on.Time\_Y.N

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Customer_care_calls	1	11.93	11.93	60.5200	7.936e-15	***
Customer_rating	1	0.51	0.51	2.6097	0.1062	
Cost_of_the_Product	1	7.98	7.98	40.5048	2.039e-10	***
Prior_purchases	1	4.35	4.35	22.0505	2.688e-06	***
Discount_offered	1	395.43	395.43	2006.5272	< 2.2e-16	***
Weight_in_gms	1	60.52	60.52	307.1162	< 2.2e-16	***
Residuals	10992	2166.20	0.20			

---

signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

**Figure 2: ANOVA Analysis**

Building upon the correlation analysis, ANOVA (Figure 2) is performed to assess the significance of variation between numerical variables. The variables "customer care calls," "cost of the product," "prior purchases," "discount offered," and "weight in gms" demonstrate a significant relationship with the response variable. In each case, the p-value is significantly less than 0.5, satisfying the commonly used significance levels.

```
Start: AIC=-17857.51
Reached.on.Time_Y.N ~ Customer_care_calls + Customer_rating +
  Cost_of_the_Product + Prior_purchases + Discount_offered +
  Weight_in_gms
```

	Df	Sum of Sq	RSS	AIC
<none>			2166.2	-17858
- Customer_rating	1	0.623	2166.8	-17856
- Cost_of_the_Product	1	2.018	2168.2	-17849
- Prior_purchases	1	5.122	2171.3	-17834
- Customer_care_calls	1	6.966	2173.2	-17824
- Weight_in_gms	1	60.524	2226.7	-17556
- Discount_offered	1	202.778	2369.0	-16875

```
Call:
lm(formula = Reached.on.Time_Y.N ~ Customer_care_calls + Customer_rating +
  Cost_of_the_Product + Prior_purchases + Discount_offered +
  Weight_in_gms, data = numerical_shipment)
```

```
Residuals:
    Min       1Q   Median       3Q      Max
-0.73576 -0.46137  0.02599  0.45664  0.77808
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.631e-01  3.278e-02  26.331  < 2e-16 ***
Customer_care_calls -2.474e-02  4.161e-03  -5.945  2.84e-09 ***
Customer_rating   5.324e-03  2.995e-03   1.778  0.07548 .
Cost_of_the_Product -3.014e-04  9.416e-05  -3.200  0.00138 **
Prior_purchases  -1.466e-02  2.876e-03  -5.098  3.49e-07 ***
Discount_offered  9.533e-03  2.972e-04  32.077  < 2e-16 ***
Weight_in_gms    -5.335e-05  3.044e-06 -17.525  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.4439 on 10992 degrees of freedom
Multiple R-squared:  0.1816,    Adjusted R-squared:  0.1812
F-statistic: 406.6 on 6 and 10992 DF,  p-value: < 2.2e-16
```

**Figure 3: Stepwise Regression Analysis**



Additionally, stepwise regression (Figure 3), a robust method for identifying influential variables, is employed. The purpose of this analysis is to systematically evaluate the impact of each variable on the likelihood of reaching on time. The stepwise regression analysis further confirms the significant relationship between the response variable and the following variables: "customer care calls," "cost of the product," "prior purchases," "discount offered," and "weight in gms." However, it is noteworthy that "customer rating" is eliminated from the model, indicating its lesser contribution to the overall analysis as its p-value is greater than 0.5.

```
[1] "Warehouse_block"
      Pearson's Chi-squared test
data:  table_data
X-squared = 1.0894, df = 4, p-value = 0.896

[1] "Mode_of_Shipment"
      Pearson's Chi-squared test
data:  table_data
X-squared = 0.74344, df = 2, p-value = 0.6895

[1] "Product_importance"
      Pearson's Chi-squared test
data:  table_data
X-squared = 12.211, df = 2, p-value = 0.00223

[1] "Gender"
      Pearson's Chi-squared test with Yates' continuity correction
data:  table_data
X-squared = 0.22308, df = 1, p-value = 0.6367

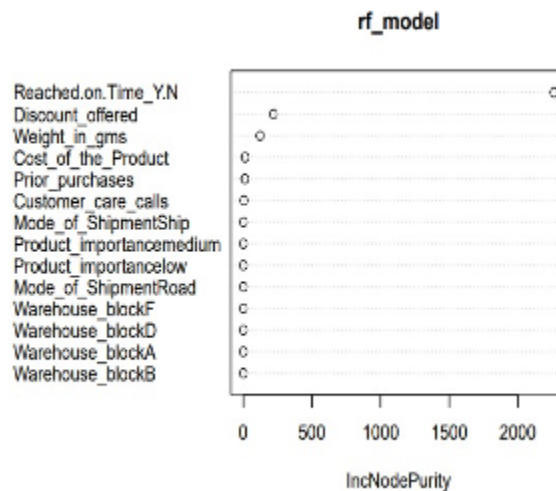
[1] "Reached.on.Time_Y.N"
      Pearson's Chi-squared test with Yates' continuity correction
data:  table_data
X-squared = 10995, df = 1, p-value < 2.2e-16
```

**Figure 4: Chi-square Test**

Upon finalizing the numerical variables, a chi-square test (Figure 4) assesses the significance of categorical variables in relation to the response variable. This statistical test enables the determination of the influence of each categorical variable on the outcome. The chi-square test results reveal that, among all the examined categorical variables, only the variable "product importance" demonstrates a significant association with the response variable, based on the obtained p-value. This insight provides valuable information for further analysis and strengthens the scientific rigor of the study. In summary, the final six features for the predictive analysis are customer care calls, cost of the product, prior purchases, discount offered, weight in gms, and product importance. These variables constitute "Set 1"

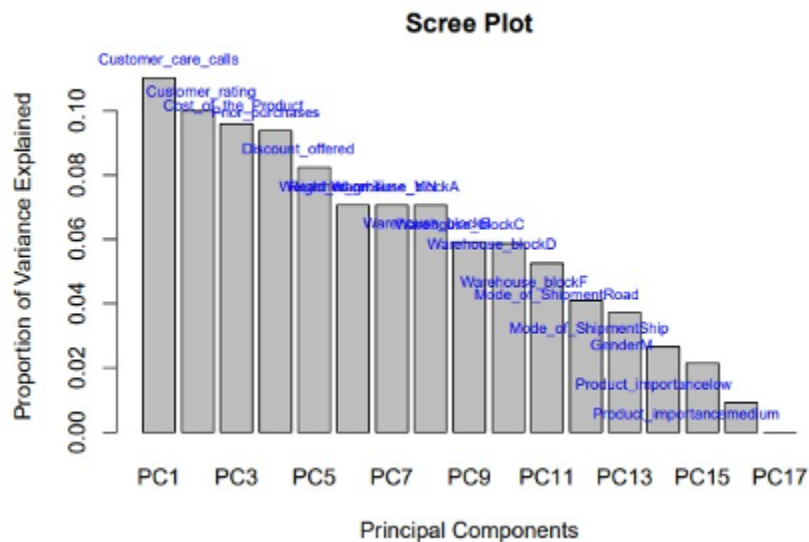
## PC4	Prior_purchases	2.840158
## PC2	Customer_rating	2.538250
## PC12	Warehouse_blockF	2.425027
## PC13	Mode_of_ShipmentRoad	2.264932
## PC17	Product_importancemedium	2.224835
## PC3	Cost_of_the_Product	2.219239
## PC1	Customer_care_calls	2.151928
## PC5	Discount_offered	2.081472
## PC14	Mode_of_ShipmentShip	2.069986
## PC8	Warehouse_blockA	2.063621
## PC11	Warehouse_blockD	2.011003
## PC6	Weight_in_gms	1.833654
## PC9	Warehouse_blockB	1.738084
## PC7	Reached.on.Time_Y.N	1.729600
## PC10	Warehouse_blockC	1.729141
## PC16	Product_importancelow	1.577414
## PC15	GenderM	1.529469

**Figure 5: PCA Importance Scores**



**Figure 6: Random Forest Plot**

In the context of the experiment's design (DOE), additional analysis is conducted to investigate if any other variables influence the target variable. PCA and random forest analyses (Figures 6 and 7) are performed using R, revealing intriguing findings. A higher PCA importance score highlights the increased significance of "customer rating" within the dataset. Based on the PCA importance score and domain knowledge, the variable "customer rating" is included in the list of selected variables. Consequently, the second set of variables comprises "customer care calls," "cost of the product," "prior purchases," "discount offered," "weight in grams," "product importance," and "customer rating," forming "Set 2"



**Figure 7: Scree Plot PCA**

##	IncNodePurity
## Customer_care_calls	4.2255907
## Cost_of_the_Product	13.0907002
## Prior_purchases	11.4688948
## Discount_offered	219.1238325
## Weight_in_gms	121.8827396
## Reached.on.Time_Y.N	2264.2645171
## Warehouse_blockA	0.4542272
## Warehouse_blockB	0.4354293
## Warehouse_blockD	0.4701353
## Warehouse_blockF	0.5211670
## Mode_of_ShipmentRoad	0.5524031
## Mode_of_ShipmentShip	0.8161598
## Product_importancelow	0.6144687
## Product_importancemedium	0.6398738

**Figure 8: Random Forest Feature Importance Score**

Subsequently, a separate analysis is performed using the random forest to determine the feature importance scores (Figure 9) that impact the target variable. The analysis reveals that only four features significantly affect the target variable: "cost of the product," "prior purchases," "discount offered," and "weight in grams." These variables constitute "Set 3" To conduct a comparative analysis, three distinct sets of variables are selected, each containing a different number of features. These sets are denoted as "Set 1" with 6 features, "Set 2" with 7 features, and "Set 3" with 4 features.

<b>Customer care calls</b>	<b>Cost of the Product</b>	<b>Prior purchases</b>	<b>Product Importance</b>	<b>Discount offered</b>	<b>Weight in gms</b>
4	177	3	low	44	1233
4	216	2	low	59	3088
2	183	4	low	48	3374
3	176	4	medium	10	1177

Table 2: Set 1 - Features

<b>Customer care calls</b>	<b>Cost of the Product</b>	<b>Prior purchases</b>	<b>Product importance</b>	<b>Discount offered</b>	<b>Weight in gms</b>	<b>Customer rating</b>
4	177	3	low	44	1233	2
4	216	2	low	59	3088	5
2	183	4	low	48	3374	2
3	176	4	medium	10	1177	3

Table 3: Set 2 - Features

<b>Cost of the Product</b>	<b>Discount offered</b>	<b>Prior purchases</b>	<b>Weight in gms</b>
177	44	3	1233
216	59	2	3088
183	48	4	3374
176	10	4	1177

Table 4: Set 3 - Features

The primary objective of this analysis is to assess how varying sets of features influence the performance of a given model or algorithm on a specific task or problem. By examining the outcomes obtained with these different feature sets, the impact of feature selection on the overall predictive power and effectiveness of the model can be understood.

## 1.2. Methodology and Machine Learning Algorithm

A selection of eight distinct machine learning algorithms with diverse characteristics have been made for this analysis including Decision tree, Logistic Regression, Gaussian Naive Bayes, Efficient Linear Support Vector Machines (SVM), Kernel Naive Bayes, Logistic regression kernel, Ensemble classifiers, neural network classifiers.

1. The Decision Tree algorithm can efficiently analyze various factors and divide the data into branches and excels in handling vast and intricate datasets without requiring a complex parametric framework [17].

2. Logistic regression calculates odds ratios considering multiple explanatory variables, resembling multiple linear regression but with a binomial response variable. This method reveals each variable's impact on the odds ratio of the observed event, helping avoid confounding effects by assessing all variables' associations simultaneously [18].
3. Gaussian Naive Bayes is a powerful and popular probabilistic classifier, well-suited for text classification and high-dimensional datasets. Its simplicity, efficient handling of continuous-valued features, and accurate class probability estimation make it widely used in various real-world applications [19].
4. Efficient Linear Support Vector Machines (SVM) algorithm signifies a substantial breakthrough in machine learning, providing a robust and scalable solution for large-scale classification endeavors. Its efficient optimization of the decision boundary allows for high-performance classification of intricate datasets, making it the preferred option across diverse real-world applications where speed and accuracy are crucial [20].
5. Kernel Naive Bayes is an extended version of the classical algorithm, using the kernel trick to handle non-linear data and improve performance in complex classification tasks. By mapping features to a higher-dimensional space, it efficiently captures intricate patterns, making it valuable for real-world applications with non-linear data distributions [21].
6. The Logistic Regression Kernel algorithm is an influential extension of standard logistic regression, allowing the handling of non-linearly separable data using kernel functions. Through feature space transformation into a higher-dimensional representation, it provides increased flexibility and accuracy, making it valuable for diverse classification tasks in real-world applications with complex data distributions [22].
7. Ensemble classifiers are powerful machine learning methods that combine multiple base classifiers to boost accuracy and handle complex problems, gaining popularity in various real-world applications [23].
8. 8. Neural Network Classifiers are a potent group of algorithms, that excel in complex classification tasks by learning hierarchical representations and adapting to non-linear relationships. Their remarkable success in diverse domains like image recognition, natural language processing, and financial forecasting makes them a promising area for further research and practical applications in machine learning [24].

Additionally, a diverse range of machine learning methods is employed, complemented by the implementation of a design of experiments (DOE). The DOE comprises 27 total runs, investigating the effects of varying test data partition sizes and the number of hold-out folds. To account for the three different sets of variables (Set 1, Set 2, Set 3), each set is subjected to 9 different test and hold-out folds, resulting in a total of 27 runs. This approach enables the assessment of how these hyperparameters impact the performance of machine learning models.

Run	Hold out	Test Data
Run 1	10	10
Run 2	10	15
Run 3	10	20
Run 4	15	10
Run 5	15	15
Run 6	15	20
Run 7	20	10
Run 8	20	15
Run 9	20	20

Table 5: Set 1 – 6 Features

Run	Hold out	Test Data
Run 19	10	10
Run 20	10	15
Run 21	10	20
Run 22	15	10
Run 23	15	15
Run 24	15	20
Run 25	20	10
Run 26	20	15
Run 27	20	20

Table 7: Set 3 – 4 Features

Run	Hold out	Test Data
Run 10	10	10
Run 11	10	15
Run 12	10	20
Run 13	15	10
Run 14	15	15
Run 15	15	20
Run 16	20	10
Run 17	20	15
Run 18	20	20

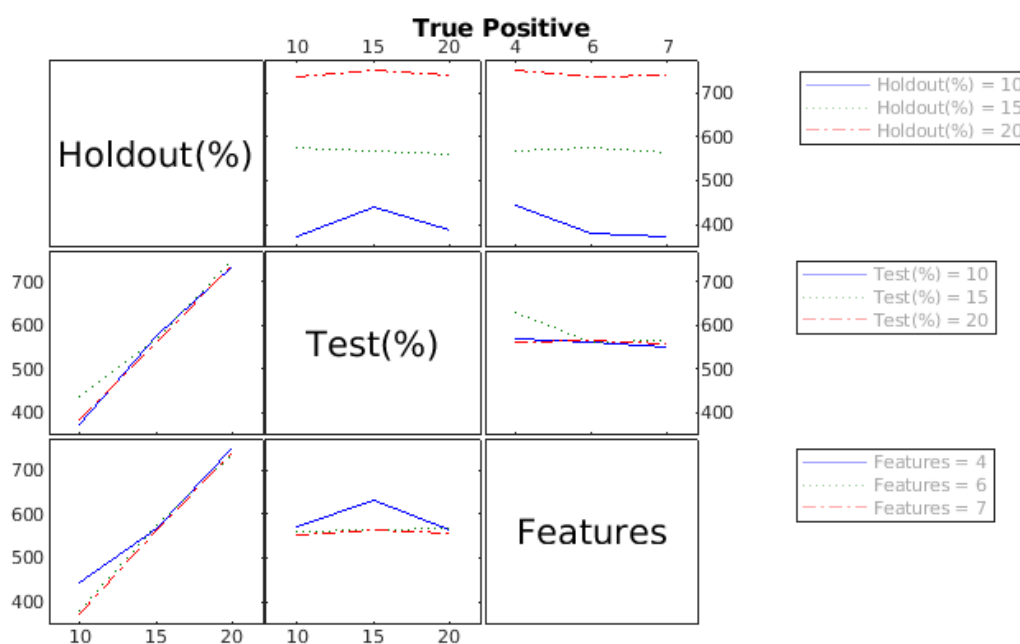
Table 6: Set 2 – 7 Features

## 5. Results

The interaction plot with three columns shows how the test performance of a model is influenced by various combinations of holdout percentages, test percentages, and features. It visually represents performance metrics like True Positive Rate (TPR) and F-score for each specific combination of these factors. Each point or line on the plot corresponds to a particular combination, providing a comprehensive view of the relationship between the mentioned factors and the model's performance.

### Interaction Plot

#### True Positive:



**Figure 8: True Positive Interaction Plot**

In the interaction plot using True Positive (TP) as the performance metric, significant interaction effects are indicated by parallel lines that intersect. Here are some key findings from the plot:

#### Holdout Percentages:

1. The upward intersection of three slanting lines indicates a significant interaction effect between features at different holdout percentages (10%, 15%, and 20%).
2. The Holdout (15%) and Test (15%) runs initially showed some changes, but after a certain point, their lines intersected with different holdout (10%, 20%) and test (10%, 20%) percentages.
3. Holdout and Test percentages (10%) with 4 features exhibited a higher rate of increase and intersected with other features (6, 7), suggesting a significant relationship.

### Test Percentages:

1. The three parallel lines correspond to different test data percentages (10%, 15%, and 20%).
2. The line forming a small pyramid indicates specific feature combinations that result in notably higher True Positive (TP) rates in certain test scenarios, such as when both Holdout and Test percentages are 10%, and the model uses 4 features.
3. The other two parallel lines show consistent model performance (TP) across the range of test percentages, irrespective of the specific feature combination.

### Features:

1. The three parallel lines correspond to different numbers of features (4, 6, and 7).
2. In one box, the three parallel lines suggest no significant interaction effect between the features at that specific configuration.
3. In the other box, the three intersecting lines form a straight line, indicating a significant interaction effect at this combination of feature levels.
4. At certain points, there is a slight decrease in performance when using 15% test data and 6 features, but it eventually intersects with the straight line formed by the other two lines, signifying a significant relationship between these factors.

### True Negative:

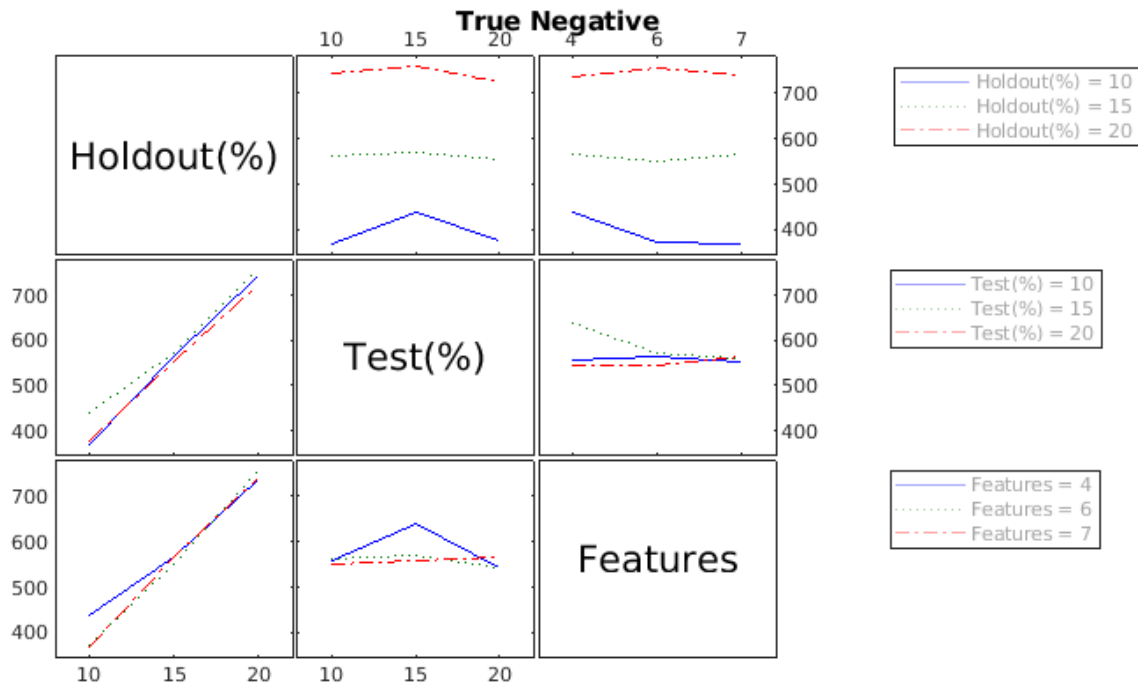
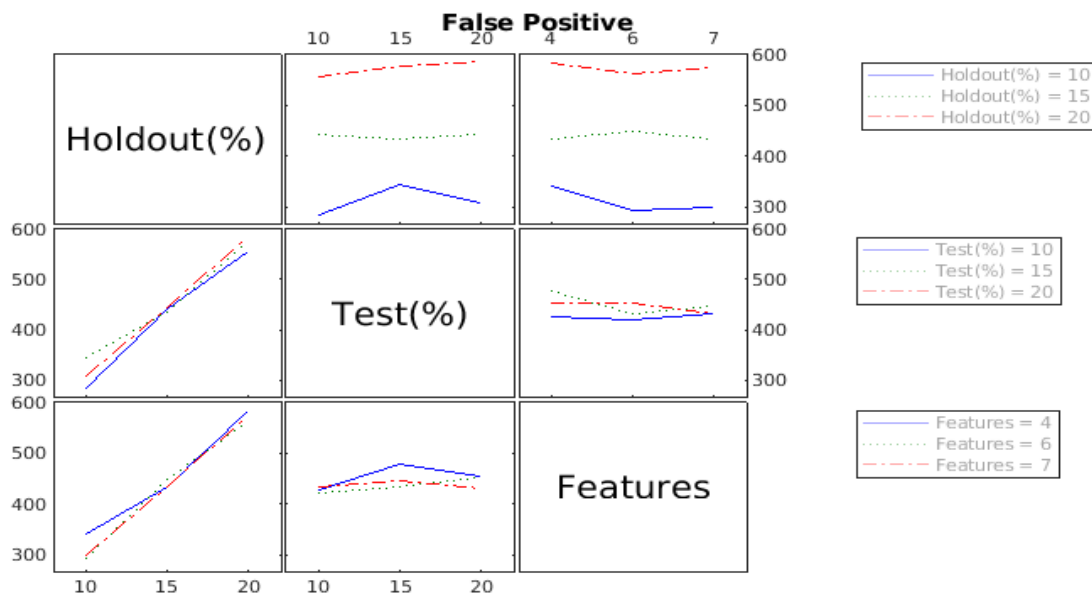


Figure 9: True Negative Interaction Plot



1. When using Holdout (15%) and Test (15%) percentages, there were initial changes in the performance. However, after a certain point, the line intersected with different holdout (10%, 20%) and test (10%, 20%) percentages, indicating significant relationships between these factors.
2. When both Holdout and Test percentages were set to 10%, and the model used 4 features, it exhibited a higher rate of increase in performance. Furthermore, this line intersected with the lines representing other feature combinations (6 and 7), suggesting significant relationships between these feature levels and model performance.
3. The line forming a small pyramid signifies specific feature combinations that yield notably higher TP rates in certain test scenarios for example (Holdout & Test=10%) and features = 4 and dropped below when compared to the other two lines
3. The two parallel lines represent consistent model performance (TP) across the entire range of test percentages, irrespective of the specific feature combination.
4. In the features column, one box shows three parallel lines, indicating no significant interaction effect between the features at this specific configuration.
5. In the other box, the three intersecting lines form a straight line, suggesting a significant interaction effect at this combination of feature levels.
6. When using a test percentage of 15% and 6 features, there is a slight decrease in performance initially. However, at a certain point, it intersects with the straight line formed by the other two lines, implying a significant relationship between these factors.

### False Positive:

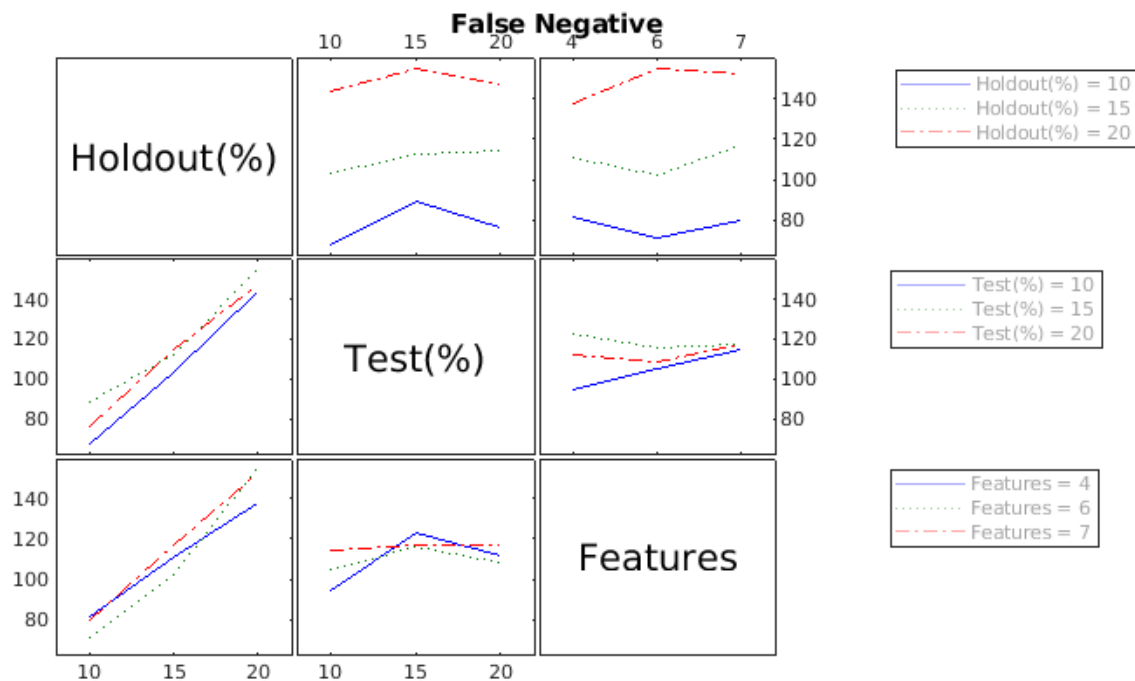


**Figure 10: False Positive Interaction Plot**

1. The false positive cases align with true positive and true negative cases, varying across different holdout and test percentages along with features.

2. There is an inverse relationship between the performance of cases with Holdout, Test, and Features set to (15, 15, 6) and (20, 20, 7). When there is an increase in performance for (15, 15, 6), there is a slight decrease in performance for (20, 20, 7), and vice versa.
3. The case with Holdout, Test, and Features set to (10, 10, 4) exhibits a relatively steady and consistent flow of performance.
4. Towards the end, the three lines intersect, indicating some significant relationships between the factors being analyzed.

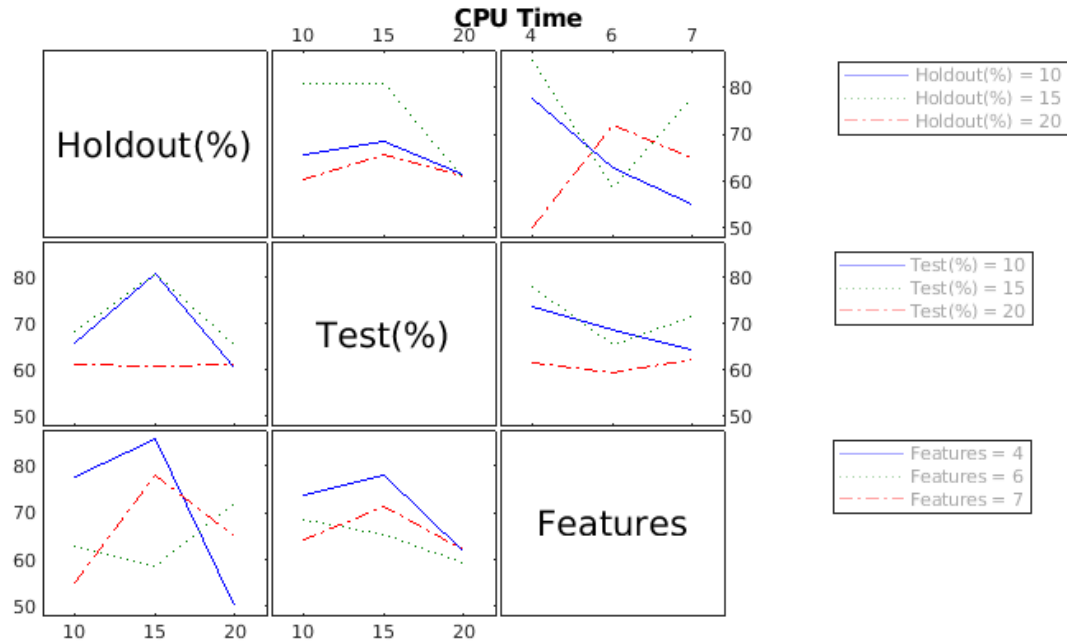
### False Negative:



**Figure 11: False Negative Interaction Plot**

1. Holdout, Test, and Features set to (10, 10, 4) consistently exhibit a certain effect compared to other combinations of holdout, test, and features.
2. The interaction plot indicates that the lines intersect at specific points, suggesting some significant relationships between the factors being studied. Additionally, the lines are somewhat parallel, indicating no main effects of significant relationships.

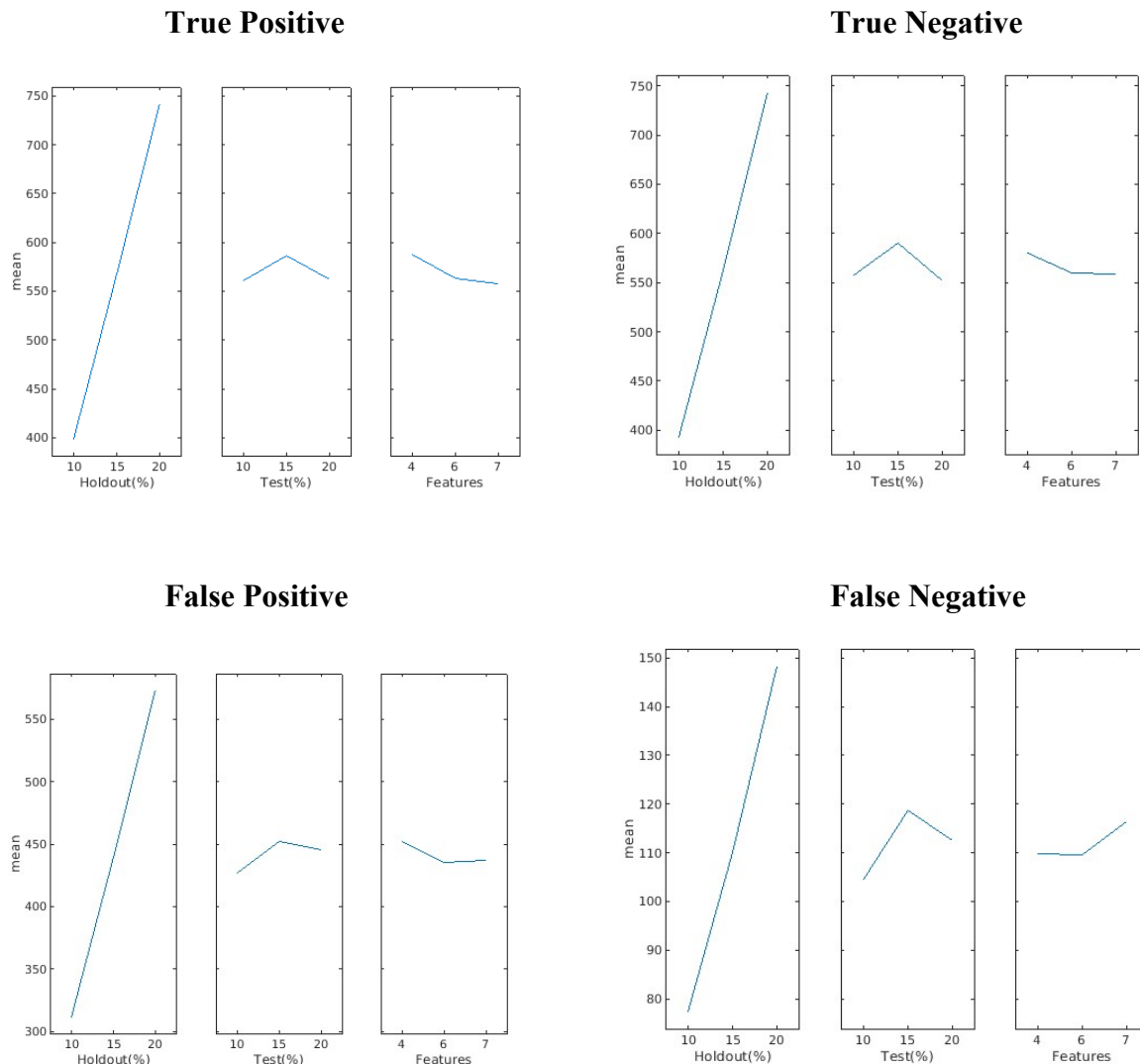
**Time:**



**Figure 12: Time Interaction Plot**

1. Initially, Holdout and Test (10, 10) combinations exhibit a significantly higher rate of increase, followed by a decline across all other holdout, test, and feature combinations.
2. An exception is observed in Case 1, where Holdout, Test, and Features (10, 10, 4) show a total decrease compared to Holdout, Test, and Features (15, 15, 6) and (20, 20, 7).
3. When Holdout is set to 15%, it decreases with an increase in test percentage. However, when the features are set to 6, there is an initial decrease followed by a steady increase after a certain point.
4. The interaction plot reveals that intersecting lines indicate a significant relationship between the factors, while parallel lines suggest no main effects of significant relationships.

## Main Effects:

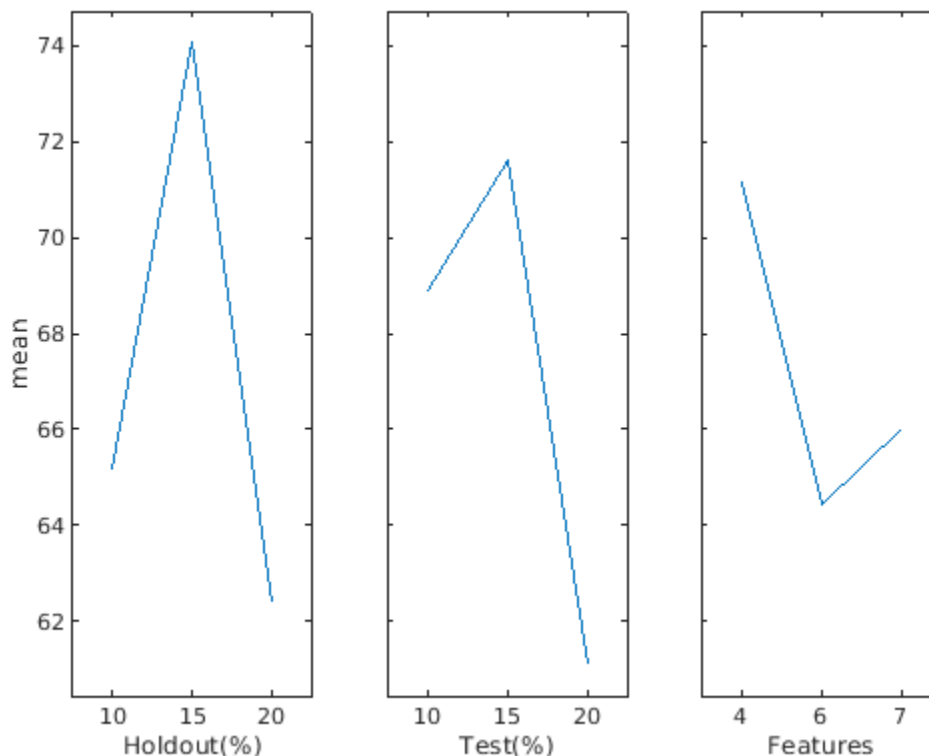


**Figure 13: Main Effects Plot**

1. In the main effect plot, the upward slanting line for holdout percentage indicates a positive main effect. This suggests that as the holdout percentage increases, the model's performance improves, leading to enhanced generalization, reduced risk of overfitting, and greater model stability.
2. The test percentage line forms a pyramid shape, with extremities (10% and 20%) resulting in underfitting and overfitting, leading to poorer model performance. The peak of the pyramid, at 15% test percentage, achieves the highest True Positive (TP) value, representing the best overall model performance.
3. The line for features with 4 exhibits a positive relationship, but it sharply declines at features=6, indicating a lack of significant impact. It then drops again with a negative relationship when features=7.

4. The observations made in cases 1, 2, and 3 are consistent across True Positive (TP), True Negative (TN), and False Positive (FP) cases, holding true for all corresponding holdout, test, and feature settings.
5. In the False Negative (FN) case, there is a flat line for features, indicating that features 4 and 6 have no significant impact on the response variable. However, the line shows a positive slope when features=7, suggesting a notable effect on the response variable.

### Main Effects Time:



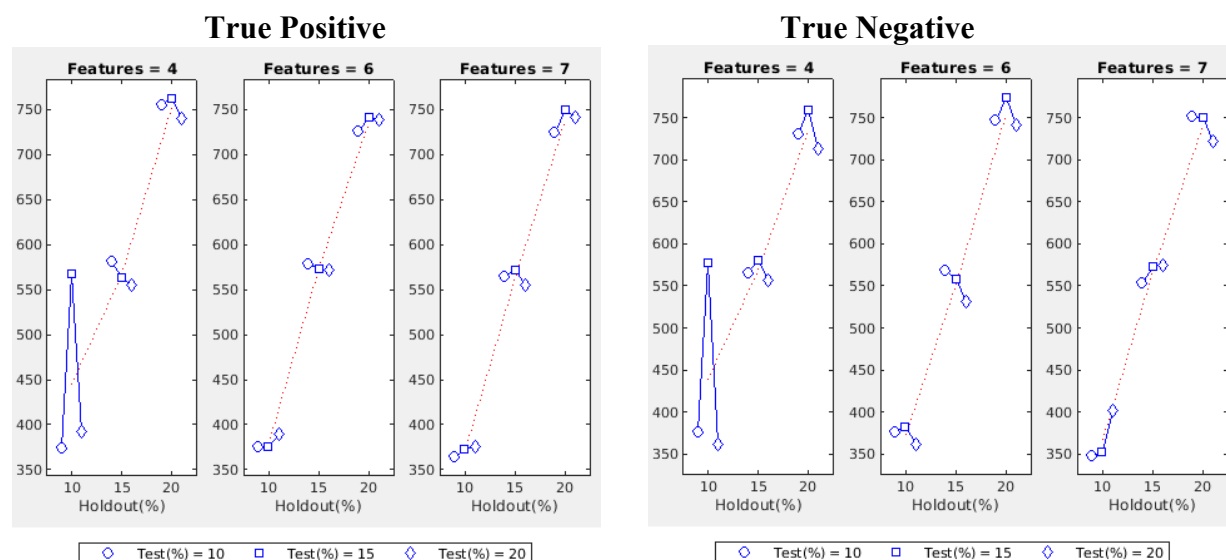
**Figure 14: Main Effects Time Plot**

1. In the main effects plot for holdout percentages (10%, 15%, and 20%), the response variable shows a non-linear relationship over time. The pyramid shape indicates varying impacts on model performance. The peak at 15% suggests an optimal holdout period, achieving the best performance. However, extremes at 10% and 20% may lead to overfitting and underfitting issues. Fine-tuning the holdout percentage can balance model complexity and generalization. It is essential to validate the significance of this pattern for reproducibility and its usefulness in time-dependent analyses.
2. In the main effects plot for test percentages (10%, 15%, and 20%), the response variable exhibits a non-linear relationship with the test percentage. The line rises quickly at 10%, indicating relatively good performance. The highest peak occurs at 15%, indicating the optimal test percentage for achieving the best model performance. However, the performance drops sharply

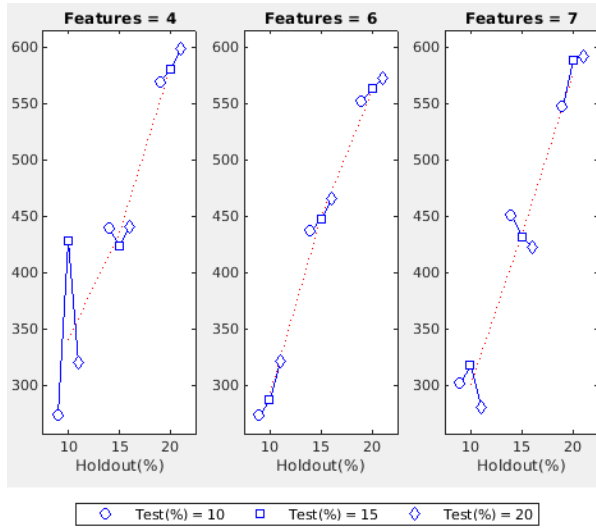
after 15%, suggesting potential overfitting or lack of generalization to new data. A test percentage of 15% strikes a balance between model accuracy and generalization. Further investigation is needed to understand the reasons behind the decline after 15% and ensure the reliability of this pattern.

3. In the main effects plot for features, the response variable demonstrates a non-linear relationship with the number of features. Performance is highest with 4 features, indicating that a smaller feature set leads to better model performance. As the number of features increases, performance declines, reaching its lowest point at 6 features. This suggests that too many features may introduce noise or overfit the model. Interestingly, performance improves slightly with 7 features but remains lower than that achieved with 4 features. This emphasizes the importance of feature selection and finding an optimal balance between model complexity and performance. Further investigation is recommended to understand the factors influencing model performance with different feature combinations.

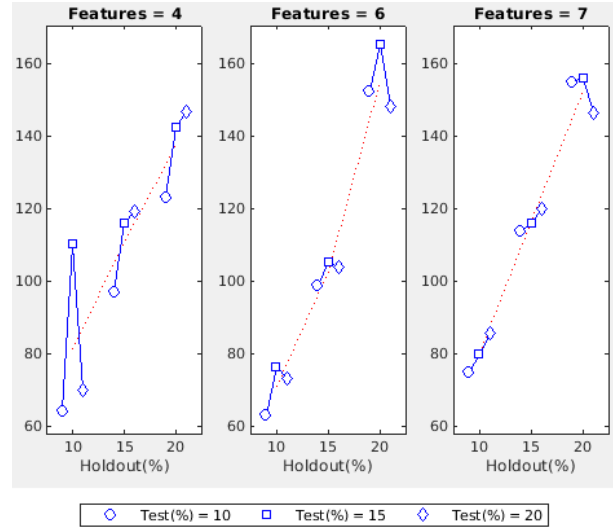
## Multivariate:



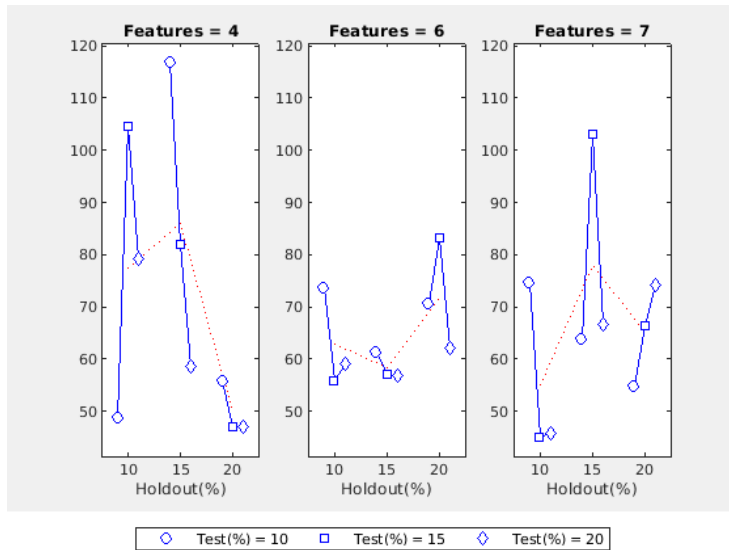
## False Positive



## False Negative



## Time:



**Figure 15: Multivariate Plots**

In the multi-variable chart with fixed features at 4 and 6, and a holdout percentage of 10% (10%, 15%, 20%), a consistent peak in the test percentage is observed at 15% across all cases. This indicates an optimal test percentage for achieving the highest model performance, demonstrating strong generalization capabilities and avoiding overfitting and underfitting issues. Moreover, there is a consistent upward trend for features=6, reinforcing the positive relationship and correlation between the variables. These findings are robust and reliable across various holdout percentages, making 15% an appropriate choice for real-world applications. The observed patterns highlight the

significance of selecting an appropriate test percentage to ensure optimal model performance and make informed decisions.

### 5.1. Selected Criteria: True Positive Rate (TPR), F1 Score, and Fowlkes-Mallows Index (FM)

This study assesses the performance of a predictive model by incorporating three pivotal evaluation criteria: True Positive Rate (TPR), F1 Score, and Fowlkes-Mallows Index (FM). These metrics hold substantial significance in model assessment, particularly in specialized domains or subject areas. TPR, akin to sensitivity or recall, gauges the model's proficiency in accurately identifying positive instances. This metric's relevance is particularly evident in vital applications such as medical diagnostics, fraud detection, and anomaly identification. The F1 Score, being the harmonic mean of precision and recall, serves as a valuable measure when confronting imbalanced datasets with limited positive instances. This amalgamation of precision and recall offers a balanced perspective of the model's performance, rendering it ideal for imbalanced classification challenges. Meanwhile, the FM Index facilitates the comparison of clustering or classification results by examining pairwise samples. Incorporating true positive, false positive, true negative, and false negative rates, this index elucidates the alignment between predicted and true clusters or classes, making it suitable for tasks like image segmentation and clustering evaluation.

The chosen metrics form a comprehensive evaluation strategy that accounts for the model's capability to accurately identify positive instances (TPR and F1 Score) and its clustering or classification performance (FM Index). The results consistently demonstrated parallel trends across diverse test conditions, reinforcing the model's stability and generalizability. This robustness provides researchers and practitioners with increased confidence in leveraging the model for applications pertaining to the study's domain.



5.2. A Comparative Visualization of Three Sets Based on IQR and Median Performance Metrics

Visualization of Median Values

Criterion 1 (TPR)

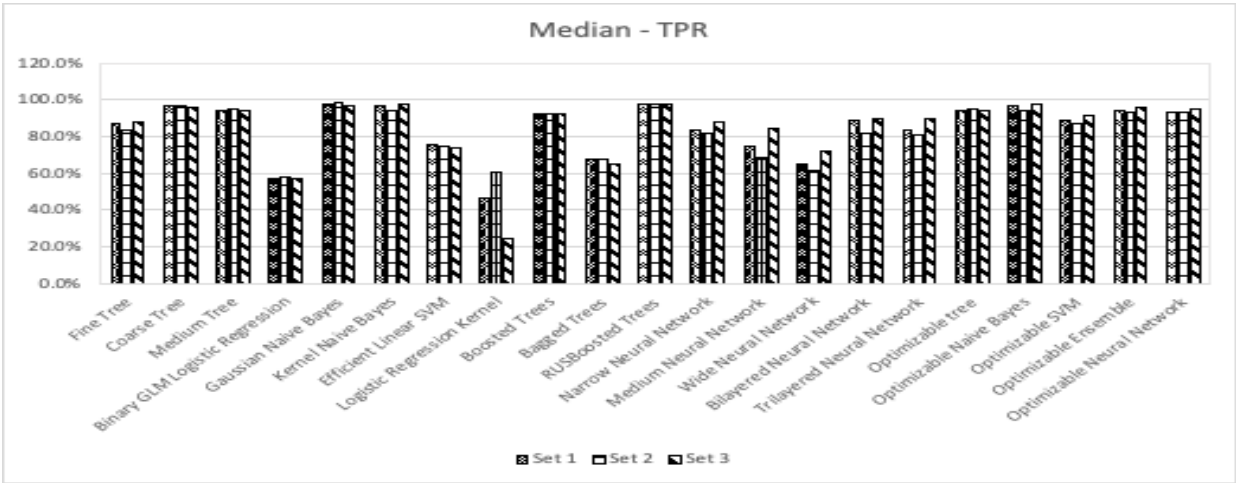


Figure 16: Median Values of TPR

The median values of TPR (True Positive Rate) for the different models across Set 1, Set 2, and Set 3 show their performance in correctly predicting positive instances. "Gaussian Naive Bayes" stands out with the highest median TPR across all three sets, achieving 97.9%, 98.2%, and 96.6% in Set 1, Set 2, and Set 3, respectively. It consistently exhibits excellent sensitivity in all scenarios. Other notable performers include "RUSBoosted Trees" with median TPR values of 97.7%, 97.3%, and 97.9%, and "Kernel Naive Bayes" with median TPR values of 96.4%, 94.0%, and 97.5% across Set 1, Set 2, and Set 3, respectively. "Coarse Tree" also demonstrates competitive median TPR scores with values of 96.4%, 96.8%, and 95.7%. However, certain models like "Logistic Regression Kernel" show lower median TPR values, particularly in Set 3 with only 24.6%, indicating a less effective ability to identify true positive cases.

### Criterion 2 (F1)

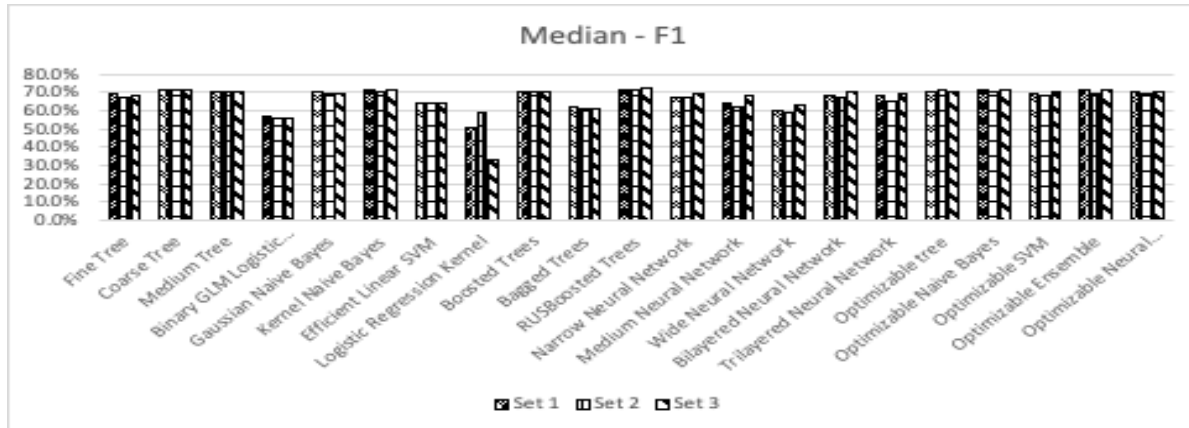


Figure 17: Median Values of F1

The median values of different models for Set 1, Set 2, and Set 3 provide insights into their overall performance in terms of accuracy. "Kernel Naive Bayes" consistently shows the highest median accuracy among the models, achieving 71.7%, 70.6%, and 71.6% accuracy in Set 1, Set 2, and Set 3, respectively. "RUSBoosted Trees" also performs well with median accuracy values of 71.8%, 71.4%, and 72.1% across the three sets. "Coarse Tree," "Boosted Trees," "Optimizable Ensemble," and "Fine Tree" demonstrate relatively high median accuracy values, ranging from 70.9% to 71.1% in Set 1, Set 2, and Set 3. However, certain models such as "Logistic Regression Kernel" exhibit lower median accuracy values, particularly in Set 3 with only 32.8%, indicating they may not be as effective in generalizing to new data samples. Overall, "Kernel Naive Bayes" stands out as the most accurate model, while other models offer competitive performance across the three sets.

### Criterion 3 (FM)

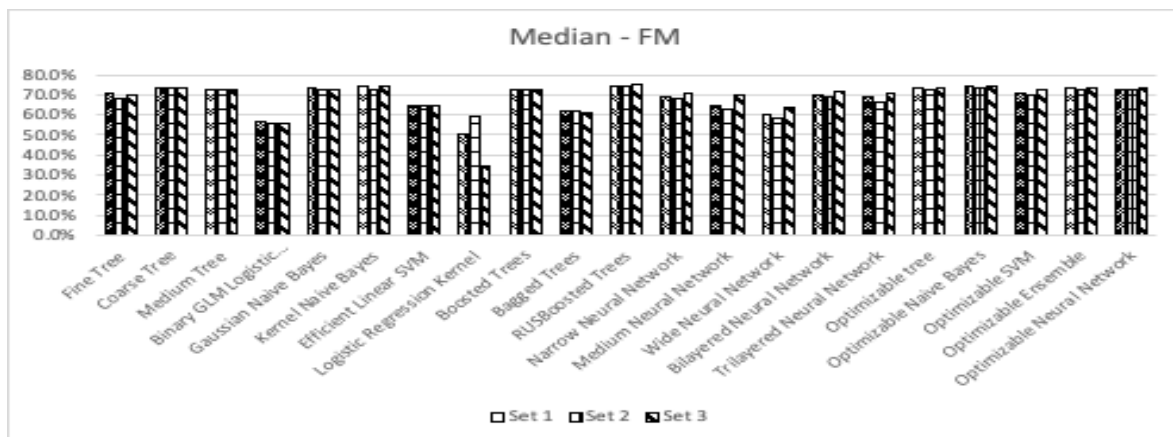


Figure 18: Median Values of FM

The median accuracy values of the various models across Set 1, Set 2, and Set 3 provide insights into their performance. "RUSBoosted Trees" consistently achieves the highest accuracy among the models, with values of 74.5%, 74.0%, and 74.8% in Set 1, Set 2, and Set 3, respectively. "Kernel Naive Bayes" and "Optimizable Naive Bayes" also display strong performance, maintaining median accuracy values of approximately 73% across all three sets. Additionally, "Coarse Tree," "Optimizable Ensemble," "Optimizable Tree," and "Medium Tree" exhibit relatively high median accuracy values, ranging from 73.5% to 73.8% in Set 1, Set 2, and Set 3. On the other hand, "Logistic Regression Kernel" and "Wide Neural Network" have lower median accuracy values, especially in Set 3, suggesting potential limitations in their generalization ability. In summary, "RUSBoosted Trees" emerges as the most accurate model across all three sets, closely followed by "Kernel Naive Bayes" and "Optimizable Naive Bayes."

Visualization of IQR Values

Criterion 1 (TPR)

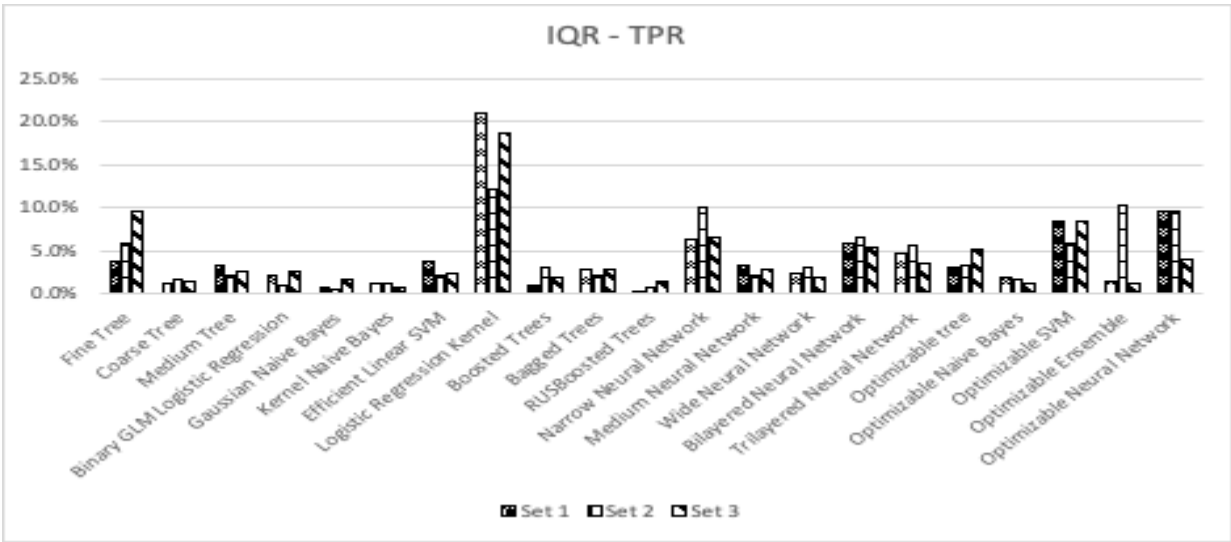
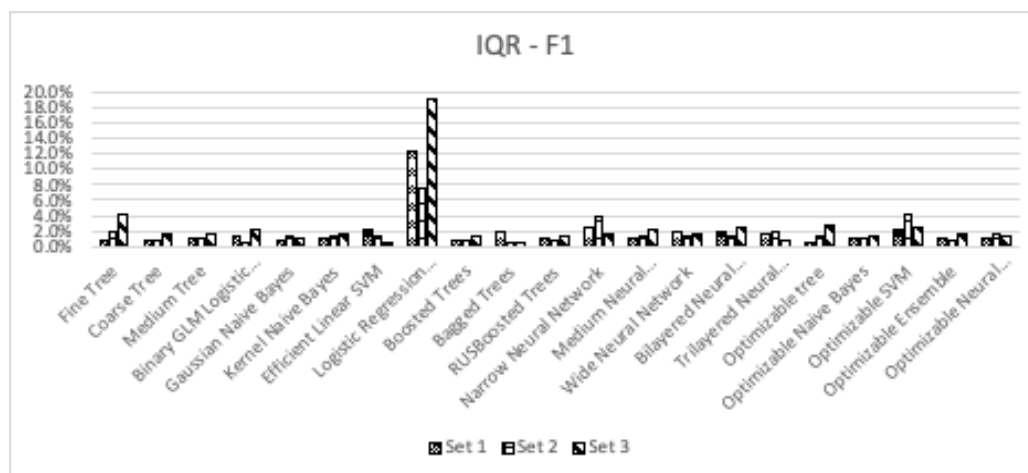


Figure 19: IQR Values of TPR

The interquartile range (IQR) values provide information about the spread or variability of the true positive rate (TPR) across the different models in Set 1, Set 2, and Set 3. Some models exhibit relatively low IQR values, indicating more consistent TPR performance. For instance, "RUSBoosted Trees" consistently demonstrates the lowest IQR values, with 0.3%, 0.9%, and 1.4% in Set 1, Set 2, and Set 3, respectively. Similarly, "Optimizable Naive Bayes" and "Trilayered Neural Network" also show low IQR values across all three sets. On the other hand, certain models exhibit higher IQR values, suggesting greater variability in their TPR results. Notably, "Logistic Regression Kernel" and "Optimizable Neural Network" show relatively large IQR values, particularly in Set 1 and Set 2. In summary, models with lower IQR values, such as "RUSBoosted Trees" and "Optimizable Naive Bayes," tend to display more consistent TPR performance, while

models with higher IQR values, like "Logistic Regression Kernel" and "Optimizable Neural Network," demonstrate more variable TPR results across the datasets.

### *Criterion 2 (F1)*



**Figure 20: IQR Values of F1**

The interquartile range (IQR) values reveal the variability in the F1 scores among the different models across Set 1, Set 2, and Set 3. Certain models exhibit relatively low IQR values, indicating consistent F1 performance. For example, "Boosted Trees" consistently demonstrates the lowest IQR values with 0.7%, 0.9%, and 1.3% in Set 1, Set 2, and Set 3, respectively. Similarly, "Coarse Tree," "Gaussian Naive Bayes," and "Trilayered Neural Network" also display relatively low IQR values across all three sets. Conversely, some models exhibit higher IQR values, implying more variability in their F1 scores. Notably, "Logistic Regression Kernel" and "Optimizable SVM" show relatively large IQR values, particularly in Set 1 and Set 3. In summary, models with lower IQR values, such as "Boosted Trees" and "Gaussian Naive Bayes," tend to display more consistent F1 performance, while models with higher IQR values, like "Logistic Regression Kernel" and "Optimizable SVM," demonstrate more variable F1 results across the datasets.

### Criterion 3 (FM)

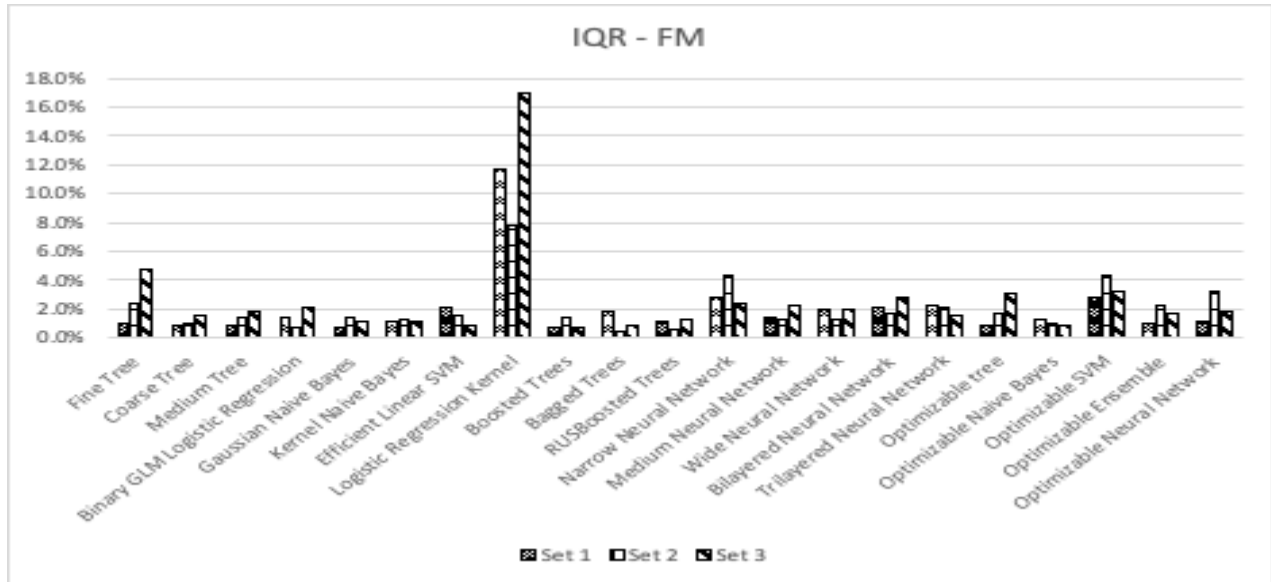


Figure 21: IQR Values of FM

The interquartile range (IQR) values illustrate the spread or dispersion of F1 scores among various models across Set 1, Set 2, and Set 3. Models with lower IQR values tend to have more consistent F1 scores, while those with higher IQR values show more variability in their performance. Among the models, "Boosted Trees" exhibit the lowest IQR values consistently across all three sets, indicating a stable and consistent F1 performance. On the other hand, "Logistic Regression Kernel" and "Optimizable SVM" show larger IQR values, especially in Set 3, suggesting more variability in their F1 scores across different datasets. Overall, models with smaller IQR values, such as "Boosted Trees," "Gaussian Naive Bayes," and "Trilayered Neural Network," are more reliable in terms of consistent F1 performance, while models with higher IQR values, like "Logistic Regression Kernel" and "Optimizable SVM," are more prone to varying F1 results across the datasets.

#### 5.2.1 A Comparative Analysis of 3 sets

##### **Set 1 (6 features): Highlights**

Set 1 exhibit a balanced performance across all three evaluation criteria (TPR, F1, FM), with median values of 94.3% TPR, 70.1% F1, and 72.8% FM. This indicates that the models in Set 1 perform well in terms of both recall and precision, striking a good trade-off between the two metrics.

While Set 3 has slightly higher median values, Set 1 still shows highly competitive performance. It outperforms Set 2 in terms of median TPR, F1, and FM, suggesting that it has a higher likelihood of predicting true positives, achieving a balanced F1 score, and maintaining an acceptable balance between precision and recall. Although Set 1 has slightly higher IQR values compared to Set 3, its IQR values are significantly lower than Set 2. This implies that the models in Set 1 are more consistent and less sensitive to dataset variations compared to Set 2. The interquartile range for Set 1 is relatively narrow for all three criteria (around 2-3%), indicating that the models within this set are stable and perform consistently across different data samples.

### ***Set 2 (Feature 7) and Set 3 (Feature 4) - Comparisons***

Set 2: While Set 2 may have competitive median values, its IQR values are considerably higher than both Set 1 and Set 3. This higher variability in performance suggests that the models in Set 2 may not always deliver consistent results, making them less reliable for practical applications.

Set 3: Set 3 has the highest median values for all three evaluation criteria (TPR, F1, FM), indicating superior overall performance. However, it's essential to note that the difference in median performance between Set 1 and Set 3 is not substantial. While Set 3 has a slight edge in terms of median values, Set 1 compensates with its lower IQR values, reflecting more stable and consistent results.

In conclusion, Set 1 offers balanced performance, competitive results, and better consistency compared to both Set 2 and Set 3. While Set 3 showcases slightly higher median values, Set 1's stability, and balanced performance make it a strong candidate for practical deployment.

## **5.3 Refining Model Configuration: The selection of Set 1**

Set 1 was carefully selected as the optimal model configuration based on its impressive performance and stability. Set 1 exhibited balanced performance across all three evaluation criteria, including a median True Positive Rate (TPR) of 94.3%, F1 Score of 70.1%, and Fowlkes-Mallows Index (FM) of 72.8%. These median values indicated that the models within Set 1 achieved a fine balance between recall and precision, making them suitable for various practical applications. Moreover, Set 1 demonstrated reasonable consistency, as reflected in its relatively narrow interquartile range (IQR) of approximately 2-3% for all three evaluation metrics. This consistency implied that the models consistently performed well across different data samples, enhancing their reliability and generalizability. The comprehensive analysis led to the confident selection of Set 1 as the most robust model configuration. Its balanced performance, reasonable consistency, and impressive stability make it a strong candidate for a wide range of real-world machine-learning applications.

High Median Values			
Criterion 1 (TPR)			
Model	Set 1	Set 2	Set 3
RUSBoosted Trees	97.7%	97.3%	97.9%
Criterion 2 (F1)			
Model	Set 1	Set 2	Set 3
RUSBoosted Trees	71.8%	71.4%	72.1%
Criterion 3 (FM)			
Model	Set 1	Set 2	Set 3
RUSBoosted Trees	74.5%	74.0%	74.8%
High IQR Values			
Criterion 1 (TPR)			
Model	Set 1	Set 2	Set 3
Logistic Regression Kernel	21.0%	12.1%	18.7%
Criterion 2 (F1)			
Model	Set 1	Set 2	Set 3
Logistic Regression Kernel	12.4%	7.7%	18.9%
Criterion 2 (F1)			
Model	Set 1	Set 2	Set 3
Logistic Regression Kernel	11.7%	7.7%	17.0%

Table 8 - Comparing high median and IQR values based on the 3 sets

Based on the numbers presented in table 8, when choosing between Set 1, Set 2, and Set 3, it becomes evident that Set 1 emerges as the preferable option. The reason for this preference lies in the following observations:

RUSBoosted Trees Model (Set 1):

- Set 1 consistently shows the highest median values for all three criteria (TPR, F1, and FM) for the RUSBoosted Trees model compared to Set 2 and Set 3.
- Set 1 has the highest performance in terms of TPR, F1, and FM for the RUSBoosted Trees model.

#### Logistic Regression Kernel Model (Set 1):

- Set 1 shows higher median values for all three criteria (TPR, F1, and FM) for the Logistic Regression Kernel model compared to Set 2 and Set 3.
- Although the Logistic Regression Kernel model's performance is lower than the RUSBoosted Trees model, within the options provided, Set 1 performs the best for this model.

Based on the numbers provided, Set 1 is the preferred choice over Set 2 and Set 3. It consistently yields higher median values for both the RUSBoosted Trees model and the Logistic Regression Kernel model, indicating better overall performance for the given criteria (TPR, F1, and FM). However, keep in mind that this conclusion is solely based on the numerical results provided in the table. In a real-world scenario, other factors such as data quality, model interpretability, and specific use-case requirements should also be taken into consideration when making a final decision.



Set 1											
TPR	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8	Test 9	MEDIA N	IQR
Fine Tree	85.65%	88.94%	92.64%	89.38%	90.12%	77.48%	87.37%	85.87%	83.54%	87.37%	3.73%
Coarse Tree	95.67%	95.13%	97.84%	95.72%	95.72%	97.04%	98.29%	96.47%	96.39%	96.39%	1.31%
Medium Tree	94.31%	92.70%	96.75%	92.92%	95.43%	96.00%	97.04%	90.07%	85.79%	94.31%	3.30%
Binary GLM Logistic Regression	58.09%	57.08%	59.09%	57.08%	55.31%	56.00%	55.86%	58.83%	57.95%	57.08%	2.09%
Gaussian Naive Bayes	97.95%	98.89%	97.40%	97.64%	98.38%	98.37%	98.52%	97.79%	97.29%	97.95%	0.74%
Kernel Naive Bayes	95.67%	97.12%	97.62%	96.31%	95.87%	96.89%	94.99%	96.36%	97.41%	96.36%	1.25%
Efficient Linear SVM	77.90%	77.43%	72.29%	77.14%	75.22%	73.19%	76.11%	73.84%	73.39%	75.22%	3.75%
Logistic Regression Kernel	59.45%	29.20%	66.02%	60.03%	44.99%	69.48%	46.19%	39.07%	37.54%	46.19%	20.96%
Boosted Trees	88.15%	88.05%	92.64%	92.33%	93.07%	94.22%	93.63%	92.05%	92.00%	92.33%	1.07%
Bagged Trees	69.02%	69.25%	67.75%	65.49%	69.91%	66.07%	66.33%	68.21%	64.60%	67.75%	2.95%
RUSBoosted Trees	97.49%	96.90%	98.48%	98.38%	96.02%	97.78%	97.61%	97.79%	97.75%	97.75%	0.30%
Narrow Neural Network	89.29%	77.88%	84.85%	88.79%	83.04%	83.41%	74.97%	83.33%	78.47%	83.33%	6.38%
Medium Neural Network	75.63%	68.36%	76.19%	75.66%	73.45%	74.52%	72.35%	72.19%	75.42%	74.52%	3.27%
Wide Neural Network	66.29%	63.72%	66.67%	66.08%	64.75%	64.59%	63.71%	60.15%	59.41%	64.59%	2.37%
Bilayered Neural Network	90.66%	88.72%	89.83%	84.37%	87.46%	96.74%	74.86%	79.58%	90.19%	88.72%	5.83%
Trilayered Neural Network	87.24%	82.96%	81.82%	86.46%	80.83%	83.85%	86.12%	79.91%	91.77%	83.85%	4.64%
Optimizable tree	94.53%	94.25%	83.12%	91.45%	95.43%	81.93%	93.63%	94.15%	94.81%	94.15%	3.09%
Optimizable Naive Bayes	96.58%	97.35%	92.64%	97.20%	95.28%	96.59%	94.65%	96.36%	97.41%	96.58%	1.92%

Optimizable SVM	86.79%	95.58%	73.16%	91.45%	97.05%	88.44%	76.11%	83.00%	90.64%	88.44%	8.44%
Optimizable Ensemble	93.62%	98.45%	94.16%	92.92%	93.22%	86.52%	94.99%	92.94%	94.48%	93.62%	1.54%
Optimizable Neural Network	98.63%	86.95%	87.23%	96.76%	93.81%	98.22%	92.26%	78.81%	93.46%	93.46%	9.53%

Table 9 - TPR percentages for 21 machine learning methods across 9 runs

## Set 1

F1	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8	Test 9	MEDIAN	IQR
Fine Tree	68.61%	68.95%	69.26%	69.98%	68.85%	65.33%	69.31%	68.37%	68.11%	68.85%	0.89%
Coarse Tree	71.79%	70.49%	69.59%	71.36%	70.62%	69.64%	71.14%	71.17%	71.25%	71.14%	0.76%
Medium Tree	72.00%	70.13%	70.01%	70.99%	70.71%	70.13%	71.74%	69.98%	68.68%	70.13%	0.98%
Binary GLM Logistic Regression	57.05%	56.39%	55.94%	56.33%	54.95%	54.19%	54.95%	58.03%	56.33%	56.33%	1.44%
Gaussian Naive Bayes	70.55%	70.84%	67.87%	70.16%	70.32%	68.70%	70.01%	69.87%	69.60%	70.01%	0.72%
Kernel Naive Bayes	72.54%	71.91%	69.87%	72.11%	71.35%	70.17%	70.94%	71.67%	71.85%	71.67%	0.97%
Efficient Linear SVM	65.83%	65.73%	61.28%	64.85%	63.99%	62.02%	65.33%	64.05%	63.20%	64.05%	2.13%
Logistic Regression Kernel	58.39%	38.65%	60.76%	57.16%	50.00%	62.37%	50.62%	45.97%	44.67%	50.62%	12.42%
Boosted Trees	69.79%	69.40%	69.48%	70.73%	70.23%	69.81%	71.26%	70.32%	70.53%	70.23%	0.73%
Bagged Trees	63.59%	62.79%	60.89%	61.54%	63.54%	60.23%	61.24%	62.49%	59.81%	61.54%	1.89%
RUSBoosted Trees	72.73%	71.74%	70.27%	72.26%	70.92%	70.33%	71.95%	71.92%	71.80%	71.80%	1.03%
Narrow Neural Network	70.31%	64.88%	67.41%	69.96%	66.75%	67.43%	65.74%	68.67%	66.16%	67.41%	2.51%
Medium Neural Network	67.00%	61.01%	64.88%	65.31%	63.68%	63.83%	64.31%	63.13%	64.67%	64.31%	1.20%
Wide Neural Network	62.85%	58.96%	60.87%	61.50%	59.89%	59.64%	60.51%	57.55%	57.69%	59.89%	1.91%
Bilayered Neural Network	70.82%	69.26%	67.26%	67.69%	68.12%	70.03%	65.54%	66.18%	69.23%	68.12%	2.00%
Trilayered Neural Network	69.32%	67.87%	66.20%	68.60%	66.71%	66.86%	67.92%	67.01%	70.17%	67.87%	1.73%
Optimizable tree	72.05%	70.65%	67.78%	70.70%	70.71%	66.83%	71.32%	71.11%	71.27%	70.71%	0.62%
Optimizable Naive Bayes	72.85%	72.01%	69.31%	72.50%	71.42%	70.11%	70.96%	71.67%	71.85%	71.67%	1.05%
Optimizable SVM	70.43%	71.94%	62.02%	70.06%	71.37%	68.31%	65.85%	68.12%	69.10%	69.10%	2.31%

Optimizable Ensemble	71.73%	72.12%	69.32%	71.23%	70.42%	68.67%	71.19%	71.11%	71.44%	71.19%	1.02%
Optimizable Neural Network	72.90%	69.93%	67.17%	71.81%	70.12%	69.21%	70.16%	65.44%	69.20%	69.93%	0.96%

Table 10 - F1 percentages for 21 machine learning methods across 9 runs

Set 1											
FM	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8	Test 9	MEDIAN	IQR
Fine Tree	70.01%	70.76%	71.57%	71.69%	70.85%	66.15%	70.84%	69.83%	69.30%	70.76%	1.02%
Coarse Tree	74.14%	72.98%	72.69%	73.79%	73.18%	72.60%	74.02%	73.75%	73.80%	73.75%	0.82%
Medium Tree	74.10%	72.30%	72.85%	73.05%	73.21%	72.82%	74.31%	71.79%	70.09%	72.85%	0.91%
Binary GLM Logistic Regression	57.06%	56.40%	56.02%	56.34%	54.95%	54.22%	54.96%	58.03%	56.35%	56.34%	1.44%
Gaussian Naive Bayes	73.48%	73.87%	71.23%	73.12%	73.37%	72.06%	73.14%	72.91%	72.60%	73.12%	0.77%
Kernel Naive Bayes	74.76%	74.46%	72.88%	74.50%	73.81%	73.00%	73.33%	74.15%	74.46%	74.15%	1.13%
Efficient Linear SVM	66.64%	66.49%	62.01%	65.69%	64.72%	62.76%	66.00%	64.62%	63.82%	64.72%	2.17%
Logistic Regression Kernel	58.40%	40.85%	60.95%	57.23%	50.31%	62.70%	50.86%	46.71%	45.50%	50.86%	11.69%
Boosted Trees	71.36%	71.01%	71.76%	72.75%	72.44%	72.28%	73.38%	72.37%	72.53%	72.37%	0.77%
Bagged Trees	63.79%	63.06%	61.21%	61.65%	63.80%	60.47%	61.42%	62.71%	59.98%	61.65%	1.85%
RUSBoosted Trees	75.19%	74.29%	73.34%	74.95%	73.47%	73.27%	74.57%	74.57%	74.47%	74.47%	1.10%
Narrow Neural Network	71.96%	65.81%	68.88%	71.59%	68.07%	68.70%	66.24%	69.76%	66.99%	68.70%	2.77%
Medium Neural Network	67.44%	61.36%	65.61%	65.93%	64.25%	64.50%	64.71%	63.63%	65.34%	64.71%	1.36%
Wide Neural Network	62.94%	59.12%	61.10%	61.64%	60.06%	59.82%	60.58%	57.60%	57.71%	60.06%	1.98%
Bilayered Neural Network	72.58%	70.99%	69.49%	69.05%	69.85%	72.86%	66.05%	67.14%	71.18%	69.85%	2.13%
Trilayered Neural Network	70.83%	69.02%	67.44%	70.11%	67.75%	68.28%	69.49%	67.90%	72.20%	69.02%	2.21%
Optimizable tree	74.18%	72.97%	68.97%	72.59%	73.21%	67.99%	73.43%	73.34%	73.58%	73.21%	0.84%
Optimizable Naive Bayes	75.16%	74.58%	71.62%	74.96%	73.77%	72.90%	73.29%	74.15%	74.46%	74.15%	1.29%
Optimizable SVM	71.71%	74.25%	62.75%	72.06%	74.01%	70.15%	66.45%	69.24%	71.14%	71.14%	2.82%
Optimizable Ensemble	73.77%	74.85%	71.87%	73.25%	72.62%	70.18%	73.53%	73.16%	73.66%	73.25%	1.04%

Optimizable Neural Network	75.51%	71.31%	69.02%	74.32%	72.47%	72.44%	72.26%	66.41%	71.66%	72.26%	1.16%
----------------------------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	-------

Table 11 - FM percentages for 21 machine learning methods across 9 runs

	TPR		FPR		PPV		NPV		Accuracy		Time (Seconds)	
Models	MEDIAN	IQR	MEDIAN	IQR	MEDIAN	IQR	MEDIAN	IQR	MEDIAN	IQR	MEDIAN	IQR
Fine Tree	87.4%	3.7%	44.3%	3.3%	56.8%	1.1%	86.8%	3.2%	67.9%	1.4%	3.34	1.54
Coarse Tree	96.4%	1.3%	50.5%	2.7%	56.0%	0.8%	95.3%	1.1%	67.9%	0.8%	3.16	3.44
Medium Tree	94.3%	3.3%	48.4%	4.6%	56.9%	1.1%	93.4%	3.3%	68.5%	1.0%	2.38	5.86
Binary GLM Logistic Regression	57.1%	2.1%	31.0%	1.5%	54.8%	1.7%	70.4%	1.0%	63.8%	1.7%	2.36	5.31
Gaussian Naive Bayes	97.9%	0.7%	55.6%	1.7%	54.4%	0.6%	97.1%	1.0%	66.0%	1.3%	1.85	3.73
Kernel Naive Bayes	96.4%	1.3%	49.3%	0.7%	56.9%	0.5%	95.5%	1.5%	69.2%	0.8%	4.61	2.06
Efficient Linear SVM	75.2%	3.7%	39.8%	1.8%	55.9%	1.5%	78.1%	2.3%	66.3%	1.9%	1.84	4.26
Logistic Regression Kernel	46.2%	21.0%	34.9%	7.4%	57.1%	0.8%	76.5%	5.9%	66.7%	1.1%	6.25	2.99
Boosted Trees	92.3%	1.1%	24.6%	12.8%	56.3%	0.7%	67.5%	6.2%	63.8%	2.2%	4.75	3.19
Bagged Trees	67.7%	2.9%	46.8%	2.2%	57.2%	0.9%	90.9%	0.9%	68.7%	1.0%	3.80	3.29
RUSBoosted Trees	97.7%	0.3%	34.0%	0.9%	57.4%	2.4%	74.5%	1.7%	66.9%	2.0%	7.54	5.54
Narrow Neural Network	83.3%	6.4%	50.2%	0.6%	56.9%	0.7%	96.9%	0.3%	69.2%	1.0%	4.77	4.73
Medium Neural Network	74.5%	3.3%	43.3%	3.7%	57.2%	2.1%	83.5%	3.6%	67.6%	2.4%	10.77	2.72
Wide Neural Network	64.6%	2.4%	38.3%	1.5%	56.5%	1.4%	77.7%	1.5%	66.7%	1.6%	14.47	1.78
Bilayered Neural Network	88.7%	5.8%	33.1%	3.5%	56.0%	2.1%	73.3%	1.5%	65.1%	1.7%	25.58	3.86
Trilayered Neural Network	83.9%	4.6%	45.5%	3.7%	56.5%	1.0%	87.4%	4.6%	67.6%	1.2%	13.40	1.90
Optimizable tree	94.2%	3.1%	43.9%	3.9%	56.8%	1.4%	83.6%	4.2%	68.0%	1.2%	14.97	6.72
Optimizable Naive Bayes	96.6%	1.9%	46.7%	2.8%	57.1%	1.1%	92.9%	3.2%	69.1%	1.4%	11.61	2.51
Optimizable SVM	88.4%	8.4%	49.1%	1.6%	57.1%	0.4%	95.4%	2.3%	69.3%	0.8%	32.54	8.02
Optimizable Ensemble	93.6%	1.5%	47.0%	6.6%	56.8%	1.9%	87.0%	6.5%	68.5%	1.3%	769.76	180.51
Optimizable Neural Network	93.5%	9.5%	47.3%	2.8%	56.9%	0.7%	92.3%	1.5%	69.3%	1.1%	38.30	7.13

Table 12 - Medians and IQRs for Set 1 performance criteria.

## **Set1 Analysis based on IQR and Median:**

### **True Positive Rate (TPR) Analysis:**

1. The model with the highest TPR is Gaussian Naive Bayes with an average of 0.976, followed closely by RUSBoosted Trees with an average of 0.975. Both models have high TPR and consistent performance.
2. The model with the lowest TPR is the Logistic Regression Kernel with an average of 0.391, indicating it has the lowest ability to correctly predict timely shipments.

### **F1 Score Analysis:**

1. RUSBoosted Trees has the highest F1 score with an average of 0.709, followed by Gaussian Naive Bayes with an average of 0.696. Both models achieve a good balance between precision and recall.
2. The model with the lowest F1 score is the Logistic Regression Kernel with an average of 0.460, indicating it performs poorly in terms of overall balanced performance.

### **Fowlkes-Mallows (FM) Score Analysis:**

1. RUSBoosted Trees achieves the highest FM score with an average of 0.735, indicating its good performance in terms of precision and recall.
2. Gaussian Naive Bayes also performs well with an FM score average of 0.726.
3. The model with the lowest FM score is the Logistic Regression Kernel with an average of 0.467, indicating its poor performance in terms of overall balanced performance.

Based on the comprehensive analysis of Set 1, considering IQR, Median, and Average:

1. Gaussian Naive Bayes consistently performs well across all three evaluation criteria (TPR, F1, and FM score). It has a high TPR, good F1 score, and one of the highest FM scores, making it a strong performer overall.
2. RUSBoosted Trees also stands out as a top-performing model with high TPR, the highest F1 score, and the highest FM score. It achieves good overall balanced performance.
3. Coarse Tree demonstrates high TPR and competitive FM score, making it another good choice for predicting timely shipment arrivals.
4. Logistic Regression Kernel consistently underperforms compared to other models across all three evaluation criteria. It may not be the best choice for this prediction task.
5. Other models like Boosted Trees, Efficient Linear SVM, and Narrow Neural Network also show decent performance based on the F1 and FM scores.

## **5.5 Comparative Analysis of Set 1, Set 2, and Set 3: Identifying Effective Models Analysis:**

### **Model Performance Variation:**

The models' performance varies across different sets (Set 1, Set 2, and Set 3) and evaluation criteria (Criterion 1, Criterion 2, and Criterion 3).

Some models consistently perform well across all sets and criteria, while others show more variability in their performance.

### **Top Performing Models:**

The "Coarse Tree" model consistently achieves high median values across all three sets and criteria. It demonstrates robust performance and low variability (low IQR), making it a reliable choice. The "Gaussian Naive Bayes" model also performs remarkably well in all sets and criteria, showing consistent and accurate results.

### **Poor Performing Model:**

The "Logistic Regression Kernel" model consistently shows low median values across all sets and criteria. Moreover, it exhibits high variability (high IQR), indicating inconsistent and unsatisfactory performance.

### **Effectiveness of Ensemble Techniques:**

Ensemble techniques, such as "Boosted Trees," "Bagged Trees," and "RUSBoosted Trees," generally perform well and yield more stable results compared to individual models. "RUSBoosted Trees" stands out as it achieves the highest median value for Criterion 3 (FM) and also has relatively low variability.

### **Effect of Neural Network Architecture:**

The performance of neural networks varies based on their architecture and size. Models with more layers, such as "Bilayered Neural Networks" and "Trilayered Neural Networks," tend to perform better than models with a single layer, such as "Narrow Neural Networks."

### **Overall Conclusions of Results:**

#### **Best Model Choice:**

Among the models tested, the "Coarse Tree" and "Gaussian Naive Bayes" models stand out as top performers. Researchers and practitioners should consider these models as primary choices for classification tasks.

#### **Ensemble Methods:**

Ensemble techniques, particularly "RUSBoosted Trees," have the potential to improve performance and robustness. These methods could be advantageous when dealing with imbalanced datasets or when higher FM scores are desired.

#### **Neural Network Selection:**

When choosing neural network models, it is crucial to consider the architecture. Deeper networks, such as "Bilayered Neural Networks" and "Trilayered Neural Networks," tend to perform better, while a "Narrow Neural Network" may not be the most effective choice.

## 6. Distinctive Contributions

This research aims to make significant contributions to the field of shipment arrival prediction by utilizing a diverse set of machine-learning techniques. While previous studies have focused on predicting the estimated time of arrival for shipments using various factors, our unique approach aims to predict an outcome of whether the shipment arrived on time or not (0 or 1) using several classification algorithms. This model will have a profound impact on the e-commerce sector, specifically in terms of enhancing customer satisfaction. This research differentiates itself by evaluating and comparing various algorithms and methodologies. The goal is to identify this specific domain's most accurate and robust predictive models.

Our research employs various variables, such as product importance, cost of the product, discounts offered, grams in weight, customer care calls, and prior purchases, to develop a classification model that aims to determine whether shipments will arrive on time. Through an extensive literature review, there was little to no evidence of similar classification models in this context. By addressing the existing gaps in the literature and introducing novel insights and methodologies, our research not only enriches the academic community but also holds substantial implications for industry practitioners, logistics providers, and businesses reliant on efficient shipment operations. The successful implementation of advanced machine learning techniques is expected to yield substantial societal and economic benefits, enhancing welfare and operational efficiency in the increasingly interconnected global marketplace.

## 7. Conclusion

To summarize, after conducting extensive experiments and analyses, this study found that the "Gaussian Naive Bayes" model consistently outperformed other models across all evaluation criteria and datasets. It demonstrated a high TPR, good F1 score, and one of the highest FM scores, making it a robust and reliable choice for predicting a shipment's timely arrival or late arrival. Furthermore, ensemble techniques, particularly "RUSBoosted Trees," also showed promising results and enhanced model performance, especially when dealing with imbalanced datasets. These methods are recommended for improving the model's robustness in real-world applications. Neural network models were also explored, and it was observed that deeper architectures, such as "Bilayered Neural Networks" and "Trilayered Neural Networks," outperformed single-layer models like "Narrow Neural Networks." Researchers and practitioners should consider these deeper architectures for similar classification tasks. This research reveals that the "Gaussian Naive Bayes" model is the most effective and reliable choice for predicting a shipment's timely arrival or late arrival. It offers accurate and consistent performance across different evaluation criteria and datasets. These findings contribute to the optimization of logistics and supply chain management operations by providing a powerful tool for predicting shipment status and ensuring timely deliveries.



## 8. References

1. Salari, N., Liu, S., & Shen. (2022). Real-Time Delivery Forecasting and Promising in Online-Retailing: When will your package arrive? *InformaPubsOnline* <https://pubsonline.informs.org/doi/epdf/10.1287/msom.2022.1081>
2. Mariappan, M. B., Devi, K., Venkataraman, y., Lim, K.M., & Theivendren, P. (2023). Using AI and ML to Predict Shipment Times of Therapeutic, Diagnostic, and Vaccines in E-pharmacy Supply Chains During Covid-19 Pandemic. *The International Journal of Logistics Management* <https://www.emerald.com/insight/content/doi/10.1108/IJLM-05-2021-0300/full/html>
3. Steinberg, F., Burggraf, P., Wagner J., Heinbach, B., Sabmannshausen, T., & Brintrup, A. (2023). A novel machine learning model for predicting late supplier deliveries of low-volume-high-variety products with application in a German machinery industry. *Supply Chain Analytics*, 1, <https://www.sciencedirect.com/science/article/pii/S294986352300002X>
4. Fancello, G., Pani, C., Pisano, M., Serra, P., Zuddas, P., & Fadda, P. (2011). Prediction of arrival times and human resources allocation for container terminal. *Maritime Economics & Logistics*, 13(2), 142–173. <https://doi.org/10.1057/mel.2011.3>
5. Salleh, N. H. M., Riahi, R., Yang, Z., & Wang, J. (2017). Predicting a Containership's Arrival Punctuality in Liner Operations by Using a Fuzzy Rule-Based Bayesian Network (FRBBN). *The Asian Journal of Shipping and Logistics*, 33(2), 95–104. <https://doi.org/10.1016/j.ajsl.2017.06.007>
6. Kandula, S., Krishnamoorthy S., & Roy, D. (2021). A prescriptive analytics framework for efficient E-commerce order delivery. *Decision Support Systems*, 147, <https://doi.org/10.1016/j.dss.2021.113584>
7. Duin, V. J. H. R., Goffau, D. W., Tavasszy, A. L., & Saes, M. (2016). Improving home delivery efficiency by using principles of address intelligence for B2C deliveries. *Transportation Research Procedia*, 12, <https://doi.org/10.1016/j.trpro.2016.02.006>
8. Florio, M. A., Feillet, D., & Hartl, F. R. (2018). The delivery problem: Optimizing hit rates in e-commerce deliveries. *Transportation Research Part B: Methodological*, 117 (A), <https://doi.org/10.1016/j.trb.2018.09.011>
9. Wani, D., Singh, R., Khanapuri, B. V., & Tiwari, M. (2022). Delay prediction to mitigate E-commerce supplier disruptions using voting mechanism. *IFAC-PapersOnLine*, 55(10), <https://doi.org/10.1016/j.ifacol.2022.09.495>
10. Hughes, S., Morena, S., Yushimito, F. W., & Huerta-Canepa, G. (2019). Evaluation of machine learning methodologies to predict stop delivery times from GPS data. *Transportation Research Part C: Emerging Technologies*, 119, <https://doi.org/10.1016/j.trc.2019.10.018>
11. Lolla, R., et al. (2023). Machine Learning Techniques for Predicting Risks of Late Delivery. *Data Science and Emerging Technologies*. Springer, Singapore. [https://doi.org/10.1007/978-981-99-0741-0\\_25](https://doi.org/10.1007/978-981-99-0741-0_25)
12. Jonquais, J. C. A., & Krempf, F. (2019). Predicting shipping time with machine learning. *Supply Chain Management – Massachusetts Institute of Technology* [https://dspace.mit.edu/bitstream/handle/1721.1/121280/Jonquais\\_Krempf\\_2019.pdf?sequence=1&isAl](https://dspace.mit.edu/bitstream/handle/1721.1/121280/Jonquais_Krempf_2019.pdf?sequence=1&isAl)

13. Tsolaki, K., Vafeiadis, T., Nizamis, A., Ioannidis, D., & Tzovaras, D. (2023). Utilizing machine learning on freight transportation and logistics applications: A review. *ICT Express*, 9(3), <https://www.sciencedirect.com/science/article/pii/S2405959522000200>
14. Chun, M. (2014). Improving inbound visibility through shipment arrival modeling. *MITLibraries*. <https://dspace.mit.edu/handle/1721.1/90779>
15. Alnahhal, M., Ahrens, D., & Salah, B. (2021). Dynamic Lead-Time Forecasting Using Machine Learning in a Make-to-Order Supply Chain. *Applied Sciences*. 11(21):10105. <https://doi.org/10.3390/app112110105>
- 16.
17. Song, Y. Y., & Lu, Y. (2015). Decision tree methods: applications for classification and prediction. *Shanghai archives of psychiatry*, 27(2), 130–135. <https://doi.org/10.11919/j.issn.1002-0829.215044>
18. Sperandei, S. (2014). Understanding logistic regression analysis. *Biochemia Medica*, 24(1), 12-18. <https://doi.org/10.11613/BM.2014.003>
19. Ontivero-Ortega, M., Lage-Castellanos, A., Valente, G., Goebel, R., & Valdes-Sosa, M. (2017). Fast Gaussian Naïve Bayes for searchlight classification analysis. *NeuroImage*, 163. <https://doi.org/10.1016/j.neuroimage.2017.09.001>
20. Ladicky, L. & Torr, H.S.P. (n.d.) Locally Linear Support Vector Machines. [http://www.icml-2011.org/papers/508\\_icmlpaper.pdf](http://www.icml-2011.org/papers/508_icmlpaper.pdf)
21. Kusumawati, R., D'arofah, A., & Pramana, P. A. (2019). *IOP SCIENCE*. <https://iopscience.iop.org/article/10.1088/1742-6596/1320/1/012016/meta>
22. Bui, D. T., Tuan, T. A., Klempe, H., Pradhan, B., & Revhaug, I. (2015). Spatial prediction models for shallow landslide hazards: A comparative assessment of the efficacy of support vector machines, artificial neural networks, kernel logistic regression, and logistic model tree. *Landslides*, 12(2), 331-347. <https://link.springer.com/article/10.1007/s10346-015-0557-6>
23. Rahman, A., & Tasnim, S. (2014). Ensemble Classifiers and Their Applications: A Review. *International Journal of Computer Trends and Technology (IJCTT)*, 10(1), 31. ISSN: 2231-2803. <https://arxiv.org/ftp/arxiv/papers/1404/1404.4088.pdf>
24. Geiger, B. C. (2021). On Information Plane Analyses of Neural Network Classifiers—A Review. *IEEE*. <https://ieeexplore.ieee.org/abstract/document/9468892>