|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ \_\_\_\_\_\_Информатика и системы управления\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_\_Системы обработки информации и управления\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ***

***НА ТЕМУ:***

***\_\_\_\_\_\_\_\_\_\_\_Предсказание своевременной\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_поставки электронного\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ оборудования \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

Студент \_\_\_ИУ5-63Б\_\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_Саргсян А.А.\_\_\_\_\_**

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_Гапанюк Ю.Е.\_\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

Консультант **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_ Гапанюк Ю.Е.\_\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

*2024 г.*

**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

УТВЕРЖДАЮ

Заведующий кафедрой \_\_\_\_\_\_\_\_\_\_

(Индекс)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(И.О.Фамилия)

« \_\_\_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_ 20 \_\_\_\_ г.

**ЗАДАНИЕ**

**на выполнение научно-исследовательской работы**

по теме \_\_\_\_Предсказание своевременной поставки электронного оборудования\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент группы \_\_ИУ5-63Б\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Саргсян Александр Альбертович\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, имя, отчество)

Направленность НИР (учебная, исследовательская, практическая, производственная, др.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Исследовательская\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Источник тематики (кафедра, предприятие, НИР) \_\_\_\_\_\_\_\_\_\_\_\_\_НИР\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

График выполнения НИР: 25% к \_\_\_ нед., 50% к \_\_\_ нед., 75% к \_\_ нед., 100% к \_\_\_ нед.

Техническое задание \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_Исследовать методы машинного обучения для решения задачи классификации\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Оформление научно-исследовательской работы:***

Расчетно-пояснительная записка на \_\_27\_\_ листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи задания « 13 » февраля 2024 г.

**Руководитель НИР**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_Гапанюк Ю.Е.\_\_\_\_\_

(Подпись, дата) (И.О.Фамилия)

**Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_Саргсян А.А.\_\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

**Содержание**

[**Введение** 4](#_Toc133806658)

[**Постановка задачи** 6](#_Toc133806659)

[**Выполнение работы** 7](#_Toc133806660)

[**Создание веб-приложения** 29](#_Toc133806661)

[**Заключение** 31](#_Toc133806662)

[**Список использованной литературы** 32](#_Toc133806663)

# **Введение**

Проблема задержки товаров очень актуальна в наше время. Продавцы всячески пытаются решить эту проблему, собирая и анализируя данные поставок. Однако можно своевременно предсказать какие товары скорее всего приедут с задержкой и более внимательно отслеживать их во время доставки.

В данной работе мы будем использовать данные, полученные из отчетов компании о доставках электронного оборудования, чтобы построить модель машинного обучения, которая сможет предсказывать проблемные доставки. Мы будем использовать алгоритмы классификации для определения факторов риска, включая складской блок, способ доставки, количество звонков в службу поддержки клиентов, рейтинг компании от клиентов, стоимость изделия, количество предыдущих заказов, параметр важности продукта, пол клиента, номинал предлагаемой скидки, вес продукта и параметр отслеживания (не)своевременной доставки продукции.

Целью данной работы является разработка эффективной модели, которая может помочь работникам быстро и точно определить вероятность возникновения проблем при доставке и принять меры для предотвращения задержек.

Для достижения поставленной цели были определены следующие этапы:

1. Поиск и выбор набора данных для построения моделей машинного обучения для решения задачи регрессии или классификации.
2. Проведение разведочного анализа данных.
3. Выбор признаков, подходящих для построения моделей.
4. Кодирование категориальных признаков. Масштабирование данных. Формирование вспомогательных признаков, улучшающих качество моделей.
5. Проведение корреляционного анализа данных. Формирование промежуточных выводов о возможности построения моделей машинного обучения.
6. Выбор метрик для последующей оценки качества моделей.
7. Выбор наиболее подходящих моделей для решения задачи классификации или регрессии.
8. Формирование обучающей и тестовой выборок на основе исходного набора данных.
9. Построение базового решения (baseline) для выбранных моделей без подбора гиперпараметров и оценка качества моделей на основе тестовой выборки.
10. Подбор гиперпараметров для выбранных моделей. Построение оптимальных моделей.
11. Формирование выводов о качестве построенных моделей на основе выбранных метрик.
12. Создание веб-приложение для демонстрации хотя бы одной модели машинного обучения.

# **Постановка задачи**

Данная работа по машинному обучению направлена на решение задачи классификации, а именно, предсказание своевременной поставки электронного оборудования.

Имеются данные о доставках электронной продукции, которые включают информацию о таких факторах, как складской блок, способ доставки, количество звонков в службу поддержки клиентов, рейтинг компании от клиентов, стоимость изделия, количество предыдущих заказов, параметр важности продукта, пол клиента, номинал предлагаемой скидки, вес продукта и параметр отслеживания (не)своевременной доставки продукции. Каждая доставка может быть классифицирована как доставленная вовремя или не доставленная вовремя.

Целью задачи является создание модели машинного обучения, которая будет использовать имеющиеся данные для предсказания риска несвоевременной поставки электронного оборудования. Для этого мы будем использовать различные алгоритмы классификации, такие как K ближайших соседей, метод опорных векторов, дерево решений, случайный лес и градиентный бустинг. Модель должна обучаться на тренировочных данных и проверяться на тестовых данных для оценки ее точности и эффективности.

Результатом работы должна быть модель, которая сможет предсказывать доставят ли товар вовремя или нет, и помочь продавцам оптимизировать функционирование доставок для дальнейшего улучшения условий доставки продукции.

# **Выполнение работы**

Для решения задачи классификации был выбран набор данных содержащий информацию о доставках.

В наборе данных присутствуют следующие столбцы:

* warehouse block: складской блок
* mode of shipment: способ доставки
* customer care calls: количество звонков клиента, сделанных в службу поддержки
* customer rating: рейтинг компании по мнению клиента
* cost of the product: стоимость продукта в долларах США
* prior purchases: количество предыдущих заказов
* product importance: важность продукта (низкая/средняя/высокая)
* gender: пол клиента
* discount offered: предлагаемая скидка
* weight in gms: вес груза в граммах
* not reached on time: переменная проверки, вовремя ли добралась доставка (1 – продукт доставлен не вовремя, 0 – продукт доставлен вовремя)

Данный датасет использован для решения задачи классификации - предсказания своевременной или задержанной поставки электронного оборудования.

Загружаем данные, получаем общую информацию о датасете и делаем предположения о влиянии признаков на целевую переменную. В наборе данных содержится 10999 строк и 11 столбцов, из которых 7 типа int64 и 4 типа object.

Меняем тип колонок warehouse\_block, mode\_of\_shipment, product\_importance, gender на ‘category’, так как она очевидно содержит категориальные данные и может внести несуществующие закономерности в данные.

Пропусков не было обнаружено.

Строим график pairplot для визуализации распределения данных попарно для множества колонок.

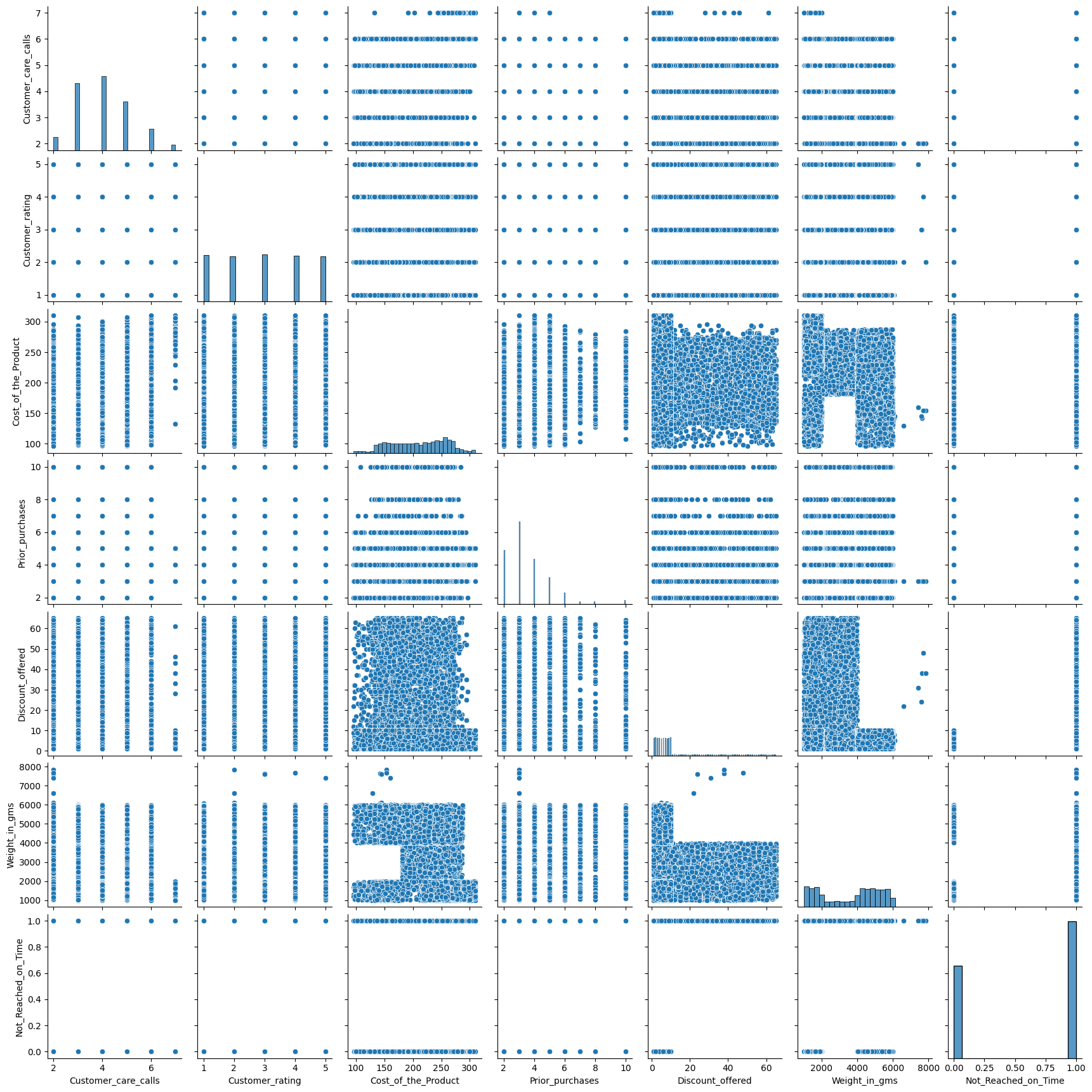


Рисунок 1 - Визуализация распределения данных попарно для множества колонок

Проверяем сбалансированы ли классы в нашем наборе данных. Получаем следующую гистограмму:

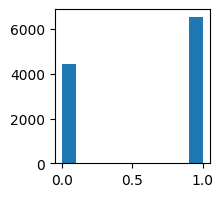


Рисунок 2 - Гистограмма классов

Видим, что классы немножко не сбалансированы.

Строим таблицу средних значений с группировкой по целевому признаку и делаем следующие предположения:

* у недоставленных вовремя товаров скидка сильно больше
* у недоставленных вовремя товаров вес значительно ниже
* у недоставленных вовремя товаров цена немного ниже

Подтвердим наши предположения графиками.

Посмотрим влияет ли размер скидки на целевой признак. Строим гистограмму, а также воспользуемся t-тестом, чтобы удостовериться что распределение не случайно. Получаем следующие значения:

t-statistic: 45.3744750332705, p-value: 0.0

P-значение называется вероятностью того, что результаты выборки данных произошли случайно. Делаем вывод, что размер скидки влияет на целевой признак, так как p-value равен 0.

Строим гистограмму зависимости размера скидки от целевого признака.

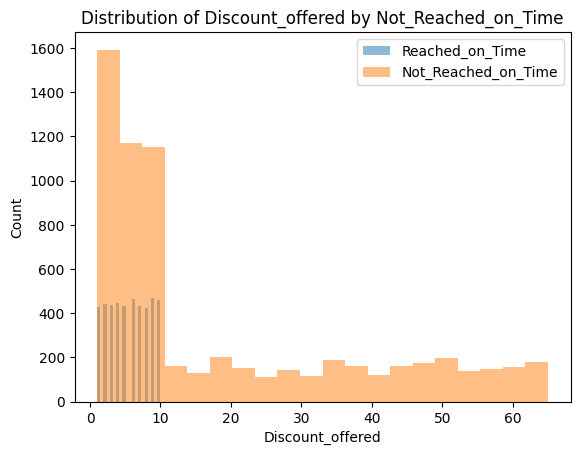


Рисунок 3 - Гистограмма зависимости размера скидки от целевого признака

Можно заметить, что при скидке до 10% доставленных вовремя товаров в 2-3 раза меньше, а при скидке более 10% доставленных вовремя товаров вообще нет.

Построим гистограмму зависимости веса продукции от произведенной вовремя доставки и проверим t-статистику.

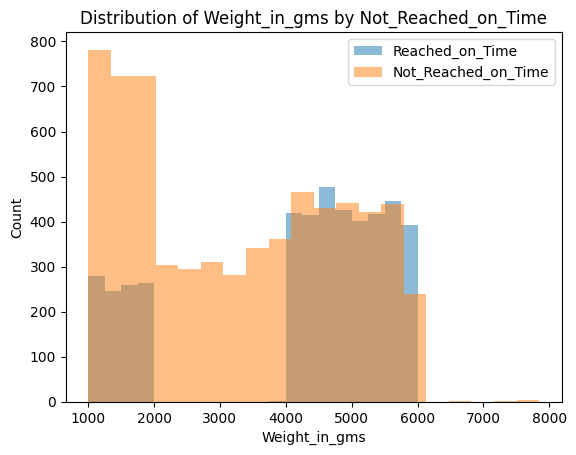


Рисунок 4 - Гистограмма зависимости веса продукции от произведенной вовремя доставки

Видно, что чем меньше вес товара, тем больше шанс того, что товар приедет не вовремя.

t-statistic: -29.264343461838504, p-value: 2.3546582802914183e-181

Построим гистограмму зависимости веса продукции от произведенной вовремя доставки и проверим t-статистику.

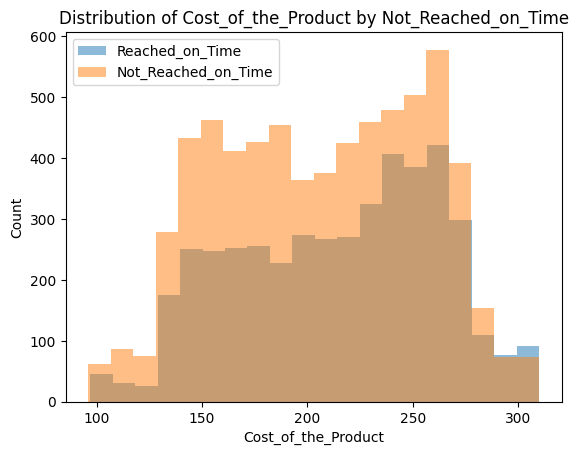


Рисунок 5 - Гистограмма зависимости стоимости продукта от произведенной вовремя доставки

В среднем, стоимость не доставленного вовремя товара выше.

t-statistic: -7.737818128158982, p-value: 1.099885972621861e-14

Далее приведем данные к нужному формату. Сначала масштабируем численные признаки методом Standard Scaler, который преобразует каждый признак таким образом, чтобы он имел среднее значение равное 0 и стандартное отклонение равное 1. Посмотрим на распределения колонок до и после масштабирования.

Таблица 1 - Распределение численных колонок до и после масштабирования

|  |  |
| --- | --- |
| До масштабирования | После масштабирования |
| Рисунок 10 - Распределение стоимости до масштабирования | Рисунок 11 - Распределение стоимости после масштабирования |
| Рисунок 12 - Распределение скидки до масштабирования | Рисунок 13 - Распределение скидки после масштабирования |
| Рисунок 14 - Распределение веса продукции до масштабирования | Рисунок 15 - Распределение веса продукции после масштабирования |

Распределение не изменилось.

Затем используем One Hot encoding для кодирования колонок warehouse\_block, mode\_of\_shipment, product\_importance, gender. В этом случае каждое уникальное значение признака становится новым отдельным признаком.

Проводим корреляционный анализ данных. Строим тепловую карту корреляций.

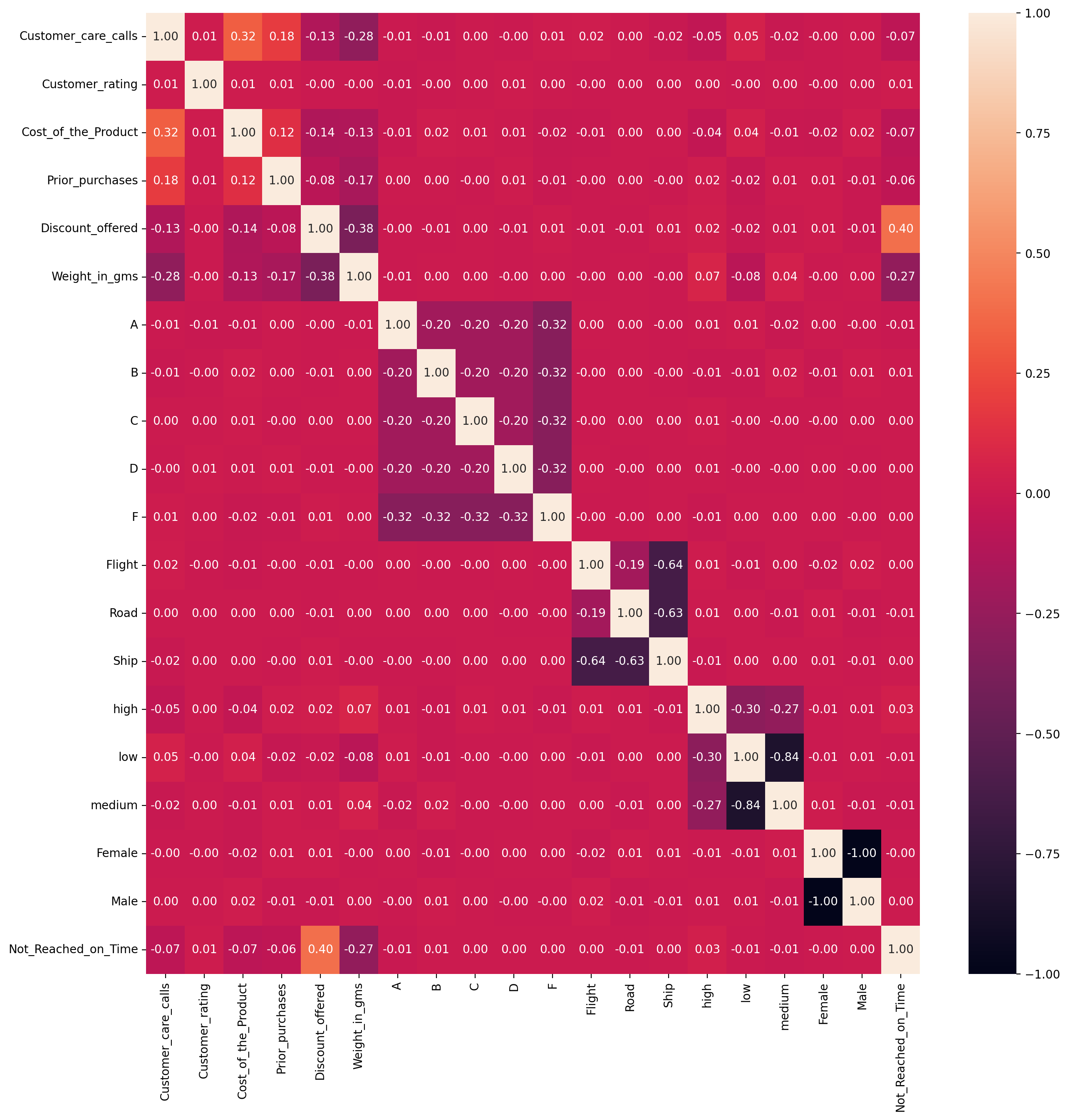


Рисунок 16 - Тепловая карта корреляций

Выводы:

* целевой признак Not\_Reached\_on\_Time больше всего коррелирует с размером скидки (0.40), весом продукции (-0.27);
* столбцы с количеством звонков клиента, сделанных в службу поддержки

(-0.07), стоимостью продукции (-0.07), оставим для построения модели, т.к. они тоже могут влияние на целевой признак;

* столбцы со количеством предыдущих заказов (-0.06) и «высоким» уровнем ценности товара (0.03) также оставим, т.к. выявлена хоть и небольшая, но возможность влияния на целевой признак;
* столбцы A, B, C, D, F (складские блоки), Flight, Road, Ship (способы доставки) и пол клиента не имеют корреляции с целевым признаком.

Выберем метрики для оценки качества модели:

* - показывает, какую долю объектов, которые модель предсказала как положительные, действительно являются положительными.
* - показывает, какую долю положительных объектов модель способна обнаружить.
* - среднее гармоническое precision и recall. Другими словами, это средневзвешенное значение точности и отзыва. [2]
* - oснована на вычислении следующих характеристик: - True Positive Rate, откладывается по оси ординат. Совпадает с recall. - False Positive Rate, откладывается по оси абсцисс. Показывает какую долю из объектов отрицательного класса алгоритм предсказал неверно. Идеальная ROC-кривая проходит через точки (0,0)-(0,1)-(1,1), то есть через верхний левый угол графика. Чем сильнее отклоняется кривая от верхнего левого угла графика, тем хуже качество классификации. [3]

Выберем модели для решения задачи классификации:

* KNN;
* SVC;
* Дерево решений;
* Случайный лес;
* Градиентный бустинг.

Формируем обучающую и тестовую выборку в соотношении 8:2. Оставляем колонки «Customer\_care\_calls», «Cost\_of\_the\_Product», «Prior\_purchases», «Discount\_offered», «Weight\_in\_gms», «high», т.к. они влияют на целевой признак.

Строим базовое решения, выводим значениями метрик и ROC-кривую.

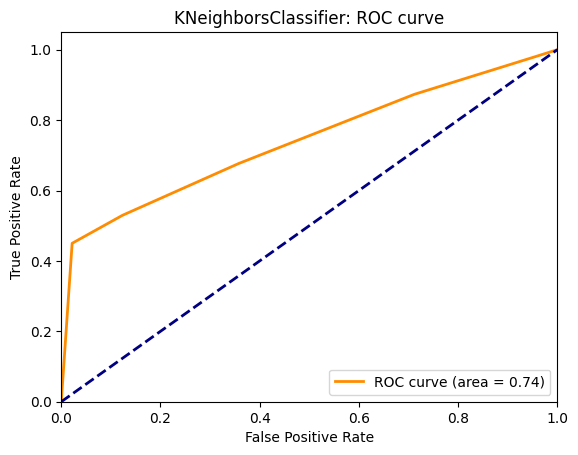


Рисунок 17 - ROC-кривая базовой модели KNN

KNeighborsClassifier:

Precision: 0.74

Recall: 0.68

F1-score: 0.71

ROC AUC score: 0.7403174322992512

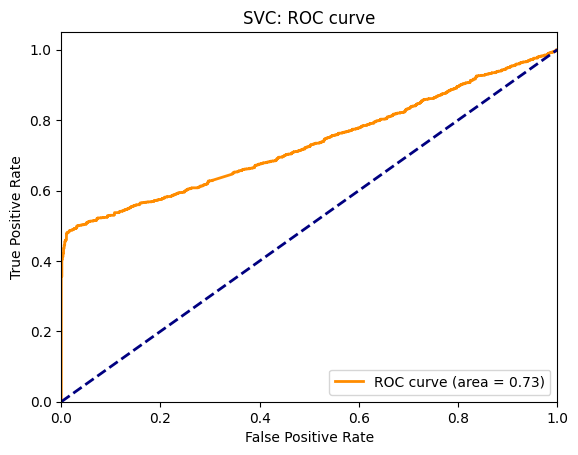


Рисунок 18- ROC-кривая базовой модели SVC

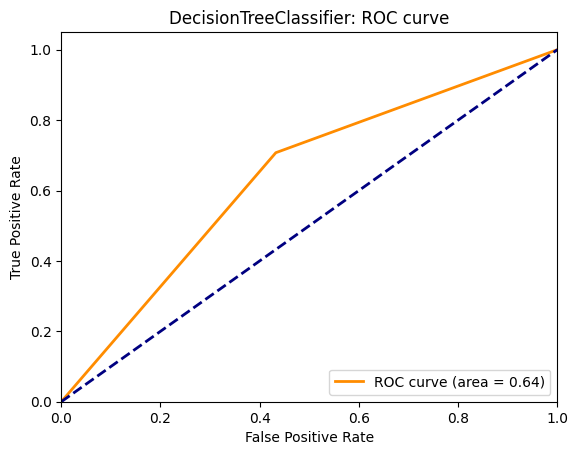
SVC:

Precision: 0.95

Recall: 0.5

F1-score: 0.66

ROC AUC score: 0.7322453450732926



*Рисунок 19 - ROC-кривая базовой модели Decision Tree*

DecisionTreeClassifier:

Precision: 0.72

Recall: 0.71

F1-score: 0.71

ROC AUC score: 0.6374504525785426

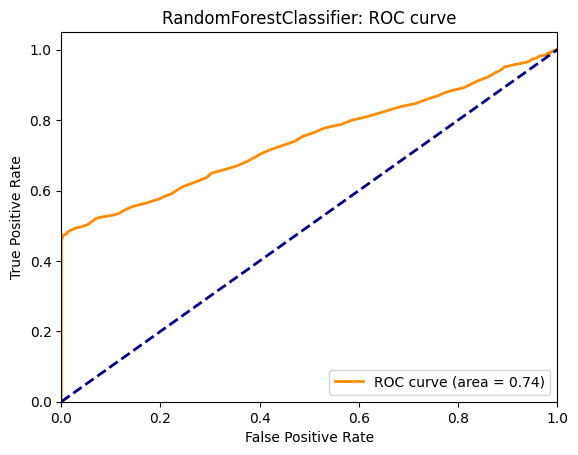


Рисунок 20 - ROC-кривая базовой модели Random Forest

RandomForestClassifier:

Precision: 0.75

Recall: 0.66

F1-score: 0.7

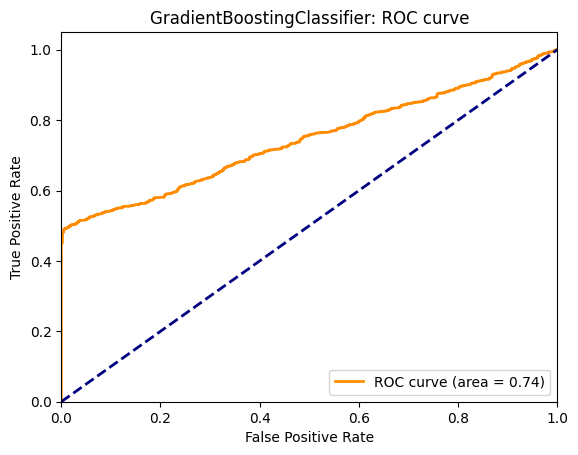
ROC AUC score: 0.743556996515565 

Рисунок 21 - ROC-кривая базовой модели Gradient Boosting

GradientBoostingClassifier:

Precision: 0.91

Recall: 0.54

F1-score: 0.67

ROC AUC score: 0.7442457500188195

Используем GridSearch для поиска оптимальных гиперпараметров для каждой модели.

KNeighboursClassifier:

Best hyperparameters: {'algorithm': 'auto', 'n\_neighbors': 8, 'weights': 'uniform'}

SVC:

Best hyperparameters: {'C': 1, 'degree': 4, 'gamma': 'scale', 'kernel': 'rbf'}

DecisionTreeClassifier:

Best hyperparameters: {'criterion': 'gini', 'max\_depth': 5, 'max\_features': 'log2', 'min\_samples\_leaf': 2, 'min\_samples\_split': 5}

RandomForestClassifier:

Best hyperparameters: {'max\_depth': 5, 'max\_features': None, 'min\_samples\_leaf': 1, 'min\_samples\_split': 5, 'n\_estimators': 100}

GradientBoostingClassifier:

Best hyperparameters: {'learning\_rate': 0.1, 'max\_depth': 3, 'max\_features': None, 'min\_samples\_leaf': 4, 'min\_samples\_split': 2}

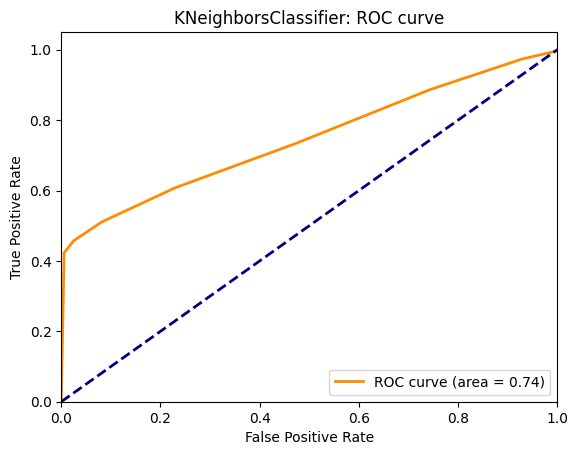


Рисунок 22 - ROC-кривая модели KNN после поиска гиперпараметров

KNeighborsClassifier:

Precision: 0.8

Recall: 0.61

F1-score: 0.69

ROC AUC score: 0.7447099664189403

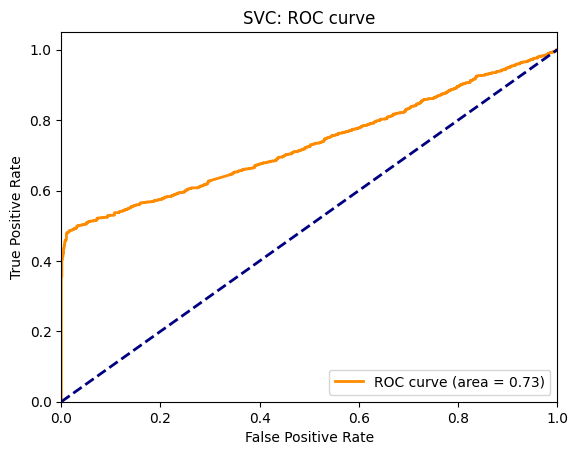


Рисунок 23 - ROC-кривая модели SVC после поиска гиперпараметров

SVC:

Precision: 0.95

Recall: 0.5

F1-score: 0.66

ROC AUC score: 0.7322713031198976

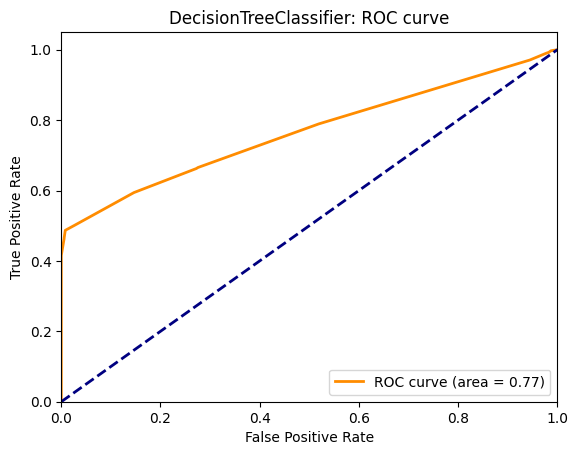


Рисунок 24 - ROC-кривая модели Decision Tree после поиска гиперпараметров

DecisionTreeClassifier:

Precision: 0.97

Recall: 0.5

F1-score: 0.66

ROC AUC score: 0.7658675049385183

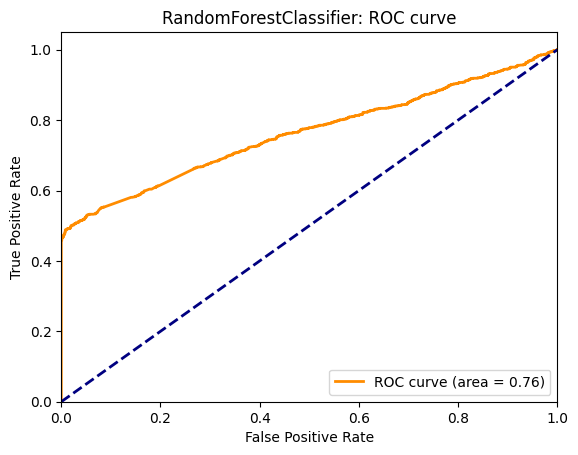


Рисунок 25 - ROC-кривая модели Random Forest после поиска гиперпараметров

RandomForestClassifier:

Precision: 0.95

Recall: 0.51

F1-score: 0.67

ROC AUC score: 0.7622333784138077

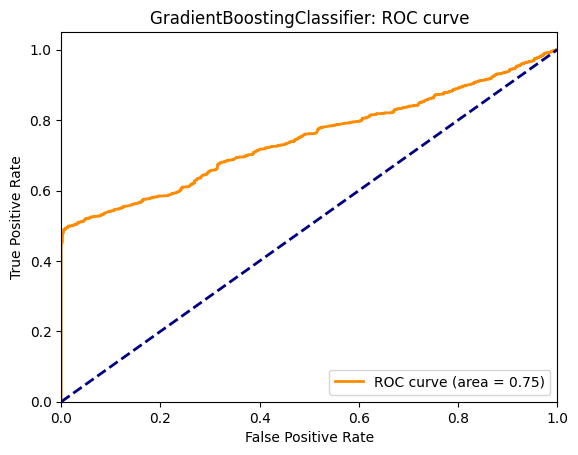


Рисунок 26 - ROC-кривая модели Gradient Boosting после поиска гиперпараметров

GradientBoostingClassifier:

Precision: 0.91

Recall: 0.53

F1-score: 0.67

ROC AUC score: 0.7467165234215128

Таблица 2 - Сравнение базовых моделей с моделями после подбора гиперпараметров по 4 метрикам

|  |  |  |
| --- | --- | --- |
| Модель | Baseline | GridSearch() |
| KNN | Precision: 0.74  Recall: 0.68  F1-score: 0.71  ROC AUC score: 0.7403174322992512 | Precision: 0.8  Recall: 0.61  F1-score: 0.69  ROC AUC score: 0.7447099664189403 |
| SVC | Precision: 0.95  Recall: 0.5  F1-score: 0.66  ROC AUC score: 0.7322453450732926 | Precision: 0.95  Recall: 0.5  F1-score: 0.66  ROC AUC score: 0.7322713031198976 |
| Decision Tree | Precision: 0.72  Recall: 0.71  F1-score: 0.71  ROC AUC score: 0.6374504525785426 | Precision: 0.97  Recall: 0.5  F1-score: 0.66  ROC AUC score: 0.7658675049385183 |
| Random forest | Precision: 0.75  Recall: 0.66  F1-score: 0.7  ROC AUC score: 0.743556996515565 | Precision: 0.95  Recall: 0.51  F1-score: 0.67  ROC AUC score: 0.7622333784138077 |
| Gradient Boosting | Precision: 0.91  Recall: 0.54  F1-score: 0.67  ROC AUC score: 0.7442457500188195 | Precision: 0.91  Recall: 0.53  F1-score: 0.67  ROC AUC score: 0.7467165234215128 |

На основании трех метрик из четырех лучшими для решения данной задачи классификации оказались модели градиентного бустинга и метод случайного леса.

# **Создание веб-приложения**

В качестве модели выбираем решающее дерево.

Сначала импортируем необходимые библиотеки, загружаем набор данных, обрабатываем данные: удаляем дубликаты, заполняем пропущенные значения, кодируем категориальные данные с помощью OneHotEncoder.

Добавляем функцию train\_model, которая принимает два аргумента: 'max\_depth', определяющий максимальную глубину дерева решений, и 'test\_size', определяющий размер тестового набора данных.

Внутри функции происходит разделение данных на обучающий и тестовый наборы, используя функцию 'train\_test\_split' из библиотеки sklearn. Затем создается экземпляр класса DecisionTreeClassifier из библиотеки sklearn, устанавливая параметр максимальной глубины, переданный в качестве аргумента. Модель обучается на обучающем наборе данных, вызывая метод 'fit'.

Затем модель используется для предсказания на тестовом наборе данных, вызывая метод 'predict', и вычисляются значения recall и precision, используя функции 'recall\_score' и 'precision\_score' из библиотеки sklearn.

Функция возвращает обученную модель решающего дерева, а также значения recall и precision в виде кортежа (model, recall, precision).

Функция 'app' создает интерфейс приложения на Streamlit.

С помощью функции 'st.title' устанавливается заголовок страницы. Затем создаются два слайдера для настройки максимальной глубины дерева и размера тестового набора данных с помощью функции.

Затем вызывается функция 'train\_model’, и результаты ее выполнения сохраняются в переменные model, recall и precision. Далее выводятся значения recall и precision.

С помощью функции 'export\_text' создается текстовое представление дерева решений, которое затем выводится с помощью функции.



Рисунок 27 - Скриншот веб-приложения

# **Заключение**

Классификация параметра, отвечающего за показатель вовремя/не вовремя доставленного товара, с помощью методов машинного обучения является актуальной и перспективной задачей в области услуг. Анализ и обработка данных с помощью алгоритмов машинного обучения могут помочь своевременно предсказать какие товары скорее всего приедут с задержкой и более внимательно отслеживать их во время доставки.

В рамках НИР была разработана эффективная модель, которая может помочь работникам быстро и точно определить вероятность возникновения проблем при доставке и принять меры для предотвращения задержек.

Данные были проанализированы, визуализированы и подготовлены к обучению. Были применены различные алгоритмы, такие как метод ближайших соседей, метод опорных векторов, дерево решений, случайный лес и градиентный бустинг.

В результате исследования было показано, что большинство использованных методов могут достичь хороших результатов, но самыми точными на основании трех метрик из четырех оказались модели градиентного бустинга и метод случайного леса.

# **Список использованной литературы**

1. T-test на Python для проверки и получения t-статистики // Помощник Python URL: https://pythonpip.ru/osnovy/t-test-na-python
2. Machine Learning Metrics in simple terms // Medium URL: https://medium.com/analytics-vidhya/machine-learning-metrics-in-simple-terms-d58a9c85f9f6
3. Опорный пример для выполнения проекта по анализу данных. // Jupyter nbviewer URL: https://scikit-learn.org/stable/tutorial/statistical\_inference/supervised\_learning.html
4. Репозиторий курса "Технологии машинного обучения", бакалавриат, 6 семестр. // GitHub URL: https://github.com/ugapanyuk/courses\_current/wiki/COURSE\_TMO\_SPRING\_2023/