



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:**

**Предсказательная модель  
расчёта заработной платы**

С  
Т  
У  
Д  
Е  
Н  
К  
И  
У  
С  
Л  
У  
Ж  
Б  
Н  
И  
К  
И  
Н  
А  
С  
Т  
А  
В  
Ш  
И  
М

(Группа)

(Подпись, дата)

(И.О.Фамилия)

(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 ИУ5  
(Индекс)  
В.И. Терехов  
(И.О.Фамилия)  
февраля \_\_\_\_\_ 2024 г.

**ЗАДАНИЕ**  
**на выполнение научно-исследовательской работы**

по теме Предсказательная модель заработной платы

Студент группы ИУ5-63Б

Балюк Андрей Валерьевич

(Фамилия, имя, отчество)

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

ИССЛЕДОВАТЕЛЬСКАЯ

Источник тематики (кафедра, предприятие, НИР) КАФЕДРА

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

Техническое задание

Построить предсказательную модель расчёта заработной платы и оценить качество её работы с помощью методов машинного обучения

**Оформление научно-исследовательской работы:**

Расчетно-пояснительная записка на 26 листах формата А4.

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

Дата выдачи задания « 07 » февраля 2024 г.

Р  
у  
Студент  
о

(Подпись, дата)

(И.О.Фамилия)

Балюк А.В.

(Подпись, дата)

(И.О.Фамилия)

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

д

и

т

е

л

ь

## Содержание

В

И

С

В

И

С

В

И

С

В

И

С

В

И

С

В

И

С

В

И

С

В

И

С

В

И

С

В

И

С

В

И

С

В

И

С

В

И

С

В

И

## Введение

Проблема выдачи кредитов людям на основе их способности выплатить его довольно распространена в нашем мире. Нужно точно предсказать, сможет ли человек из своей зарплаты выплатить кредит, дабы ни человек, ни банк не понесли потерь.

В данной работе мы будем использовать данные, полученные из отчетов компаний, в которых работают люди, чтобы построить модель машинного обучения, которая сможет предсказывать зарплату людей. Мы будем использовать алгоритмы классификации для определения факторов риска, включая возраст, класс труд устроенности, образование, и другие параметры.

Целью данной работы является разработка эффективной модели, которая может помочь работникам банка быстро и точно возможно человека выплатить кредит.

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.

Подбор гиперпараметров для выбранных моделей. Построение оптимальных моделей.

Формирование выводов о качестве построенных моделей на основе выбранных метрик.

## **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 своевременности 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 К 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
- с
- u
- v
- p
- p
- gender: пол клиента
- discount offered: предлагаемая скидка
- v
- p
- h

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

В Загружаем данные, получаем общую информацию о датасете и делаем предположения о влиянии признаков на целевую переменную. В наборе данных содержится 10999 наблюдений. Из них 10 признаков, из которых 7 типа int64 и 4 типа object.

Меняем тип колонок `warehouse_block`, `mode_of_shipment`, `gender` на `varchar` и указываем, что в этих колонках содержатся текстовые данные и можем использовать в них функции из библиотеки `GDAL` для работы с данными.

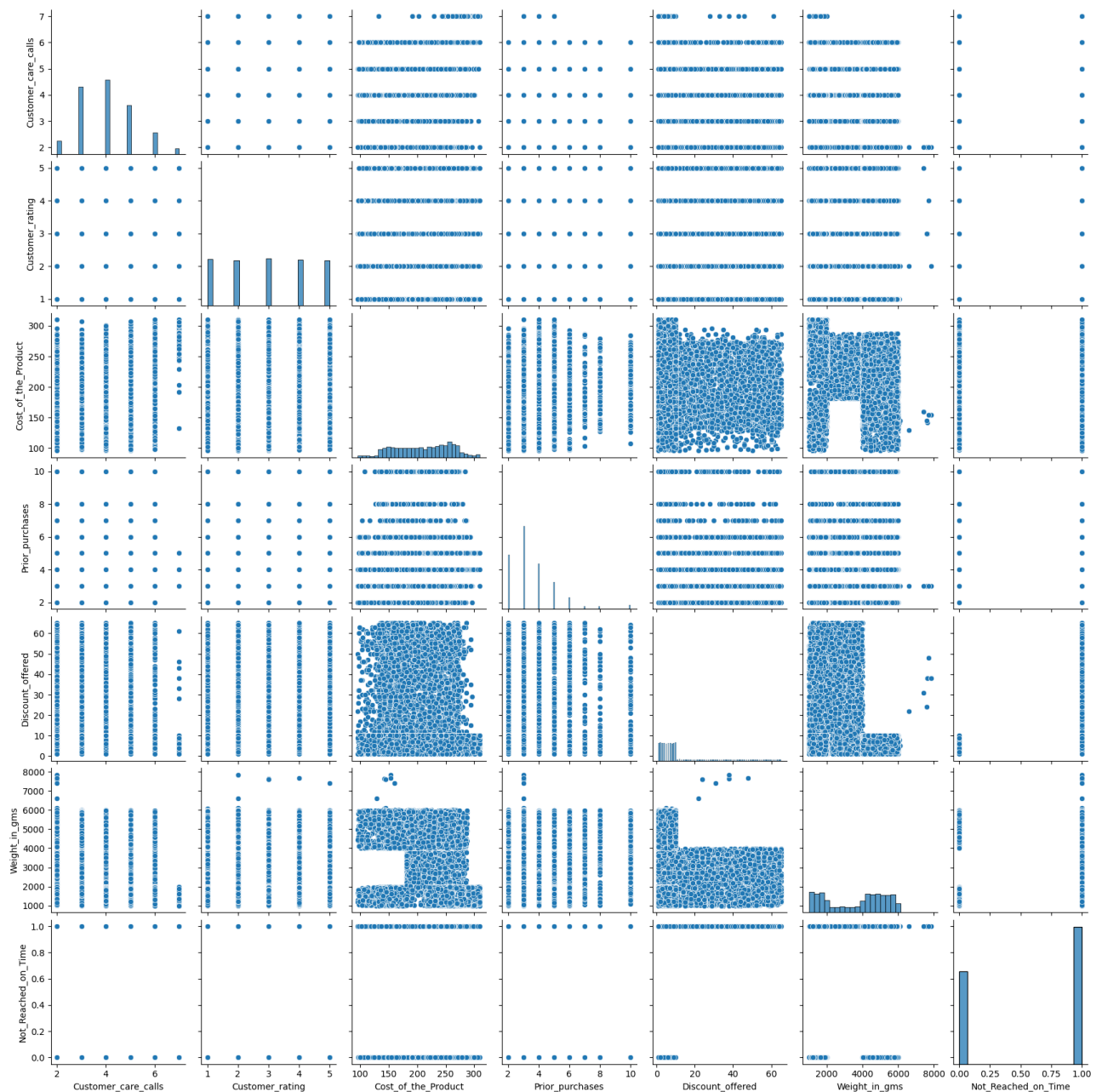
**Q**

• Пропусков не было обнаружено.

iv) Строить граф (график/решения/высказия) распределения данных попарно для множества колонок.

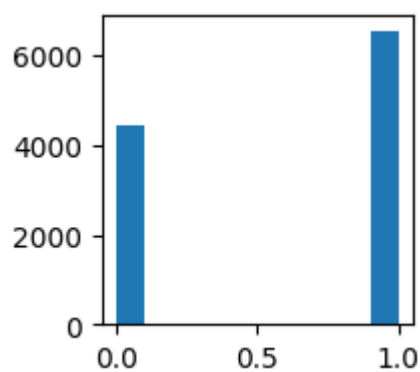
e

переменная проверки, вовремя ли добралась доставка (1 – продукт доставлен вовремя, 0 – продукт доставлен не вовремя)



*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:

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.

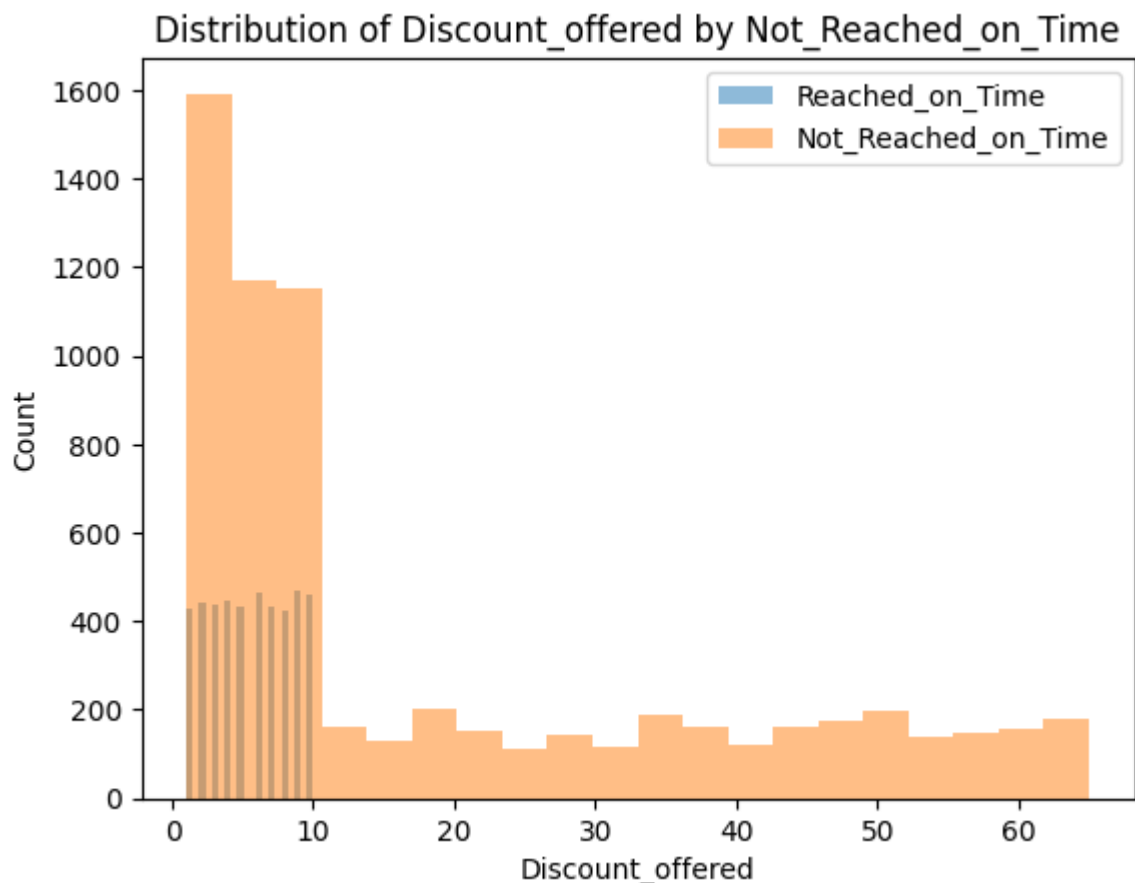
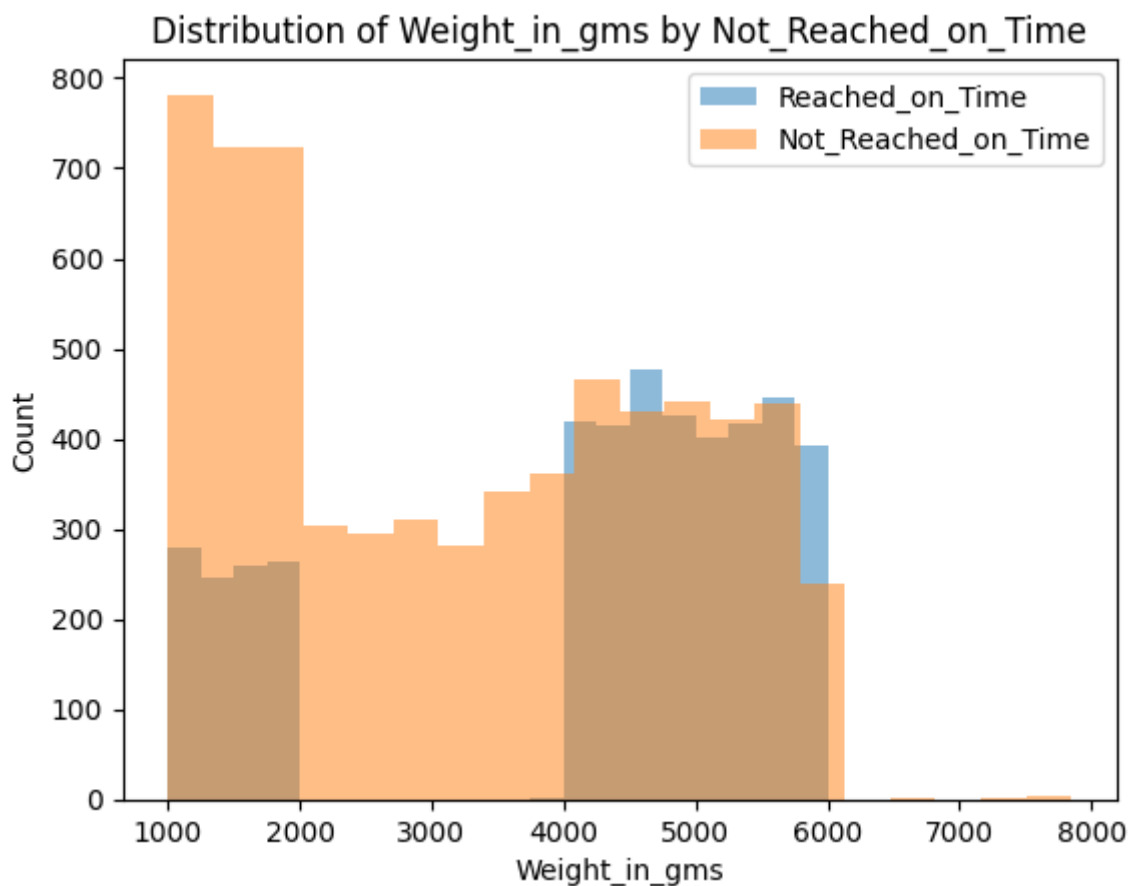


Рисунок 3 - Гистограмма зависимости размера скидки от целевого признака

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.

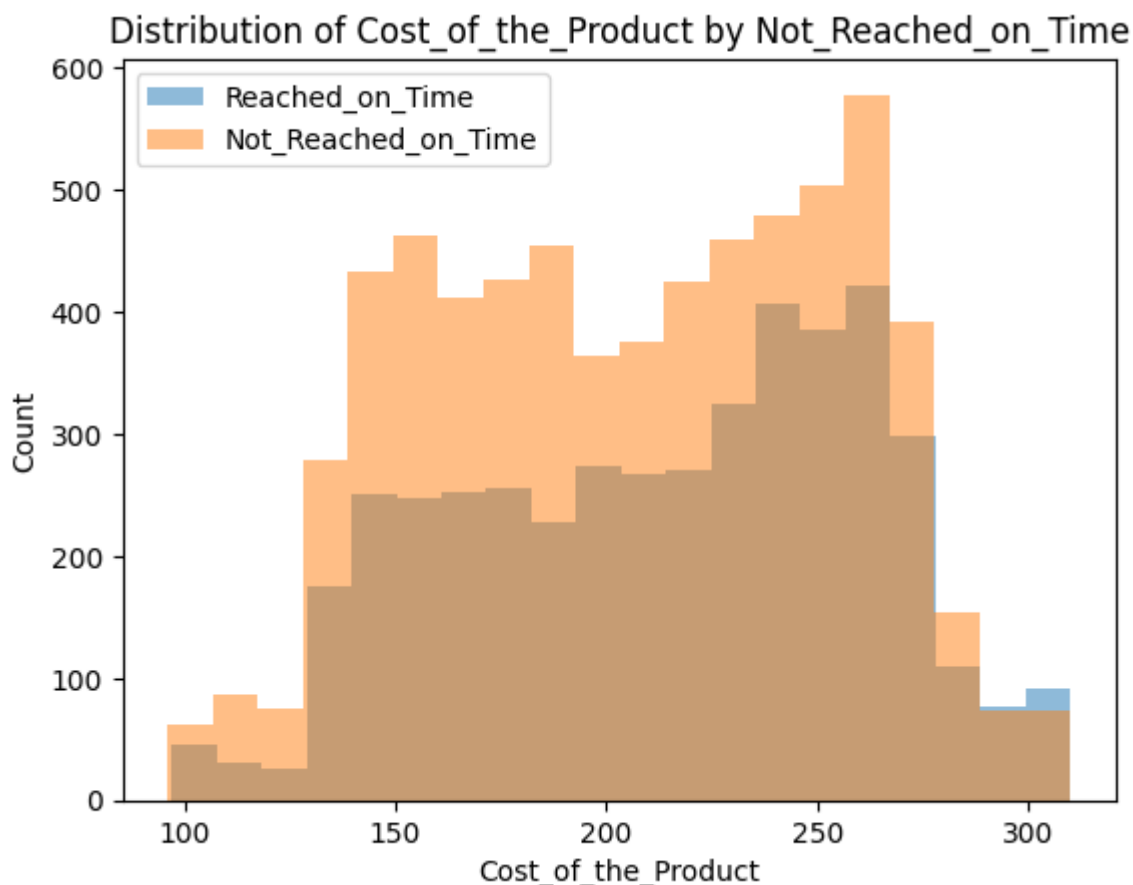


*Рисунок 4 - Гистограмма зависимости веса продукции от произведенной вовремя доставки*

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

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

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



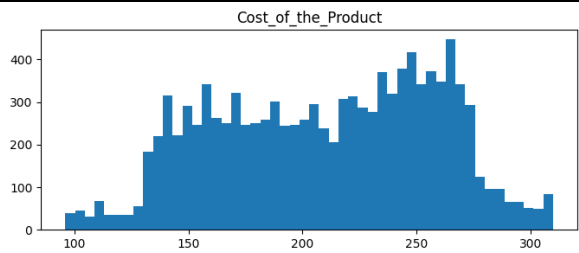
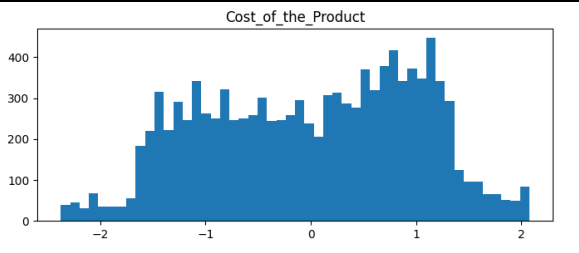
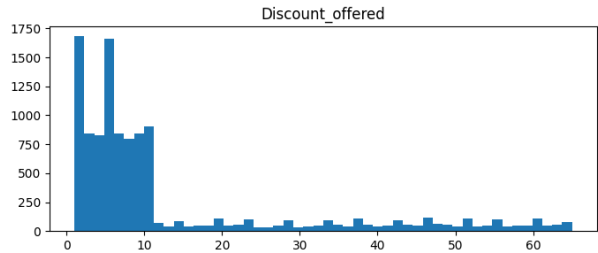
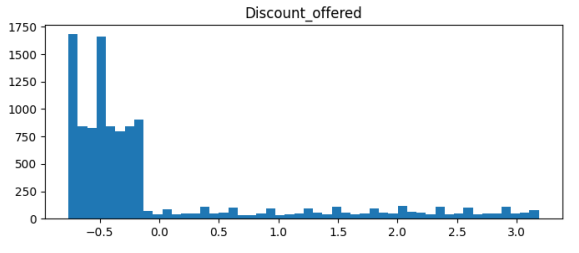
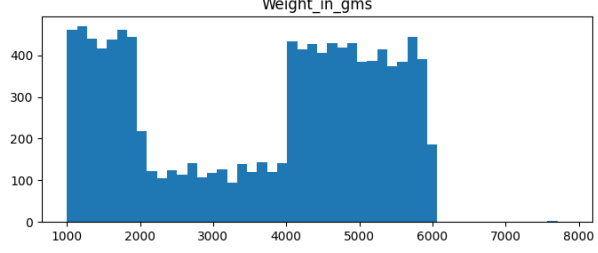
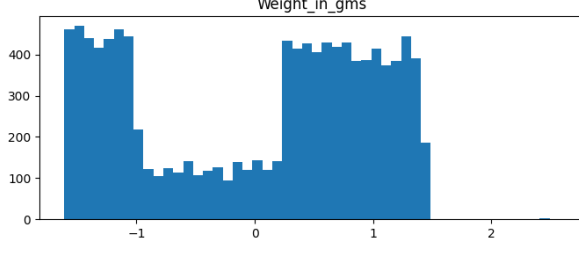
*Рисунок 5 - Гистограмма зависимости стоимости продукта от произведенной вовремя доставки*

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 равно 0 и стандартное отклонение равно 1. Посмотрим на распределения колонок до и после масштабирования.

Таблица 1 - Распределение численных колонок до и после масштабирования

До масштабирования	После масштабирования
 <p>Рисунок 6 - Распределение стоимости до масштабирования</p>	 <p>Рисунок 7 - Распределение стоимости после масштабирования</p>
 <p>Рисунок 8 - Распределение скидки до масштабирования</p>	 <p>Рисунок 9 - Распределение скидки после масштабирования</p>
 <p>Рисунок 10 - Распределение веса продукции до масштабирования</p>	 <p>Рисунок 11 - Распределение веса продукции после масштабирования</p>

Распределение не изменилось.

Затем используем One Hot encoding для кодирования колонок  
каждое уникальное значение признака становится новым отдельным признаком.

Проводим корреляционный анализ данных. Строим тепловую карту  
корреляций.

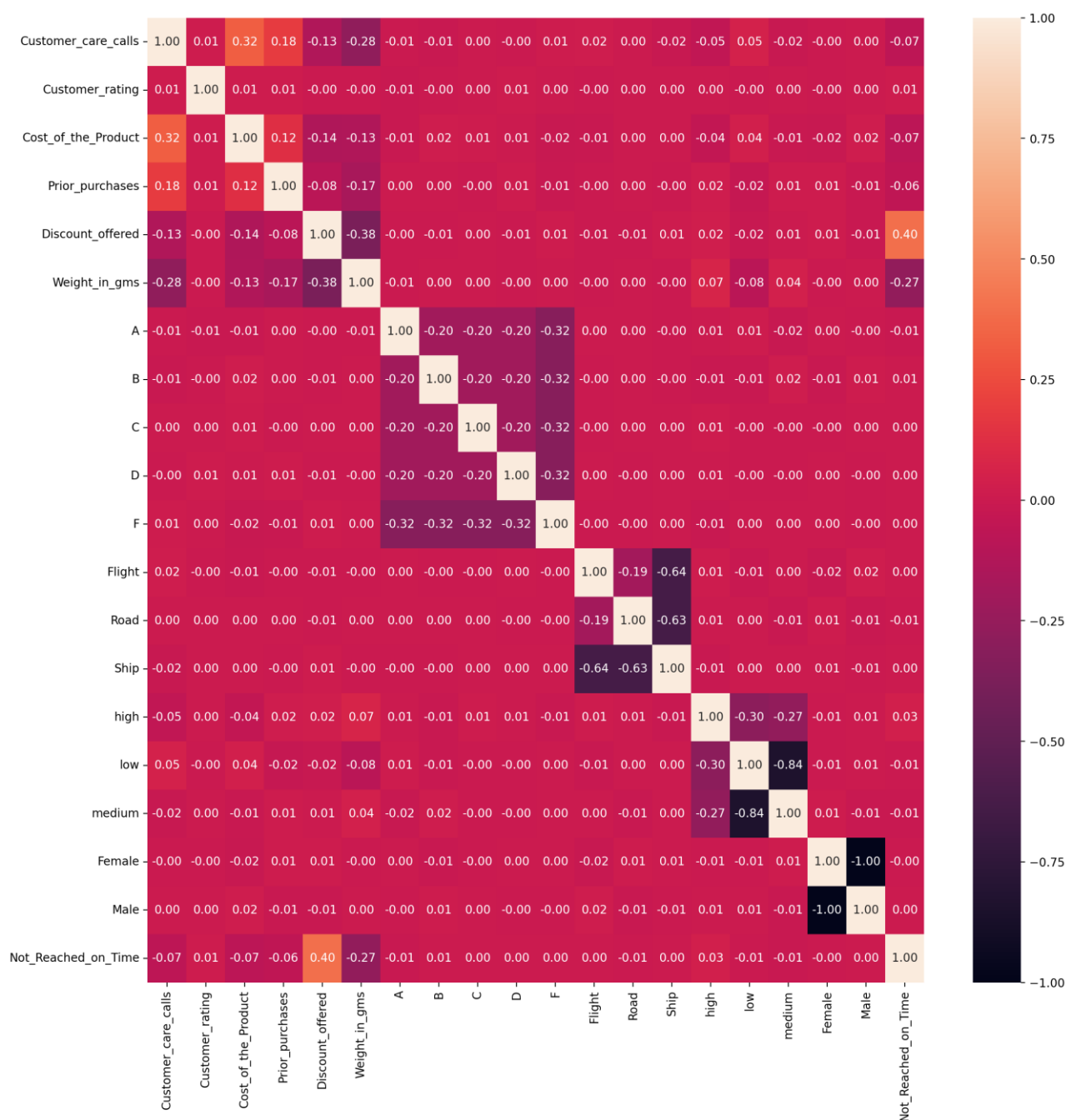


Рисунок 12 - Тепловая карта корреляций

#### 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 с 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:

- $Precision = \frac{TP}{TP+FP}$  - показывает, какую долю объектов, которые модель предсказала как положительные, действительно являются положительными.  
 $F1 = \frac{2 \times TP}{2 \times TP + FN + FP}$  - показывает, какую долю положительных объектов модель способна обнаружить.
- $F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$  - среднее гармоническое precision и recall. Другими словами, это средневзвешенное значение точности и отзыва. [2]

основана на вычислении следующих характеристик:  $TPR = \frac{TP}{TP+FN}$  - True

Positive Rate, откладывается по оси ординат. Совпадает с recall.  $FPR = \frac{FP}{FP+TN}$

- 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]

;

;

- 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 значения метрик и ROC-кривую.

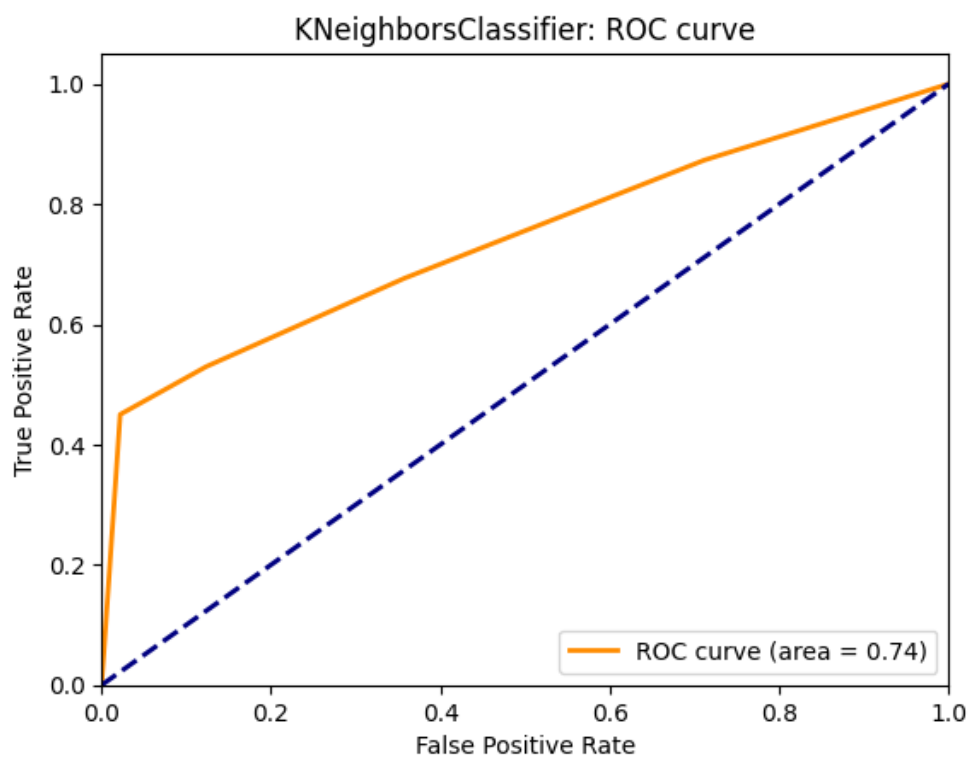


Рисунок 13 - ROC-кривая базовой модели KNN

KNeighborsClassifier:

Precision: 0.74

Recall: 0.68

F1-score: 0.71

ROC AUC score: 0.7403174322992512

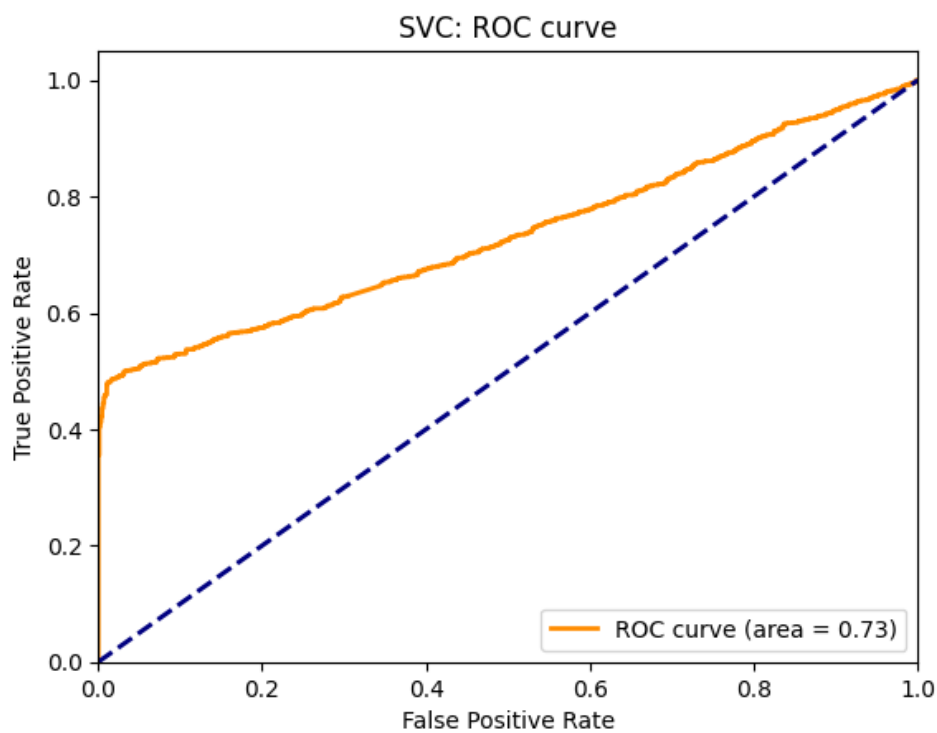


Рисунок 14- ROC-кривая базовой модели SVC

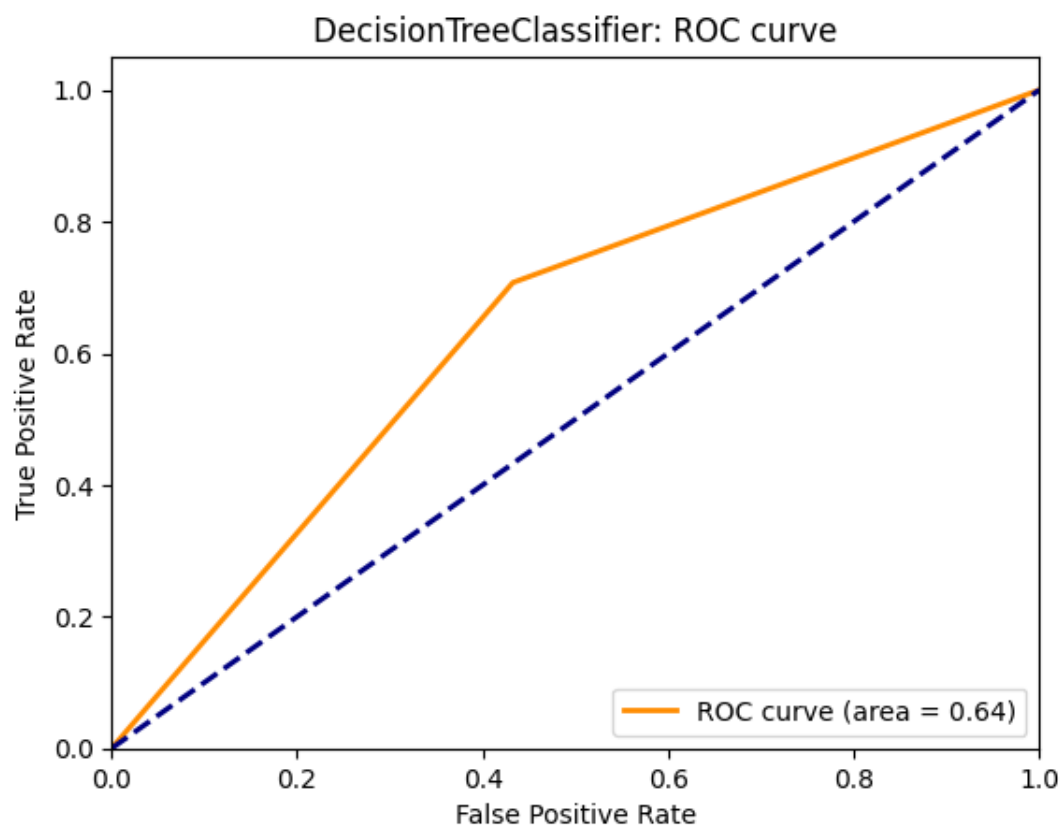
SVC:

Precision: 0.95

Recall: 0.5

F1-score: 0.66

ROC AUC score: 0.7322453450732926



*Рисунок 15 - ROC-кривая базовой модели Decision Tree*

DecisionTreeClassifier:

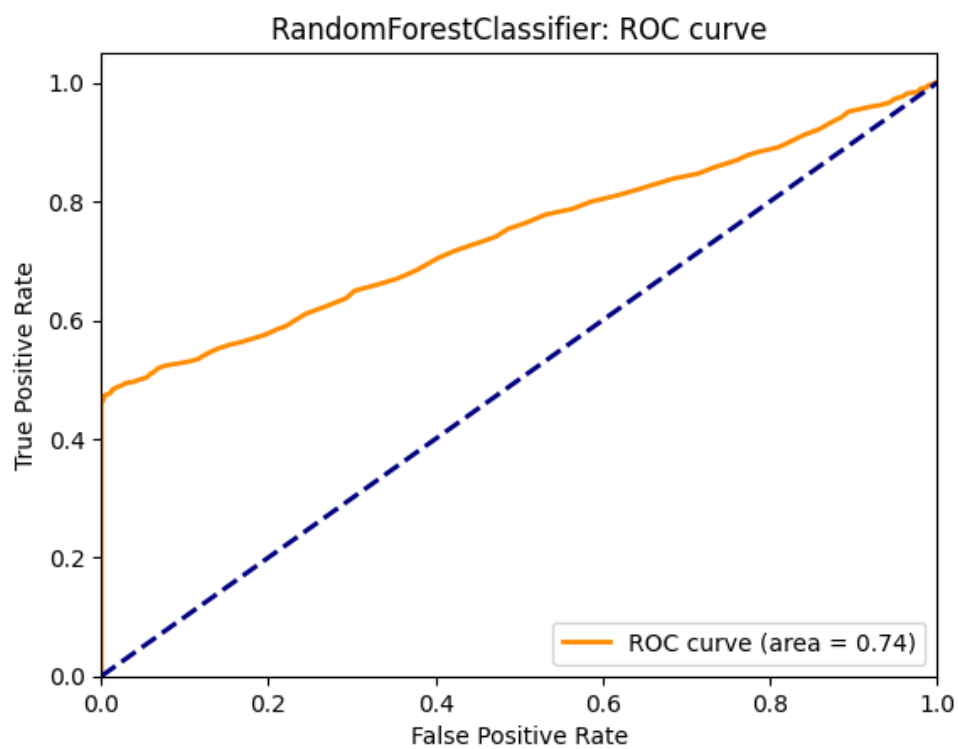
Precision: 0.72

Recall: 0.71

F1-score: 0.71

ROC AUC score: 0.6374504525785426





*Рисунок 16 - ROC-кривая базовой модели Random Forest*

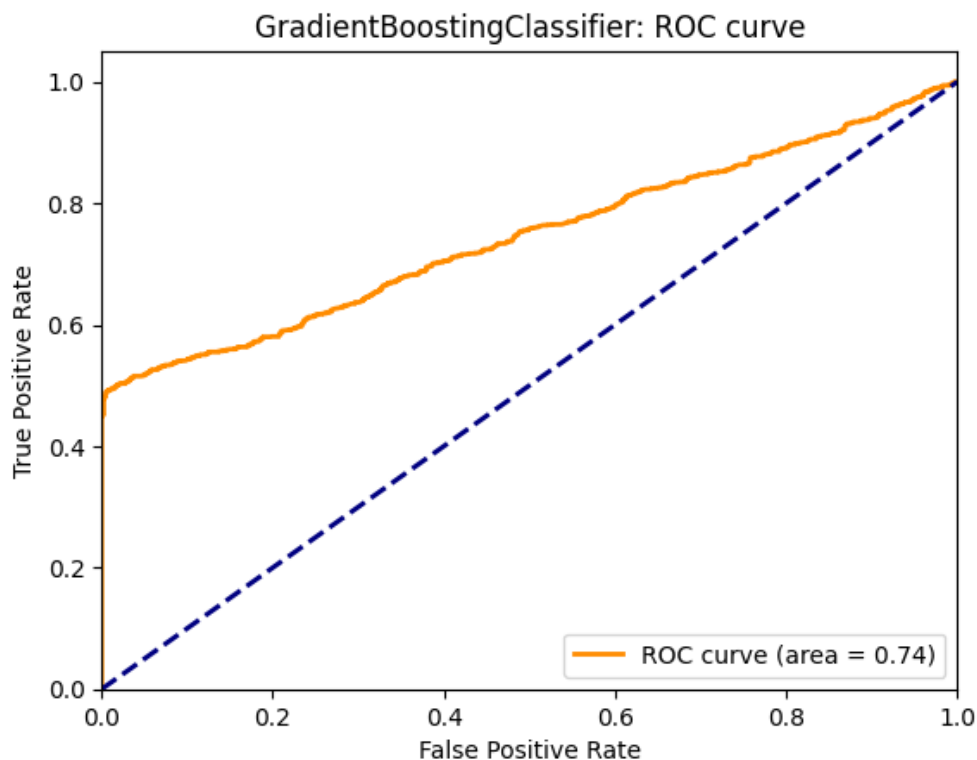
RandomForestClassifier:

Precision: 0.75

Recall: 0.66

F1-score: 0.7

ROC AUC score: 0.743556996515565



*Рисунок 17 - 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:

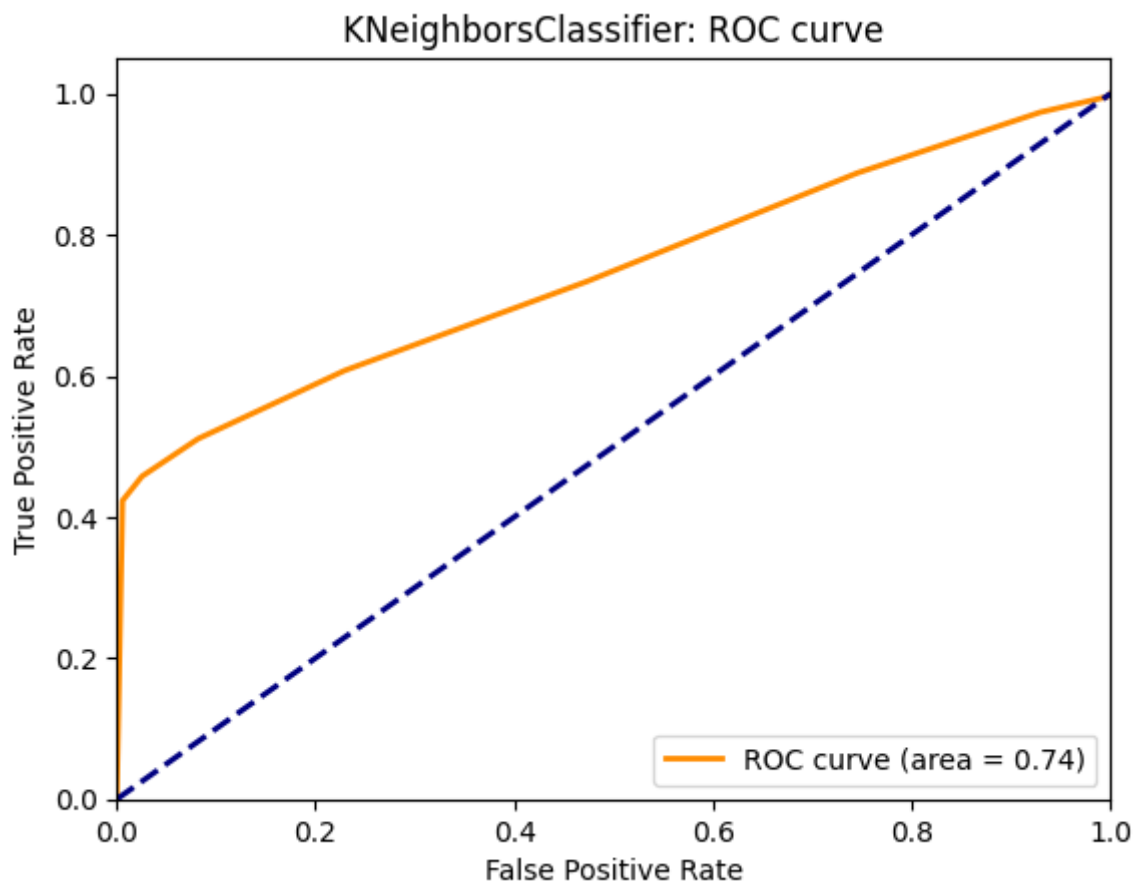
Best hyperparameters: {'criterion': 'gini', 'max\_depth': 5, 'max\_features': 'log2', 'min\_samples\_leaf': 2, 'min\_samples\_split': 5}

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 18 - ROC-кривая модели KNN после поиска гиперпараметров*

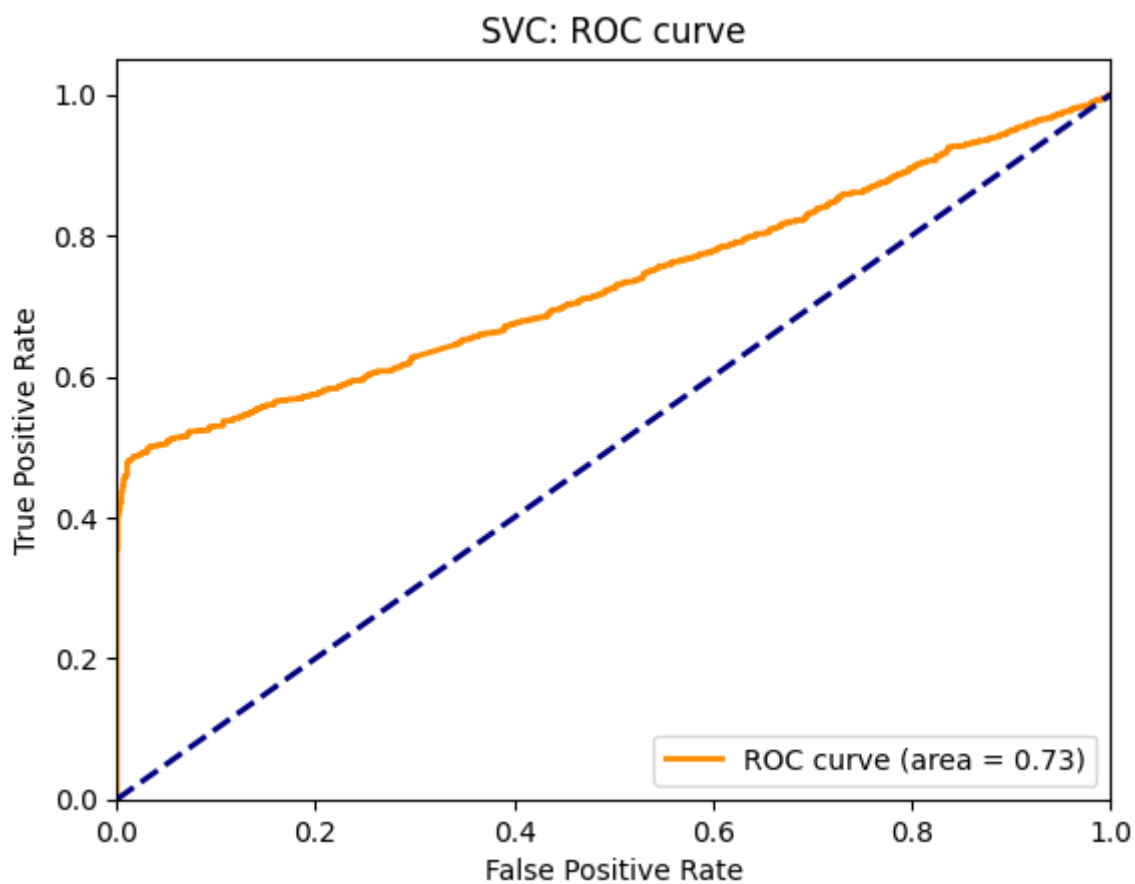
KNeighborsClassifier:

Precision: 0.8

Recall: 0.61

F1-score: 0.69

ROC AUC score: 0.7447099664189403



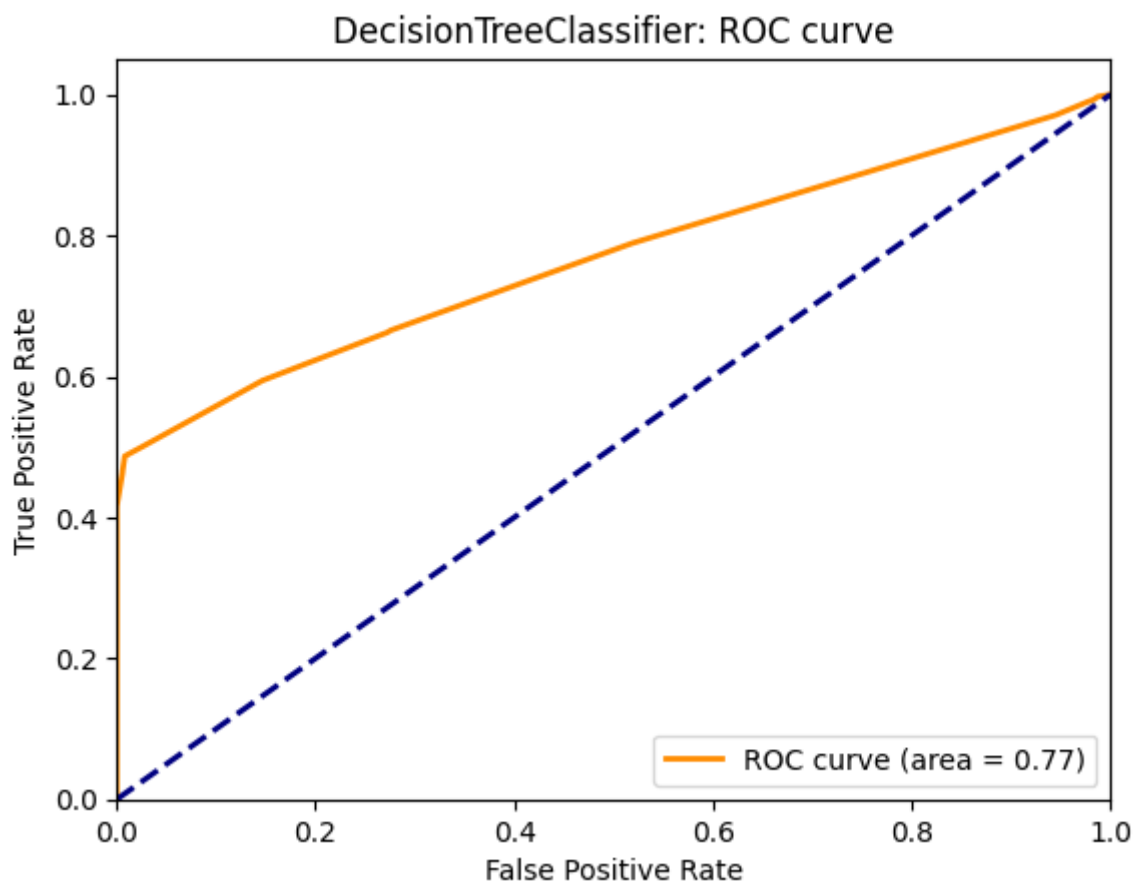
*Рисунок 19 - ROC-кривая модели SVC после поиска гиперпараметров SVC:*

Precision: 0.95

Recall: 0.5

F1-score: 0.66

ROC AUC score: 0.7322713031198976



*Рисунок 20 - ROC-кривая модели Decision Tree после поиска гиперпараметров*

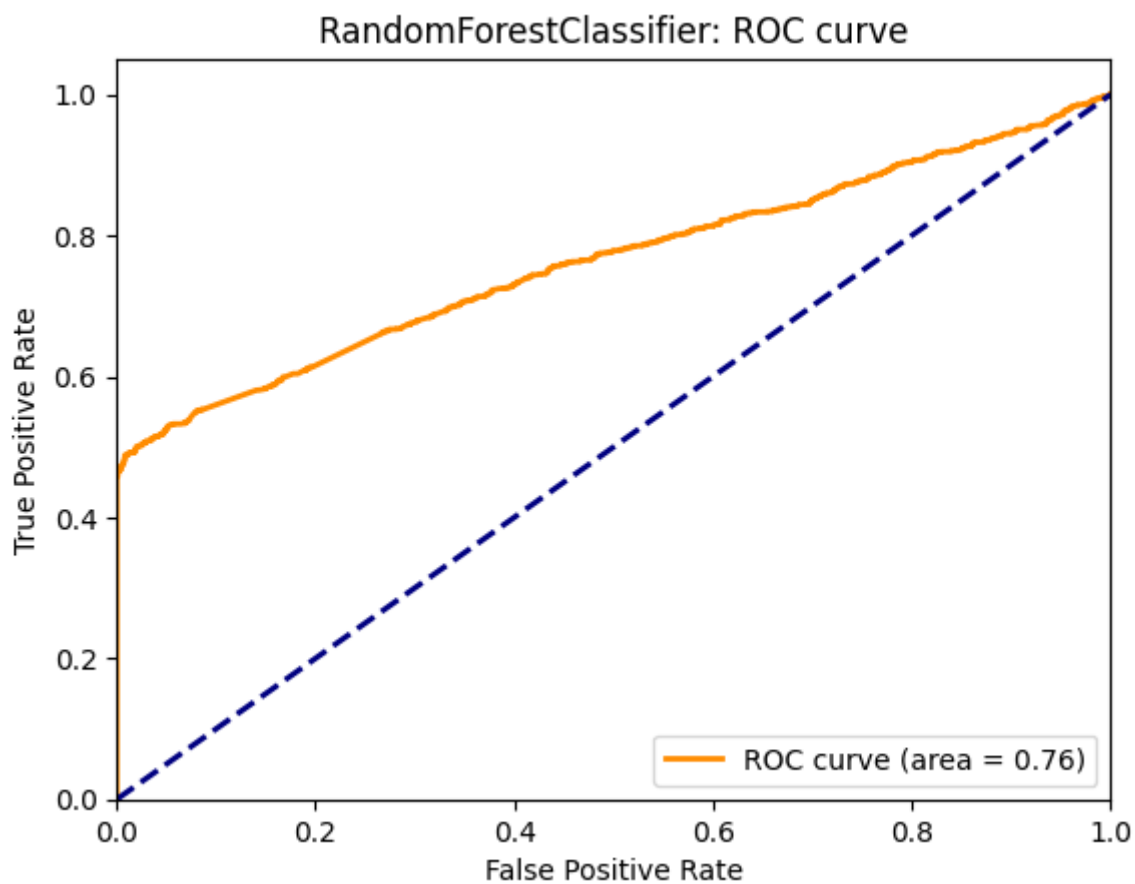
DecisionTreeClassifier:

Precision: 0.97

Recall: 0.5

F1-score: 0.66

ROC AUC score: 0.7658675049385183



*Рисунок 21 - ROC-кривая модели Random Forest после поиска гиперпараметров*

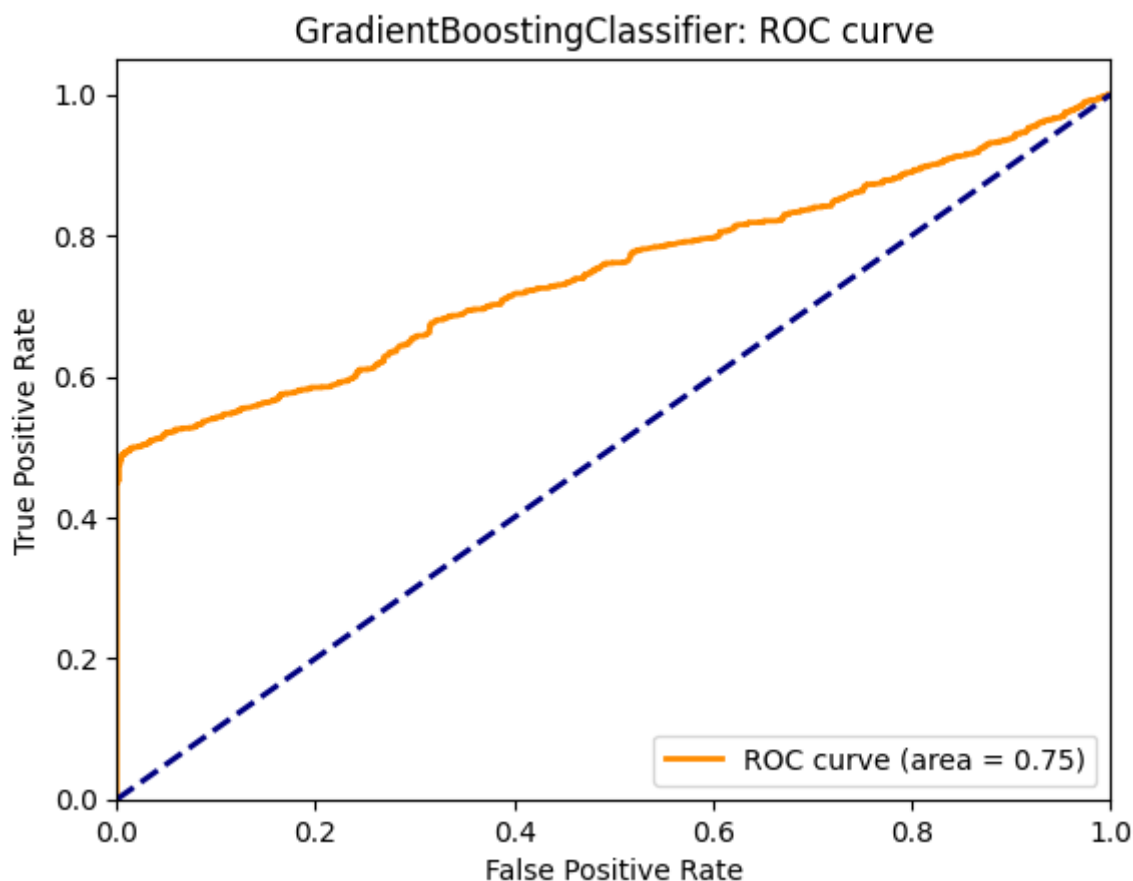
RandomForestClassifier:

Precision: 0.95

Recall: 0.51

F1-score: 0.67

ROC AUC score: 0.7622333784138077



*Рисунок 22 - 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

<b>Decision Tree</b>	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
<b>Random forest</b>	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
<b>Gradient Boosting</b>	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.



## **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 придут 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**

-test на Python для проверки и получения t-статистики // Помощник Python URL:

2. Machine Learning Metrics in simple terms // Medium URL:  
<https://medium.com/analytics-vidhya/machine-learning-metrics-in-simple-terms-d58a9c85f9f6>

порный пример для выполнения проекта по анализу данных. // Jupyter nbviewer

епозиторий курса "Технологии машинного обучения", бакалавриат, 6 семестр. //