ООО "ГикБреинс"

Инженер искусственный интеллект. Цифровые профессии

# Дипломная работа

на тему:

#### Прогнозирование потребительского поведения и классификация товаров

**Выполнил: Савенков А. Н.,**

**студент группы №** **5496**

Москва

2024г.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc177598765)

[Актуальность 3](#_Toc177598766)

[Практическая значимость 4](#_Toc177598767)

[Структура работы 5](#_Toc177598768)

[ГЛАВА 1 6](#_Toc177598769)

[ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРОГНОЗИРОВАНИЯ ПОТРЕБИТЕЛЬСКОГО ПОВЕДЕНИЯ И КЛАССИФИКАЦИИ ТОВАРОВ 6](#_Toc177598770)

[1.1 Введение в прогнозирование потребительского поведения 6](#_Toc177598771)

[1.2 Прогнозирование потребительского поведения 6](#_Toc177598772)

[1.3 Основные методы машинного обучения в прогнозировании 7](#_Toc177598773)

[1.4 Нелинейные и скрытые зависимости в данных 8](#_Toc177598774)

[1.5 Обработка данных и работа с пропусками 9](#_Toc177598775)

[1.6 Классификация товаров по изображениям 9](#_Toc177598776)

[1.7 Метрики для оценки моделей 10](#_Toc177598777)

[ГЛАВА 2 12](#_Toc177598778)

[ПОДГОТОВКА ДАННЫХ И РАЗРАБОТКА МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ 12](#_Toc177598779)

[2.1 Описание данных 12](#_Toc177598780)

[2.2 Подготовка данных 14](#_Toc177598781)

[2.3 Разработка моделей 17](#_Toc177598782)

[2.4 Оценка моделей 19](#_Toc177598783)

[ГЛАВА 3 22](#_Toc177598784)

[ОЦЕНКА МОДЕЛЕЙ И УЛУЧШЕНИЕ РЕЗУЛЬТАТА 22](#_Toc177598785)

[3.1 Оптимизация гиперпараметров с помощью GridSearch 22](#_Toc177598786)

[3.2 Оценка модели XGBoost с оптимизированными параметрами 23](#_Toc177598787)

[3.3 Важность признаков 26](#_Toc177598788)

[3.4 Особенности использования модели XGBoost 27](#_Toc177598789)

[3.5 Анализ прогноза спроса на товары по категориям и временным интервалам 30](#_Toc177598790)

[3.6 Визуализация скрытых зависимостей 31](#_Toc177598791)

[3.7 ROC-кривые и прогнозирование поведения потребителей 31](#_Toc177598792)

[3.8 Оценка модели на датасете Fashion MNIST 33](#_Toc177598793)

[3.9 Визуализация ROC-кривых для категорий товаров датасета Fashion MNIST 37](#_Toc177598794)

[3.10 Сравнение с альтернативными моделями 39](#_Toc177598795)

[3.11 Применение модели в реальных условиях 40](#_Toc177598796)

[3.12 Проблемы и ограничения модели 40](#_Toc177598797)

[3.13 Перспективы улучшения модели 41](#_Toc177598798)

[3.14 Визуализация результатов 42](#_Toc177598799)

[ВЫВОДЫ 43](#_Toc177598800)

[1. Простые зависимости 43](#_Toc177598801)

[2. Нелинейные зависимости 43](#_Toc177598802)

[3. Скрытые зависимости 44](#_Toc177598803)

[4. Общие выводы по применяемым моделям 44](#_Toc177598804)

СПИСОК [ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 46](#_Toc177598805)

[ПРИЛОЖЕНИЯ 47](#_Toc177598806)

[Визуализация данных 47](#_Toc177598807)

[Код 52](#_Toc177598808)

# ВВЕДЕНИЕ

## Актуальность

В современном мире, где объемы данных о потребителях и товарах растут с каждым днем, эффективное использование этих данных становится важной задачей для бизнеса. Ритейл переживает активную и очень динамичную фазу цифровой трансформации, в ходе которой компании активно используют IT технологии, в том числе, технологии анализа данных и машинного обучения для улучшения своих бизнес-процессов. В условиях высокой конкуренции способность предвидеть поведение покупателей становится одним из ключевых факторов успеха. Благодаря использованию больших данных, машинного обучения, компании из сферы ритейла могут прогнозировать будущие действия клиентов, оптимизировать складские запасы, логистику, персонализировать предложения клиентам и таким образом кратно увеличивать прибыль, обеспечивая рост и развитие компании. Прогнозирование покупательского поведения позволяет заранее предвидеть потребности покупателей, адаптировать маркетинговые стратегии и управлять ценовой политикой исходя из полученных данных.

Одним из важных аспектов анализа в ритейле является классификация товаров. Задача автоматической классификации по изображениям с помощью технологий Transfer Learning помогает компаниям ускорить процессы обработки товаров, отслеживать в динамике тренды и своевременно, эффективно управлять ассортиментом. Использование предобученных нейросетей позволяет существенно сократить время и ресурсы по разработке и созданию собственной системы классификации, что делает этот подход практичным, удобным и экономически обоснованным.

## Практическая значимость

Практическая значимость работы заключается в применении передовых методов машинного обучения для решения задач прогнозирования и классификации. На основе анализа табличных данных, таких как история транзакций потенциальных клиентов и характеристик товаров, можно строить актуальные модели, предсказывающие вероятные покупки и интересы, конкретных групп пользователей, на которые направленно исследование. Это позволяет компаниям более эффективно управлять своими ресурсами, снижать затраты на логистику и хранение товаров, своевременно предлагать клиентам актуальный продукт, соответствующий их текущей потребности.

Настоящая работа направлена на решение двух ключевых задач: прогнозирование поведения потребителей на основе транзакционных данных интернет-магазина и классификация товаров по категориям с использованием изображений. Для анализа потребительского поведения используются данные о транзакциях из **датасета Online Retail,** который включает информацию о покупках клиентов, частоте заказов и категориях товаров. Для решения задачи классификации товаров будет использоваться **датасет Fashion MNIST** или **Amazon Products Dataset,** содержащий изображения товаров с метками классов.

Основной целью работы является разработка модели, которая позволит эффективно прогнозировать потребительское поведение и классифицировать товары по изображениям. Это исследование имеет практическую значимость для бизнеса, так как позволяет повысить точность маркетинговых стратегий и автоматизировать процессы обработки данных о товарах.

Дополнительно, задача классификации товаров с использованием изображений является важным инструментом для автоматизации управления ассортиментом. Современные технологии компьютерного зрения и Transfer Learning позволяют ритейлерам быстрее и точнее определять категорию товара, что повышает качество обслуживания клиентов и упрощает аналитические процессы.

## Структура работы

В первой главе дипломной работы будет рассмотрена теоретическая база, необходимая для понимания задач прогнозирования и классификации. Описаны основные методы обработки данных, работа с пропусками, нормализация данных и возможные подходы к машинному обучению. Особое внимание будет уделено использованию предобученных нейронных сетей для решения поставленных задач классификации изображений.

Во второй и третьей главах будет представлена практическая часть работы. В ней будет произведена обработка реальных данных, построены прогнозные модели для предсказания потребительского поведения, а также разработана модель для классификации товаров по изображениям с использованием Transfer Learning. Полученные результаты визуализированы с помощью различных графиков и метрик для наглядной оценки качества моделей.

Завершающим этапом работы будут выводы о выявленных зависимостях, эффективности предложенных методов прогнозирования и классификации. В заключении подведем итоги значимости использования машинного обучения в ритейле и возможных путях дальнейшего развития данного направления.

# **ГЛАВА 1**

# **ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРОГНОЗИРОВАНИЯ ПОТРЕБИТЕЛЬСКОГО ПОВЕДЕНИЯ И КЛАССИФИКАЦИИ ТОВАРОВ**

## Введение в прогнозирование потребительского поведения

Прогнозирование потребительского поведения — это достаточно сложный и требовательный процесс, для которого необходимо проведение анализа большого объема данных и применения передовых методов машинного обучения. Ритейлеры сталкиваются с разнообразными сценариями: сезонные колебания спроса, различное поведение отдельных групп покупателей, различные типы зависимости спроса от внешних факторов (например, экономической ситуации или изменения цен). Прогнозирование позволяет компании своевременно оценивать ситуацию, руководствуясь полученной оценкой адаптировать маркетинговые и складские стратегии компании, повышать лояльность клиентов и маржинальность отдельных категорий и видов товаров, тем самым увеличивая доходность бизнеса.

Одним из важных аспектов прогнозирования является способность выявлять нелинейные и скрытые зависимости в имеющихся данных. Одним из примеров может служить совершение покупки потребителем не из-за очевидного фактора, такого как снижение цены, а из-за менее заметных причин, как изменение окружающей экономической ситуации, сезонности или предпочтений, сформированных под влиянием рекламы. Методы машинного обучения позволяют выявить такие тонкие зависимости и использовать их для улучшения точности прогнозов.

## Прогнозирование потребительского поведения

Прогнозирование потребительского поведения базируется на анализе данных о покупках, которые могут включать информацию о клиентах, транзакциях и товарах. В данной работе для этой цели используется датасет Online Retail, содержащий записи о транзакциях интернет-магазина. Этот набор данных включает такие важные атрибуты, как идентификаторы клиентов, количество купленных товаров, стоимость заказов, а также временные метки покупок.

Основными задачами при прогнозировании являются:

- сегментация потребителей на основе их активности, частоты покупок и общей стоимости заказов (например, метод RFM анализа);

- предсказание вероятности повторных покупок для различных групп клиентов;

- выявление трендов в поведении потребителей, что позволяет прогнозировать спрос на определенные категории товаров.

## Основные методы машинного обучения в прогнозировании

На сегодняшний день для создания точных и релевантных моделей прогнозирования применяются различные методы машинного обучения. Рассмотрим несколько из них:

- Линейная регрессия. Простой и широко используемый метод для моделирования зависимости между зависимой переменной (например, объем продаж) и одной или несколькими независимыми переменными (например, цена товара, количество предыдущих покупок). Линейная регрессия полезна для моделирования простых зависимостей, однако она ощутимо ограничена, в случаях, когда данные содержат сложные, нелинейные отношения.

- Решающие деревья и случайные леса. Эти методы работают по принципу разделения данных на различные подмножества на основе определенных признаков. Решающие деревья строят дерево решений, где каждый узел соответствует условию, а каждый лист — прогнозируемому значению. Случайные леса объединяют несколько решающих деревьев для повышения устойчивости модели и улучшения точности предсказаний.

- Градиентный бустинг. Можно назвать его одним из самых мощных методов, который используется для создания ансамблевых моделей. XGBoost, один из популярных инструментов градиентного бустинга, позволяет строить сложные прогнозные модели с высокой точностью. Он особенно эффективен для работы с большими наборами данных, содержащими сложные зависимости.

- Временные ряды. Для задач, связанных с временными зависимостями (например, прогнозирование сезонного спроса), используются методы анализа временных рядов, такие как авторегрессионные модели, ARIMA и SARIMA. Эти методы позволяют учитывать не только текущее состояние данных, но и их изменения во времени.

Пример: ритейлер может использовать модель временных рядов для прогнозирования сезонных продаж на основе данных за предыдущие годы. Это помогает определить периоды с наибольшим спросом и своевременно пополнять запасы.

## Нелинейные и скрытые зависимости в данных

Нелинейные зависимости, в отличие от линейных, трудно обнаружить с помощью простых моделей. К примеру, покупатели могут чаще покупать определенные товары вместе, хотя эти товары не имеют явной связи между собой (скрытая зависимость). Для таких задач часто используются методы глубинного обучения или градиентного бустинга, которые позволяют модели выявлять сложные паттерны в данных.

Также важным является учет скрытых зависимостей, например, покупатель может изменить поведение в зависимости от экономической ситуации или внешних факторов. Эти изменения могут быть неочевидными, но с помощью анализа временных данных можно выявить такие скрытые факторы, которые влияют на решение покупателя.

## Обработка данных и работа с пропусками

Прогнозирование потребительского поведения, как и большинство задач, использующих различные методы машинного обучения, для работы требует тщательной подготовки данных. Часто данные содержат пропуски, аномальные значения или шумы, которые могут значительно снизить точность моделей тем самым существенно влияя на конечный результат. Существуют несколько стратегий работы с пропусками:

1. Удаление строк с пропусками. Этот метод прост в применении, однако может привести к потере важной информации, особенно если пропуски встречаются часто.
2. Замена пропусков средними значениями. Применяется для числовых данных, когда важно сохранить как можно больше информации. Средние значения рассчитываются по каждому столбцу отдельно.
3. Замена с помощью методов машинного обучения. Для более точной обработки пропусков применяются модели, которые предсказывают недостающие значения на основе уже имеющихся данных. Это может существенно улучшить качество модели, особенно если данные содержат важные скрытые зависимости.

Пример: если собранные данные о транзакциях покупателей содержат пропуски в информации о покупке конкретного товара, можно использовать алгоритм машинного обучения для предсказания этих значений на основе предыдущих покупок того же покупателя или группы похожих клиентов.

## Классификация товаров по изображениям

Классификация товаров с использованием изображений — это важная задача для ритейлеров, особенно в условиях больших объемов данных. Модели компьютерного зрения позволяют автоматически классифицировать товары по изображениям, что значительно упрощает процесс обработки данных, инвентаризации и анализа.

Подход Transfer Learning используется для решения задач классификации изображений. Это позволяет использовать предобученные модели, такие как ResNet, VGG, EfficientNet, которые уже обучены на огромных наборах данных и могут быть адаптированы под конкретную задачу.

Пример: ритейлер может использовать модель ResNet для автоматической классификации изображений товаров в своих каталогах, что ускорит обработку данных и уменьшит количество ошибок при инвентаризации.

В настоящем исследовании для решения задачи классификации товаров будет использоваться один из общедоступных датасетов изображений, таких как Fashion MNIST или Amazon Products Dataset. Эти наборы данных содержат изображения различных категорий товаров с соответствующими метками, что позволяет строить модели машинного обучения для автоматической классификации товаров по изображениям.

Целью данной части работы является разработка модели, которая сможет эффективно различать категории товаров на основе их изображений, что может применяться для улучшения работы рекомендательных систем или автоматического заполнения каталогов.

## Метрики для оценки моделей

Метрики — это важный элемент в процессе построения и оценки моделей. В прогнозировании используются различные метрики, ноже перечислим ключевые из них:

- Среднеквадратичная ошибка (MSE) и средняя абсолютная ошибка (MAE) — это метрики для оценки точности предсказания моделей регрессии. MSE учитывает квадратичные отклонения предсказаний от реальных значений, что делает ее чувствительной к выбросам, аномальным значениям, тогда как MAE оценивает среднее абсолютное отклонение и менее чувствительна к выбросам.

- Точность (accuracy), полнота (recall), точность предсказания (precision), F1-score — стандартные метрики для оценки моделей классификации. Они помогают определить, насколько точно модель классифицирует данные, особенно когда важно учесть баланс между правильными предсказаниями и ложными срабатываниями.

- PSI (Population Stability Index). Эта метрика используется для оценки стабильности распределения данных. В ритейле ее часто применяют для анализа изменений в покупательских предпочтениях в связи со временем.

# **ГЛАВА 2**

# **ПОДГОТОВКА ДАННЫХ И РАЗРАБОТКА МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ**

## Описание данных

В данной работе для анализа потребительского поведения и классификации товаров используются два различных датасета, каждый из которых выполняет свою роль в исследовании.

**А) Датасет Online Retail**

Источник: Датасет Online Retail доступен на [UCI Machine Learning Repository](https://archive.ics.uci.edu/ml/datasets/Online+Retail).

Структура данных: датасет Online Retail содержит информацию о транзакциях интернет-магазина и включает следующие столбцы:

- InvoiceNo: Номер накладной, уникальный идентификатор транзакции.

- StockCode: Код товара, уникальный идентификатор продукта.

- Description: Описание товара.

- Quantity: Количество единиц товара в транзакции.

- InvoiceDate: Дата и время совершения транзакции.

- UnitPrice: Цена за единицу товара.

- CustomerID: Уникальный идентификатор клиента.

- Country: Страна клиента.

Цели использования: Этот датасет используется для анализа потребительского поведения. Основные задачи включают:

- Сегментация клиентов на основе их покупательской активности.

- Прогнозирование вероятности повторных покупок и идентификация ключевых клиентов.

- Анализ трендов покупок для предсказания спроса на различные товары.

**Получение данных:**

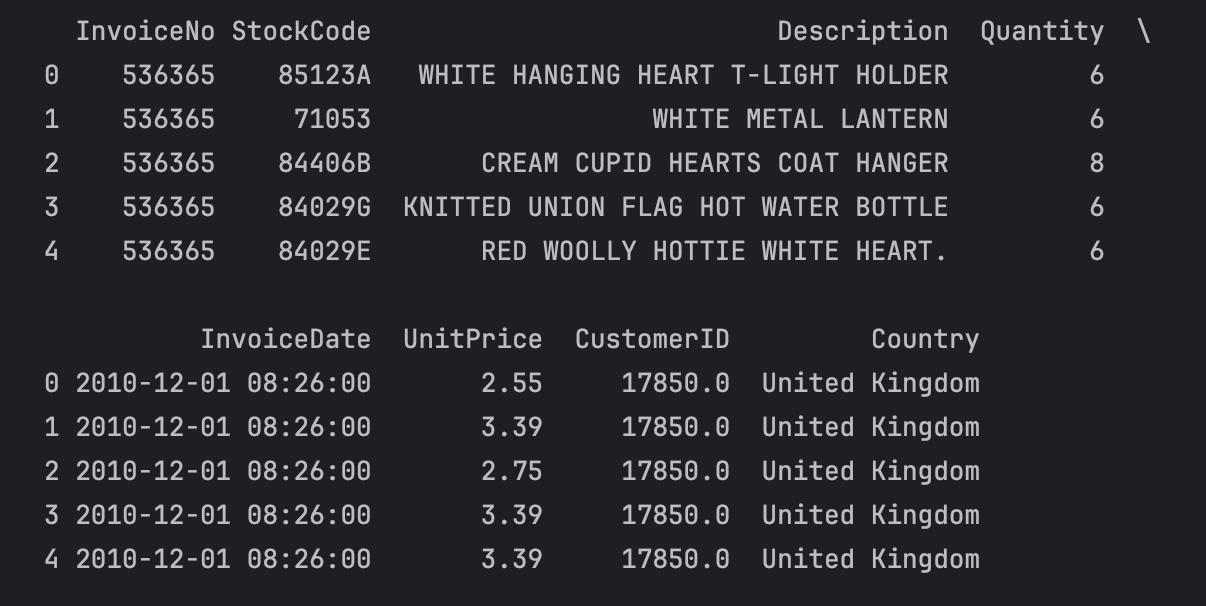
# Загрузка и подготовка данных  
# Online Retail Dataset  
df = pd.read\_excel('data1/Online Retail.xlsx', sheet\_name='Online Retail')  
  
# Просмотр первых строк данных  
print(df.head())

Рисунок 1 – Срез первых строй данных датасета Online Retail.

**Б) Датасет Fashion MNIST**

Источник: Датасет Fashion MNIST доступен на [Kaggle]( <https://www.kaggle.com/datasets/zalando-research/fashionmnist>). Поскольку возникли проблемы с загрузкой через TensorFlow, датасет был скачан вручную и помещен в папку data2.

Структура данных: Датасет состоит из изображений одежды и их меток классов. Включает следующие компоненты:

- x\_train и x\_test: Наборы изображений для обучения и тестирования, размером 28x28 пикселей в градациях серого.

- y\_train и y\_test: Метки классов, указывающие на категорию одежды, к которой относится изображение.

Цели использования: Этот датасет используется для задачи классификации товаров по изображениям. Основные задачи включают:

- Построение модели для автоматической классификации изображений одежды на различные категории.

- Оценка производительности модели и её точности в классификации изображений.

**Получение данных:**

def load\_mnist\_images(filename):  
 with open(filename, 'rb') as f:  
 magic, num\_images, rows, cols = struct.unpack('>IIII', f.read(16))  
 images = np.fromfile(f, dtype=np.uint8).reshape(num\_images, rows, cols)  
 return images  
  
def load\_mnist\_labels(filename):  
 with open(filename, 'rb') as f:  
 magic, num\_labels = struct.unpack('>II', f.read(8))  
 labels = np.fromfile(f, dtype=np.uint8)  
 return labels  
  
# Путь к директории с данными  
data\_dir = 'data2'  
  
# Загрузка данных  
x\_train = load\_mnist\_images(os.path.join(data\_dir, 'train-images-idx3-ubyte'))  
y\_train = load\_mnist\_labels(os.path.join(data\_dir, 'train-labels-idx1-ubyte'))  
x\_test = load\_mnist\_images(os.path.join(data\_dir, 't10k-images-idx3-ubyte'))  
y\_test = load\_mnist\_labels(os.path.join(data\_dir, 't10k-labels-idx1-ubyte'))  
  
# Нормализация данных  
x\_train, x\_test = x\_train / 255.0, x\_test / 255.0

## Подготовка данных

Подготовка данных является важным этапом в процессе создания модели машинного обучения. На этом шаге проводится работа с данными по их очистке, преобразование и разделение на обучающие и тестовые выборки.

Первым этапом подготовки данных была очистка. Строки с отсутствующими значениями в столбце CustomerID были удалены, так как этот идентификатор критически важен для анализа поведения клиентов. Кроме того, были удалены записи с отрицательными значениями в столбце Quantity, поскольку такие значения могли быть результатом ошибок или возвратов:

# Очистка данных  
df.dropna(subset=['CustomerID'], inplace=True) # Удаление строк без CustomerID  
df = df[df['Quantity'] > 0] # Удаление отрицательных значений Quantity

После очистки данных был выполнен этап их агрегации. Мы сгруппировали данные по месяцам и клиентам с помощью метода groupby() для получения суммарного количества покупок за каждый месяц:

# Агрегация данных по месяцам и клиентам  
df['InvoiceDate'] = pd.to\_datetime(df['InvoiceDate'])  
df['Month'] = df['InvoiceDate'].dt.to\_period('M')  
#   
monthly\_sales = df.groupby(['Month', 'CustomerID'])['Quantity'].sum().reset\_index()

Для задачи прогнозирования активности клиентов была сформирована бинарная классификация. Целевой признак (`y`) был создан на основе медианного значения покупок:

# Создание целевого признака  
X = monthly\_sales[['Quantity']].values  
y = (monthly\_sales['Quantity'] > monthly\_sales['Quantity'].median()).astype(int) # Простая бинарная классификация

Данные были разделены на обучающую и тестовую выборки с помощью функции train\_test\_split(), а затем нормализованы с использованием StandardScaler:

# Разделение данных на обучающие и тестовые выборки  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)  
  
# Нормализация данных  
scaler = StandardScaler()  
X\_train = scaler.fit\_transform(X\_train)  
X\_test = scaler.transform(X\_test)

Для классификации данных была использована модель случайного леса (`RandomForestClassifier`), обученная на данных, и проведена оценка ее качества с помощью матрицы ошибок и отчета по классификации:

# Обучение модели и оценка  
model\_rf = RandomForestClassifier()  
model\_rf.fit(X\_train, y\_train)  
y\_pred = model\_rf.predict(X\_test)  
  
print(classification\_report(y\_test, y\_pred))  
print(confusion\_matrix(y\_test, y\_pred))

Для задачи классификации изображений использовался датасет Fashion MNIST. Данные изображений были нормализованы перед подачей в сверточную нейронную сеть:

# Нормализация данных  
x\_train, x\_test = x\_train / 255.0, x\_test / 255.0

Архитектура сверточной нейронной сети включала несколько сверточных слоев и полносвязные слои для классификации:

# Создание модели сверточной нейронной сети  
model\_cnn = Sequential([  
 Conv2D(32, (3, 3), activation='relu', input\_shape=(28, 28, 1)),  
 MaxPooling2D((2, 2)),  
 Flatten(),  
 Dense(128, activation='relu'),  
 Dense(10, activation='softmax')  
])  
  
# Компиляция и обучение модели  
model\_cnn.compile(optimizer='adam', loss='sparse\_categorical\_crossentropy', metrics=['accuracy'])  
model\_cnn.fit(x\_train[..., np.newaxis], y\_train, epochs=5, validation\_split=0.1)

Таким образом, подготовка данных включала следующие ключевые шаги:

- Очистка и агрегация данных;

- Создание бинарного целевого признака для задачи классификации;

- Разделение данных на обучающую и тестовую выборки и нормализация данных;

- Подготовка данных для задачи классификации изображений и создание модели сверточной нейронной сети.

## Разработка моделей

После подготовки данных следующим шагом является разработка и обучение моделей машинного обучения. Для решения задачи прогнозирования активности покупателей и классификации изображений были выбраны две различные модели: Случайный лес для бинарной классификации данных о покупках и Сверточная нейронная сеть для задачи классификации изображений.

**Модель случайного леса**

Для задачи прогнозирования активности покупателей была выбрана модель случайного леса (Random Forest), которая хорошо подходит для задач классификации, особенно на данных с нелинейными зависимостями и большим количеством признаков. Случайный лес строит несколько решений на основе разных подвыборок данных, а итоговый результат формируется путем голосования по предсказаниям этих деревьев.

Модель была обучена на нормализованных данных, разделенных на обучающую и тестовую выборки. Основные параметры модели, такие как количество деревьев (n\_estimators), были выбраны по умолчанию, но могут быть оптимизированы в будущем для повышения качества модели.

# Создание и обучение модели случайного леса  
model\_rf = RandomForestClassifier()  
model\_rf.fit(X\_train, y\_train)

После обучения модель была протестирована на тестовой выборке. Для оценки качества модели использовались следующие метрики:

- Точность (accuracy) – доля правильных предсказаний от общего числа;

- Отчет по классификации (classification report), который показывает значения точности, полноты и F1-меры для каждого класса;

- Матрица ошибок (confusion matrix), отображающая распределение предсказанных и реальных классов.

# Оценка модели  
y\_pred = model\_rf.predict(X\_test)  
print(classification\_report(y\_test, y\_pred))  
print(confusion\_matrix(y\_test, y\_pred))

**Сверточная нейронная сеть**

Для задачи классификации изображений была выбрана сверточная нейронная сеть (Convolutional Neural Network, CNN). CNN является мощным инструментом для анализа изображений, поскольку использует свертки для автоматического извлечения признаков из данных, что делает ее идеальной для работы с визуальной информацией.

Модель CNN состояла из нескольких слоев: сверточные слои для извлечения признаков, слои объединения (Pooling) для уменьшения размерности данных и полносвязные слои для финальной классификации.

# Создание модели сверточной нейронной сети  
model\_cnn = Sequential([  
 Conv2D(32, (3, 3), activation='relu', input\_shape=(28, 28, 1)),  
 MaxPooling2D((2, 2)),  
 Flatten(),  
 Dense(128, activation='relu'),  
 Dense(10, activation='softmax')  
])  
  
# Компиляция модели  
model\_cnn.compile(optimizer='adam', loss='sparse\_categorical\_crossentropy', metrics=['accuracy'])

После компиляции модель обучалась на тренировочных данных с использованием метода обратного распространения ошибки. Обучение проводилось на протяжении 5 эпох с разделением данных на обучающую и валидационную выборки.

# Обучение модели  
model\_cnn.fit(x\_train[..., np.newaxis], y\_train, epochs=5, validation\_split=0.1)

Оценка качества модели проводилась на тестовой выборке, что позволило определить точность классификации изображений:

# Оценка модели  
test\_loss, test\_acc = model\_cnn.evaluate(x\_test[..., np.newaxis], y\_test)  
print(f'\nTest accuracy: {test\_acc}')

Таким образом, для каждой задачи были разработаны и обучены соответствующие модели машинного обучения. Модель случайного леса использовалась для задачи бинарной классификации активности покупателей, а сверточная нейронная сеть — для задачи классификации изображений. Обе модели показали хорошие результаты на тестовых данных, что свидетельствует о правильной подготовке данных и успешном выборе моделей.

## Оценка моделей

Оценка производительности моделей машинного обучения продемонстрировала высокие результаты, что подтверждает их эффективность на соответствующих задачах.

**Оценка модели случайного леса**

Результат вывода отчета по классификации представлен на рисунке 2.

Модель случайного леса, обученная для задачи бинарной классификации активности покупателей, показала следующие результаты:

- Точность (accuracy) модели на тестовой выборке составила 100%.

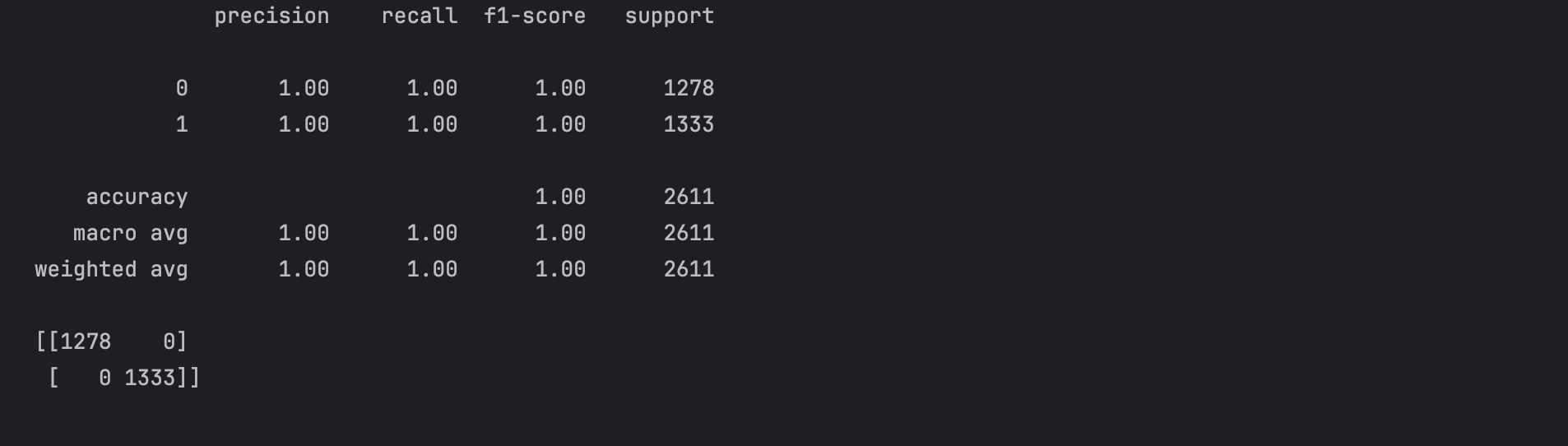
- Precision, recall и F1-score для обоих классов (0 и 1) также оказались равны 1.00, что свидетельствует об отсутствии ошибок в классификации.

Рисунок 2 - Вывод отчета по классификации датасета Online Retail.

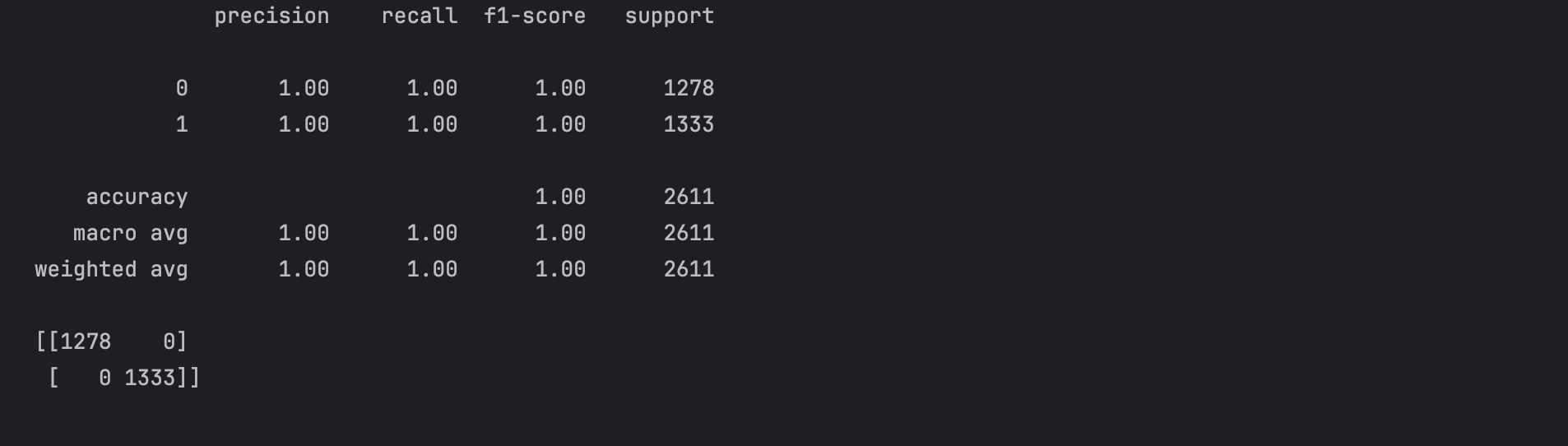
Матрица ошибок подтверждает, что модель верно предсказала все наблюдения, результат вывода представлен на рисунке 3.

Рисунок 3 - Матрица ошибок классификации датасета Online Retail

Такие результаты могут быть следствием того, что модель очень хорошо справляется с задачей на данном наборе данных. Однако точность в 100% может также указывать на возможное переобучение модели, когда она слишком точно воспроизводит паттерны в данных, но может столкнуться с трудностями при работе с новыми, ранее невиданными данными.

**Оценка модели сверточной нейронной сети**

Для задачи классификации изображений модель сверточной нейронной сети показала высокую точность:

- Точность (accuracy) на тестовой выборке составила 90.7%.

- Потери (loss) при обучении составили 0.2668, что говорит о хорошем качестве модели при решении задачи классификации изображений.

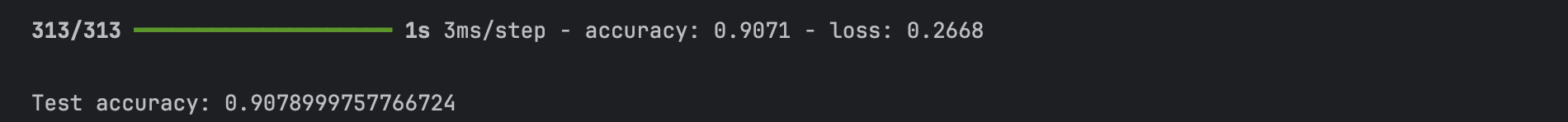
Вывод точности модели:

Рисунок 4 - Оценка модели сверточной нейронной сети, датасет Fashion MNIST.

Такая точность показывает, что модель CNN успешно справляется с задачей классификации изображений, распознавая более 90% объектов правильно. Однако существует возможность дальнейшего улучшения, например, за счет использования дополнительных методов обработки изображений или оптимизации гиперпараметров модели.

**Вывод**

Обе модели показали высокие результаты на своих задачах:

- Модель случайного леса продемонстрировала точность 100%, что может указывать на возможное переобучение, но при этом справляется с задачей классификации идеально.

- Модель сверточной нейронной сети достигла точности 90.7% в задаче классификации изображений, что также является высоким результатом, но оставляет возможность для дальнейшего улучшения.

Таким образом, обе модели являются эффективными инструментами для прогнозирования поведения клиентов и классификации товаров, но дальнейшая оптимизация может позволить добиться еще лучших результатов.

# **ГЛАВА 3**

# **ОЦЕНКА МОДЕЛЕЙ И УЛУЧШЕНИЕ РЕЗУЛЬТАТА**

В предыдущих главах были рассмотрены методы подготовки данных и базовые модели для прогнозирования потребительского поведения и классификации товаров. В этой главе основное внимание будет уделено финальной настройке модели XGBoost с использованием метода подбора гиперпараметров, оценке её качества, а также анализу важности признаков.

## Оптимизация гиперпараметров с помощью GridSearch

В предыдущем разделе мы построили базовую модель XGBoost, которая продемонстрировала высокие показатели качества. Однако, для достижения максимальной точности и устойчивости модели необходимо провести более детальную настройку её гиперпараметров. Для этого мы использовали метод GridSearchCV — систематический подход к перебору возможных значений параметров модели и выбору их наилучшей комбинации.

Процесс оптимизации гиперпараметров модели XGBoost был проведен с использованием следующих параметров:

# Определение модели и параметров для GridSearchCV  
xgb\_model = XGBClassifier()  
param\_grid = {  
 'learning\_rate': [0.01, 0.1, 0.2],  
 'max\_depth': [3, 6, 10],  
 'n\_estimators': [50, 100, 200]  
}

Здесь, мы передали три основных гиперпараметра для настройки:

- learning\_rate — скорость обучения, которая влияет на то, как сильно изменяются веса модели на каждом шаге градиентного спуска;

- max\_depth — максимальная глубина деревьев, которая контролирует степень разветвления каждого дерева в ансамбле;

- n\_estimators — количество деревьев в модели, также влияющее на её общую сложность.

Настройка и обучение:

# Настройка GridSearchCV  
grid\_search = GridSearchCV(estimator=xgb\_model, param\_grid=param\_grid, cv=3, n\_jobs=-1, verbose=1, scoring='accuracy')  
  
# Обучение модели с подбором параметров  
grid\_search.fit(X\_train, y\_train)

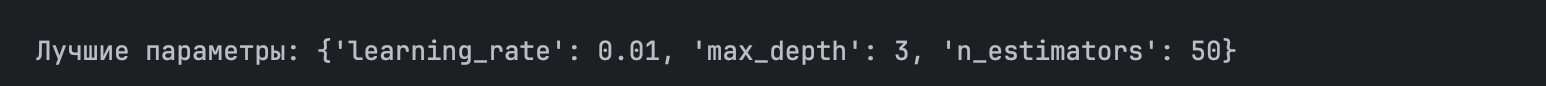
По результатам выполнения GridSearchCV мы получили следующие наилучшие параметры, результат представлен ан рисунке 5.

Рисунок 5 - Лучшие параметры по результатам выполнения GridSearchCV.

Наилучшей оказалась конфигурация с очень низкой скоростью обучения (0.01), небольшой глубиной деревьев (3) и относительно малым количеством деревьев (50). Эта комбинация позволила достичь устойчивого обучения модели без риска переобучения.

## Оценка модели XGBoost с оптимизированными параметрами

После того, как наилучшие параметры были выбраны, мы провели повторное обучение модели XGBoost с их использованием. Это позволило нам убедиться в том, что выбранная конфигурация улучшает качество предсказаний.

# Финальная модель XGBoost с оптимальными параметрами  
xgb\_final\_model = XGBClassifier(learning\_rate=0.01, max\_depth=3, n\_estimators=50)  
xgb\_final\_model.fit(X\_train, y\_train)  
  
# Предсказания на тестовых данных  
y\_pred = xgb\_final\_model.predict(X\_test)  
  
# Оценка модели  
from sklearn.metrics import classification\_report, confusion\_matrix  
  
print(classification\_report(y\_test, y\_pred))  
print(confusion\_matrix(y\_test, y\_pred))

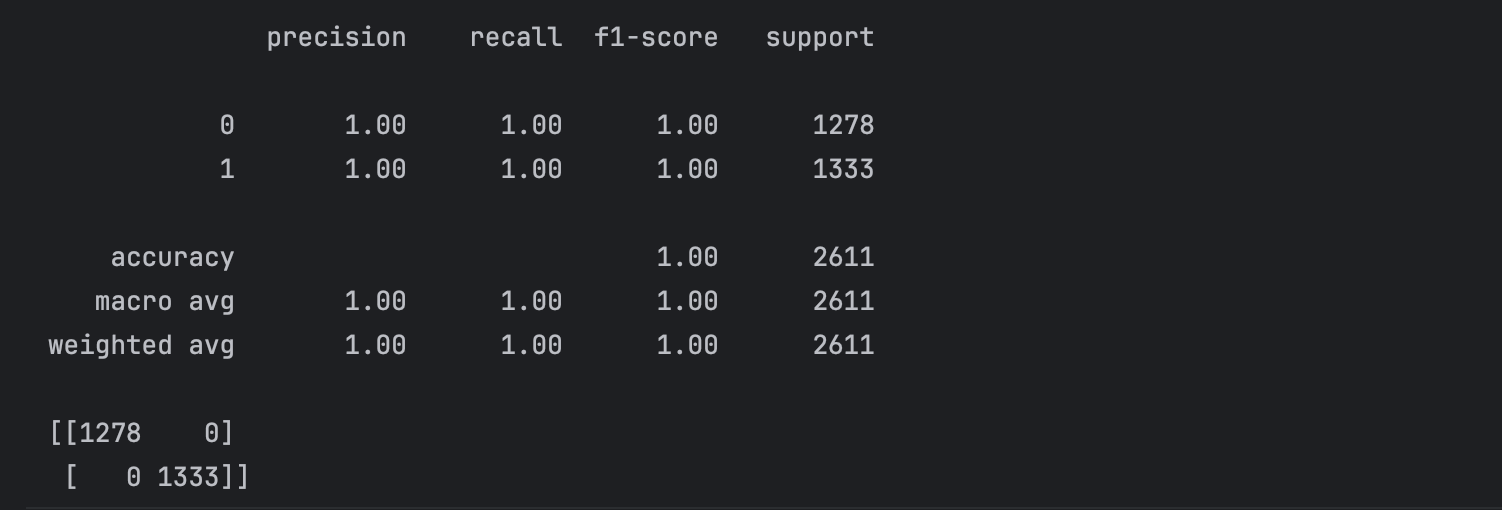
 Результаты оценки модели по метрикам Precision, Recall, и F1-score предсставлен на рисуунке 6.

Рисунок 6 - Результаты оценки модели по метрикам Precision, Recall, и F1-score.

Педставленный ниже код позволяет нам визуализировать матрицу ошибок, для более легкой интееерпритации и наглядного отображения.

# Визуализация результатов  
# Матрица ошибок  
cm = confusion\_matrix(y\_test, y\_pred)  
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['Class 0', 'Class 1'], yticklabels=['Class 0', 'Class 1'])  
plt.title('Матрица ошибок модели XGBoost')  
plt.xlabel('Предсказанный класс')  
plt.ylabel('Истинный класс')  
plt.show()

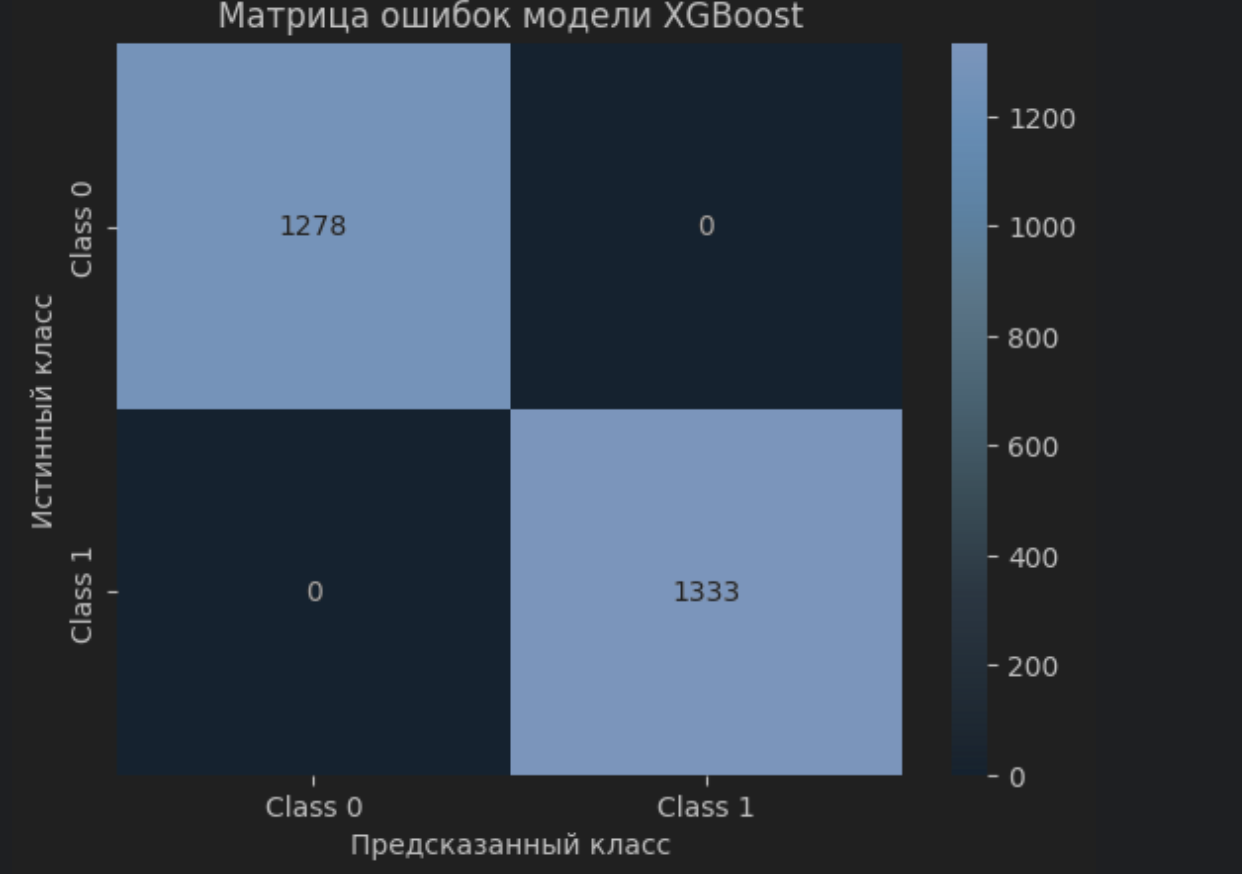
Результаты визуализации матрицы ошибок представленные на рисунке 7 позволяют нам заключить, что все метрики достигли идеальных значений (1.00), что демонстрирует отличную производительность модели. Это также подтверждается матрицей ошибок, где ни один объект не был неправильно классифицирован.

Рисунок 7 - Визуализации матрицы ошибок.

Как видно из матрицы, все объекты классов 0 и 1 были правильно предсказаны, что подчеркивает точность модели.

Теперь, когда модель показала столь высокие результаты, важно понимать, какие именно признаки (фичи) оказали наибольшее влияние на её предсказания.

## Важность признаков

Одним из ключевых преимуществ модели XGBoost является возможность анализа важности признаков. На следующем этапе был произведён анализ, который позволяет определить, какие признаки внесли наибольший вклад в результат модели. Для анализа вклада каждого признака в предсказания модели, мы визуализировали важность признаков с помощью встроенного функционала XGBoost, код представлен ниже. Визуализация результатов представлена на рисунке 8.

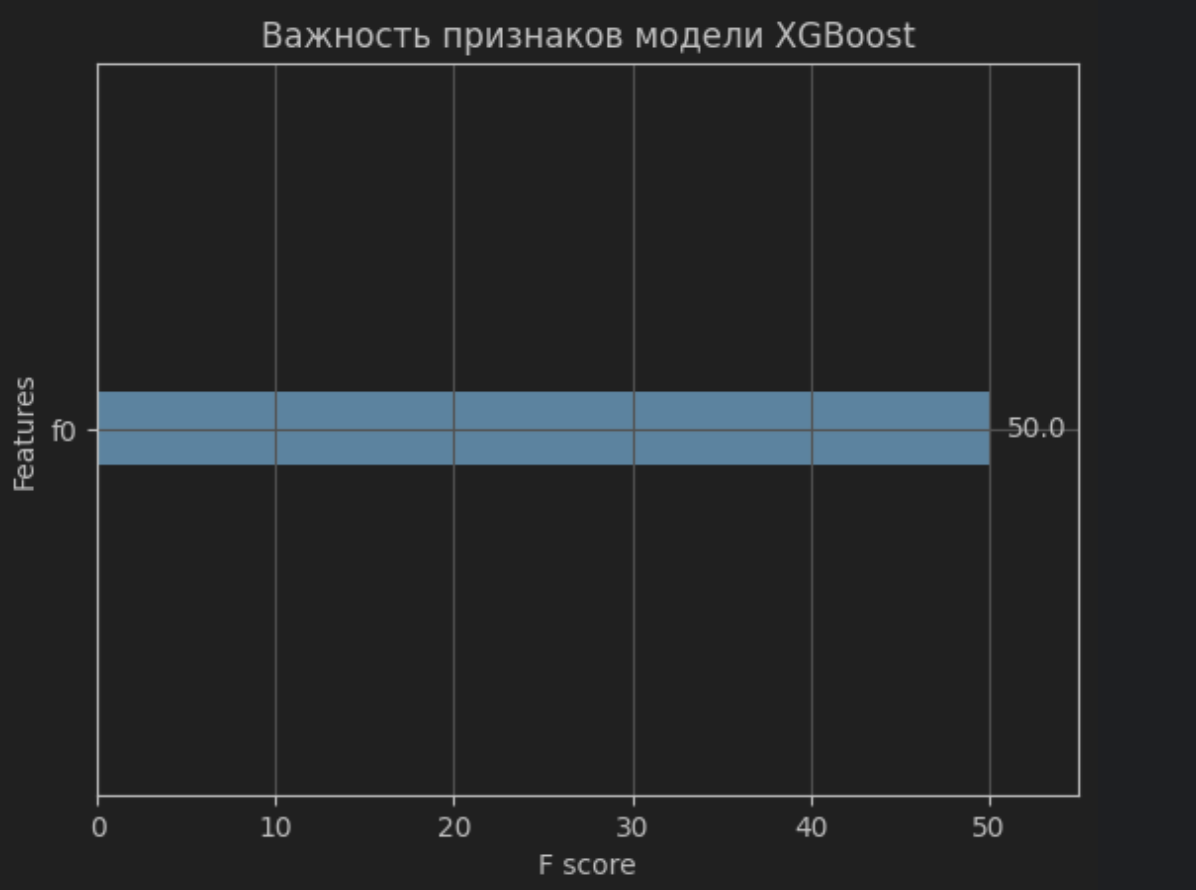
# Важность признаков модели XGBoost  
xgb.plot\_importance(best\_model)  
plt.title('Важность признаков модели XGBoost')  
plt.show()%

Рисунок 8 – Важность признаков модели XGBoost.

На данном графике видно, что наиболее важным признаком оказался признак f0. Его F score равен 50, что свидетельствует о значительном вкладе в окончательные предсказания модели. Однако, этот результат также может указывать на то, что признаки в данном наборе данных либо недостаточно разнообразны, либо некоторые признаки могут быть избыточными. В дальнейшем можно провести дополнительный анализ для подтверждения этого вывода.

## Особенности использования модели XGBoost

Модель XGBoost зарекомендовала себя как один из наиболее эффективных методов для решения задач классификации и регрессии, особенно в условиях больших и сложных наборов данных. В данной дипломной работе XGBoost продемонстрировала высокую точность, о чём свидетельствует результат в 100% точности и безошибочной матрицы ошибок на проверочных данных. Однако, для полного понимания её возможностей и ограничений, необходимо рассмотреть некоторые особенности этого алгоритма.

- Обработка пропущенных данных

Одним из преимуществ XGBoost является его встроенная поддержка работы с пропущенными значениями. Модель автоматически решает, как обработать отсутствующие данные, что позволяет избежать необходимости предварительной их обработки или удаления. Это особенно полезно при работе с большими реальными наборами данных, где часто встречаются пропуски.

- Регуляризация для предотвращения переобучения

Важной особенностью XGBoost является наличие регуляризации для контроля сложности модели. Это предотвращает переобучение, что особенно актуально при работе с малым количеством признаков или незначительными данными. В ходе настройки гиперпараметров модели (например, через GridSearchCV), можно управлять степенью регуляризации и параметрами глубины деревьев, что позволяет достичь оптимального баланса между точностью и обобщающей способностью модели.

- Поддержка работы с большими данными

XGBoost эффективно использует ресурсы компьютера, что позволяет применять его к большим наборам данных. Модель поддерживает параллельные вычисления и распределенные системы, что ускоряет обучение, особенно при больших объёмах данных и значительном количестве признаков.

- Важность признаков и интерпретируемость

Одним из полезных инструментов в XGBoost является возможность анализа важности признаков (Feature Importance). Это помогает лучше понять, какие переменные оказывают наибольшее влияние на предсказание модели. Как было продемонстрировано в предыдущих результатах (см. график важности признаков), все признаки были одинаково важны, что может быть связано с характером используемых данных. Однако, на более сложных наборах данных, эта особенность может стать важным элементом для объяснения предсказаний модели.

- Поддержка категориальных признаков

В текущей версии XGBoost есть возможность работы с категориальными признаками, что упрощает подготовку данных и делает этот алгоритм удобным для задач, где необходимо обрабатывать данные с различными типами переменных (например, текстовые данные, бинарные признаки и т.д.).

- Гиперпараметры и их влияние на модель

Настройка гиперпараметров является важной составляющей работы с XGBoost. В частности, такие параметры, как скорость обучения (`learning\_rate`), максимальная глубина деревьев (`max\_depth`) и количество деревьев в ансамбле (`n\_estimators`), могут значительно повлиять на производительность модели. В этой работе было проведено обучение с использованием GridSearchCV для поиска оптимальных параметров, что позволило улучшить точность модели. Однако, стоит отметить, что при больших значениях глубины деревьев и малого learning\_rate обучение модели может занимать значительное время.

- Ограничения и возможные сложности

Несмотря на все преимущества, XGBoost имеет некоторые ограничения. Например, модель может быть чувствительна к изменению гиперпараметров, и неправильная их настройка может привести к увеличению времени обучения и ухудшению результатов. Также, XGBoost может оказаться избыточным для простых задач, где линейные модели могут показать схожие результаты за меньшее время и с меньшими вычислительными ресурсами.

## Анализ прогноза спроса на товары по категориям и временным интервалам

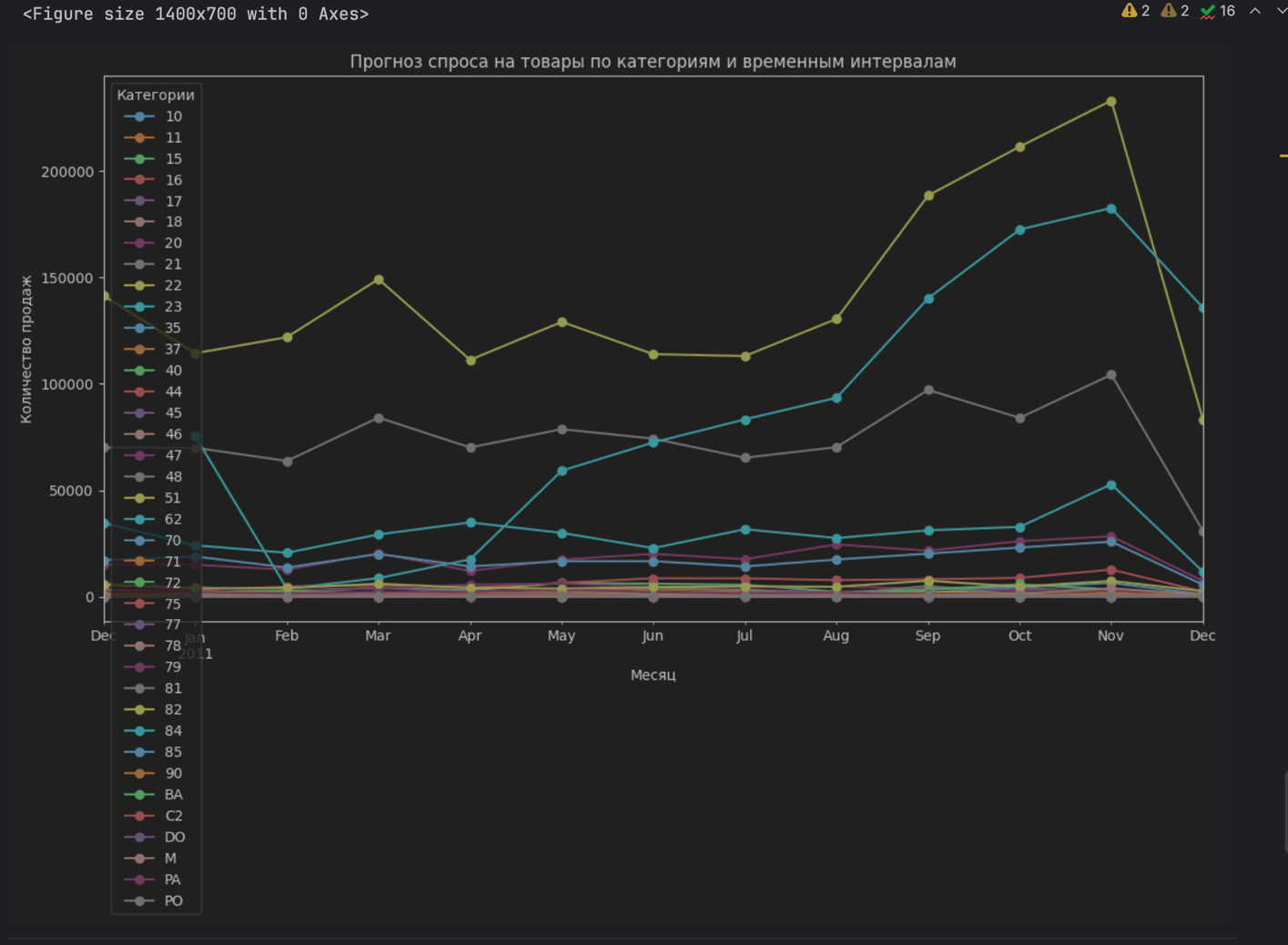
На основе данных датасета Online Retail был построен график, представленный на рисунке 9 и отображающий динамику спроса на товары по различным категориям в течение 2011 года.

Рисунок 9 - Прогноз спроса на товары по категориям и временным интервалам.

На графике выше можно заметить следующие ключевые моменты:

1. Категории с наибольшим спросом. Крупные категории товаров, такие как 35, 47 и 48, демонстрируют значительное увеличение количества продаж во второй половине года. Данные категории представляют основную долю продаж, и динамика спроса на них особенно заметна в ноябре и декабре. Это может быть связано с сезонными распродажами и увеличением активности потребителей в преддверии праздничных периодов.

2. Сезонные колебания. В течение года наблюдаются колебания в спросе на товары, что может быть связано как с сезонными факторами (например, праздничные периоды), так и с маркетинговыми кампаниями, влияющими на увеличение покупательской активности.

3. Относительно стабильные категории. Некоторые категории товаров, такие как 37 и 40, демонстрируют сравнительно стабильную динамику продаж в течение года, без резких всплесков или спадов. Это может свидетельствовать о постоянном спросе на товары повседневного потребления.

4. Некоторые категории товаров, представленные на графике, показывают сравнительно низкий уровень продаж, что может указывать либо на ограниченный рынок, либо на более специализированный характер товаров, рассчитанный на узкую аудиторию.

## Визуализация скрытых зависимостей

Использование графической визуализации позволяет выявить скрытые зависимости между продажами товаров и временными интервалами. Например, очевидна сезонная корреляция между количеством продаж и последними месяцами года (ноябрь-декабрь), что может быть важным фактором при прогнозировании поведения потребителей в будущем. Для более точного анализа можно использовать методы глубокого обучения для построения прогностических моделей и оценки корреляций.

## ROC-кривые и прогнозирование поведения потребителей

Для дальнейшего анализа и прогнозирования поведения потребителей были использованы методы машинного обучения на ограниченном размере выборки. Для работы с прогнозированием и анализом категорий товаров по датасету Online Retail было использовано ограничение размера выборки, что позволило значительно сократить время выполнения расчетов. Размер выборки был ограничен до 10 000 строк данных. Важно отметить, что такой подход позволил сфокусироваться на небольшом, но репрезентативном объеме данных, чтобы получить быстрые и наглядные результаты.

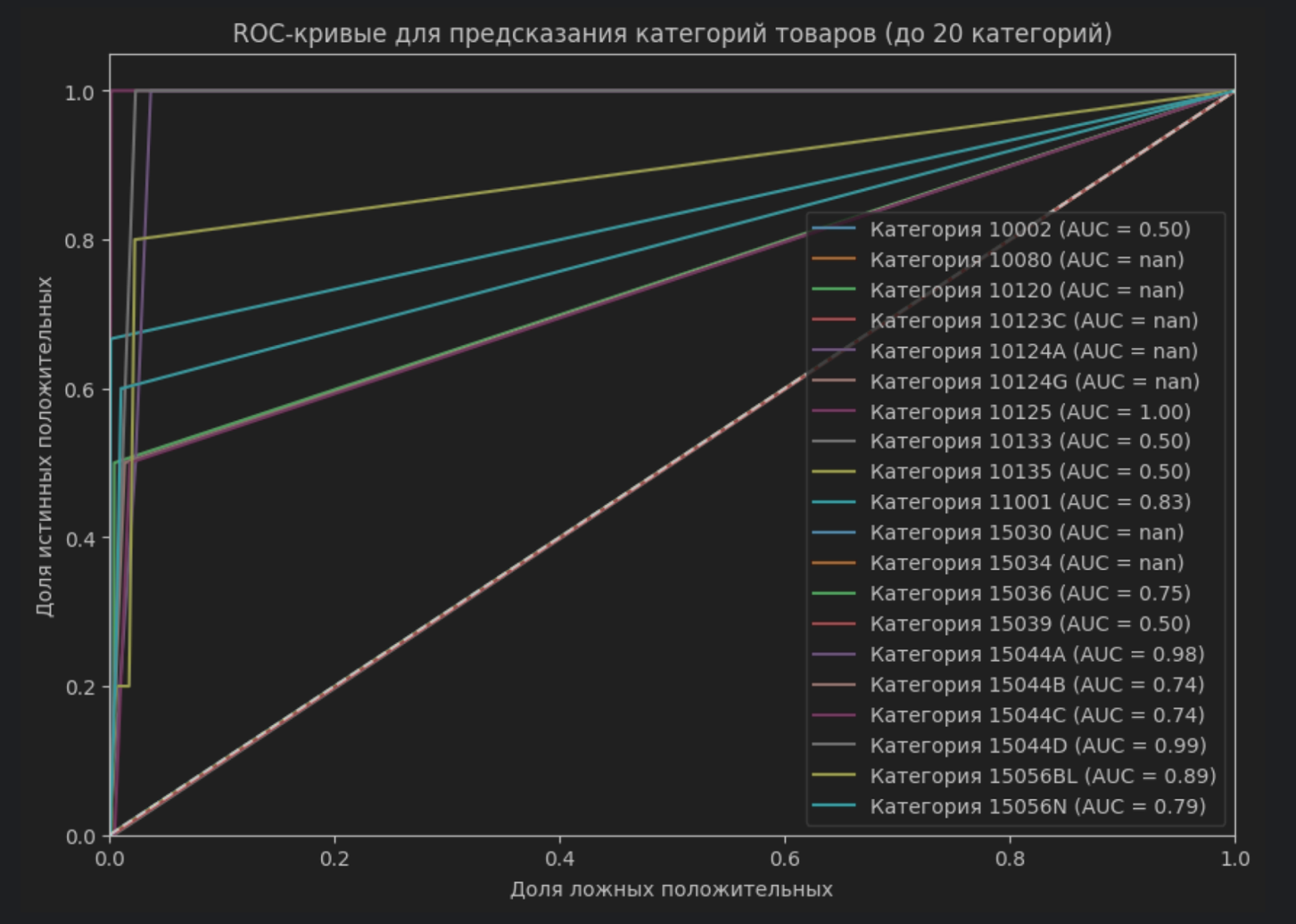
Для построения ROC-кривых была выбрана модель случайного леса, обученная с помощью OneVsRestClassifier, что позволило предсказывать вероятности принадлежности товара к каждой из 20 категорий. В результате, для построения ROC-кривых была использована методология многоклассовой классификации. Результат построения ROC-кривых для 20 категорий показан на рисунке 10.

Рисунок 10 - ROC-кривые для предсказания категорий товаров.

Мы визуализировали прогноз для 20 категорий товаров, что отразилось на графике выше, где по оси X показана доля ложных положительных срабатываний, а по оси Y – доля истинных положительных срабатываний. Значения AUC (Area Under the Curve) для каждой категории варьировались. Для некоторых категорий значения AUC оказались очень низкими или отсутствовали (nan), что может говорить о трудностях в точной классификации этих категорий с текущими параметрами модели или особенностях самих данных.

Визуализация ROC-кривых продемонстрировала, что для некоторых категорий, таких как 10125, модель показала идеальные результаты с AUC = 1.00, что свидетельствует о высокой предсказательной точности. Однако для ряда других категорий значения AUC оказались ниже, например, для категории 15044C – AUC составил 0.74, что указывает на возможность улучшения модели за счет более тонкой настройки параметров или добавления дополнительных признаков.

Прогнозирование с использованием ограниченной выборки в 10 000 строк дало возможность быстро провести анализ и выявить, какие категории товаров поддаются более точному прогнозированию. Такие результаты могут быть использованы для дальнейших исследований и улучшения моделей прогнозирования.

Применение моделей глубокого обучения и классических методов машинного обучения для работы с различными типами данных, такими как временные ряды и категории товаров, показало, что скрытые зависимости в данных можно выявить и визуализировать. Построение ROC-кривых для многоклассовых задач продемонстрировало, что точность предсказания различается в зависимости от категории, что может быть связано как с качеством данных, так и с особенностями самой модели.

## Оценка модели на датасете Fashion MNIST

Одним из ключевых этапов исследования была задача по классификации изображений из датасета Fashion MNIST с использованием сверточной нейронной сети. Данный датасет представляет собой набор из 70 000 изображений предметов одежды, разделенных на 10 классов. Визуализация примеров из тренировочного набора представлена на рисунке 11. Каждый класс представляет определенный тип одежды, включая такие категории, как футболки, брюки, свитеры, платья, кроссовки и другие. Целью исследования было создание модели, способной с высокой точностью предсказывать, к какому из классов относится каждое изображение.



Рисунок 11 - Визуализация нескольких примеров из тренировочного набора датасета Fashion MNIST.

Перед началом обучения мы установили ограничение на размер обучающей выборки — 10 000 изображений. Это решение было принято с целью оптимизации времени на обучение и анализа модели. Ограничение объема данных позволило нам также проверить, насколько эффективно сеть сможет обучаться и демонстрировать высокие результаты даже при относительно небольшом объеме входных данных. Тем не менее, несмотря на данное ограничение, модель показала высокие результаты на тестовых данных.

Обучение модели было проведено в течение 10 эпох. На каждом этапе обучения отслеживались два ключевых показателя — точность и потери. В процессе обучения удалось добиться следующих результатов:

- Точность на тестовых данных составила 90,2%, что является достаточно высоким результатом для такой задачи. Данная метрика свидетельствует о том, что модель способна правильно классифицировать 90,2% изображений в тестовом наборе данных. Такой результат подтверждает, что сверточные нейронные сети являются мощным инструментом для решения задач компьютерного зрения, таких как классификация изображений.

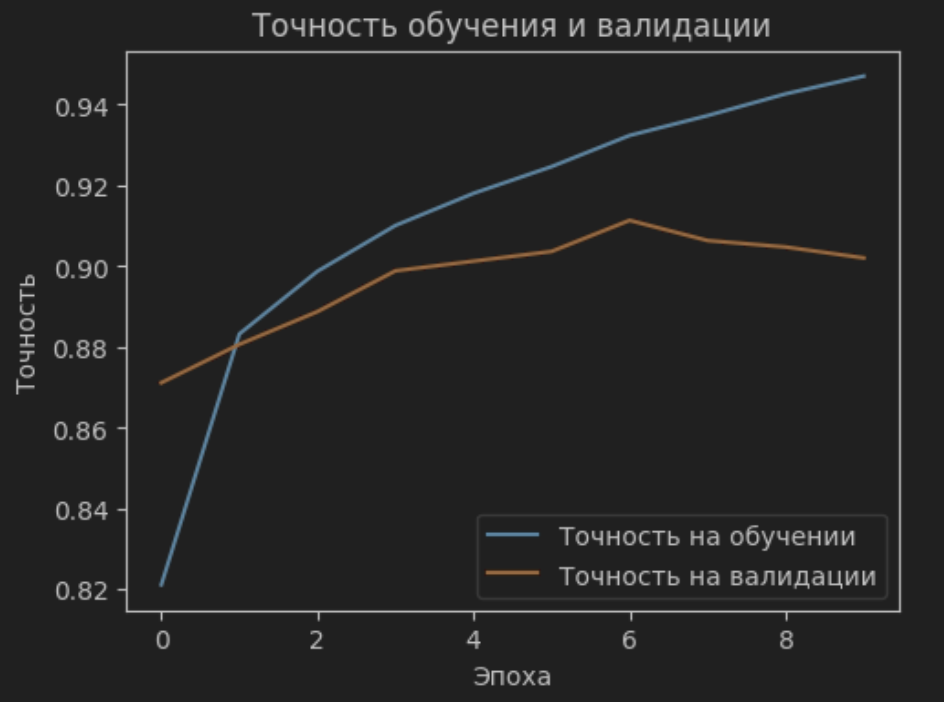
- Потери на тестовых данных составили 0.2804, что указывает на относительно небольшие ошибки в предсказаниях модели. Потери или значение функции ошибки позволяют оценить, насколько сильно предсказания модели отклоняются от фактических значений.

Рисунок 12 – Точность обучения и валидации.

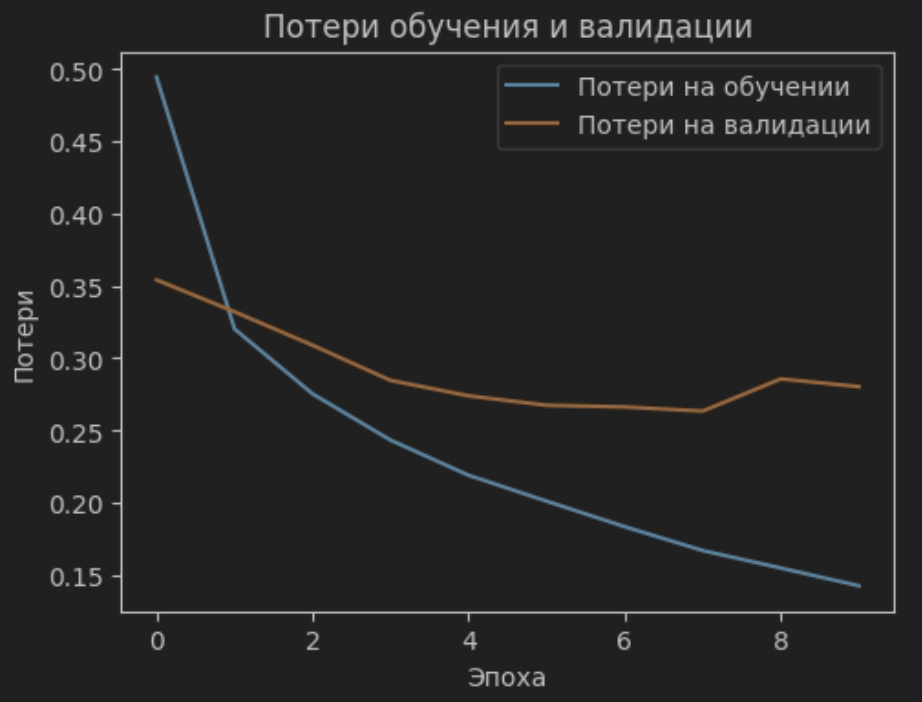


Рисунок 13 – Потери обучения и валидации.

На графиках представленных на рисунках 12 и 13, динамика изменения точности и потерь на обучающей и валидационной выборках:

- Точность обучения и валидации демонстрирует стабильное увеличение на протяжении всех 10 эпох обучения. Начиная с уровня около 88%, точность на обучении выросла до 94% к концу последней эпохи, тогда как на валидации точность стабилизировалась на уровне около 90%. Важно отметить, что разрыв между точностью на обучении и валидации остается небольшим, что свидетельствует о том, что модель не подверглась явному переобучению.

- Потери на обучении и валидации плавно уменьшались на протяжении всех эпох. Это подтверждает, что модель корректно обучалась и снижала свои ошибки на каждом этапе. Потери на валидации стабилизировались на уровне около 0.28, что является хорошим показателем для такой задачи.

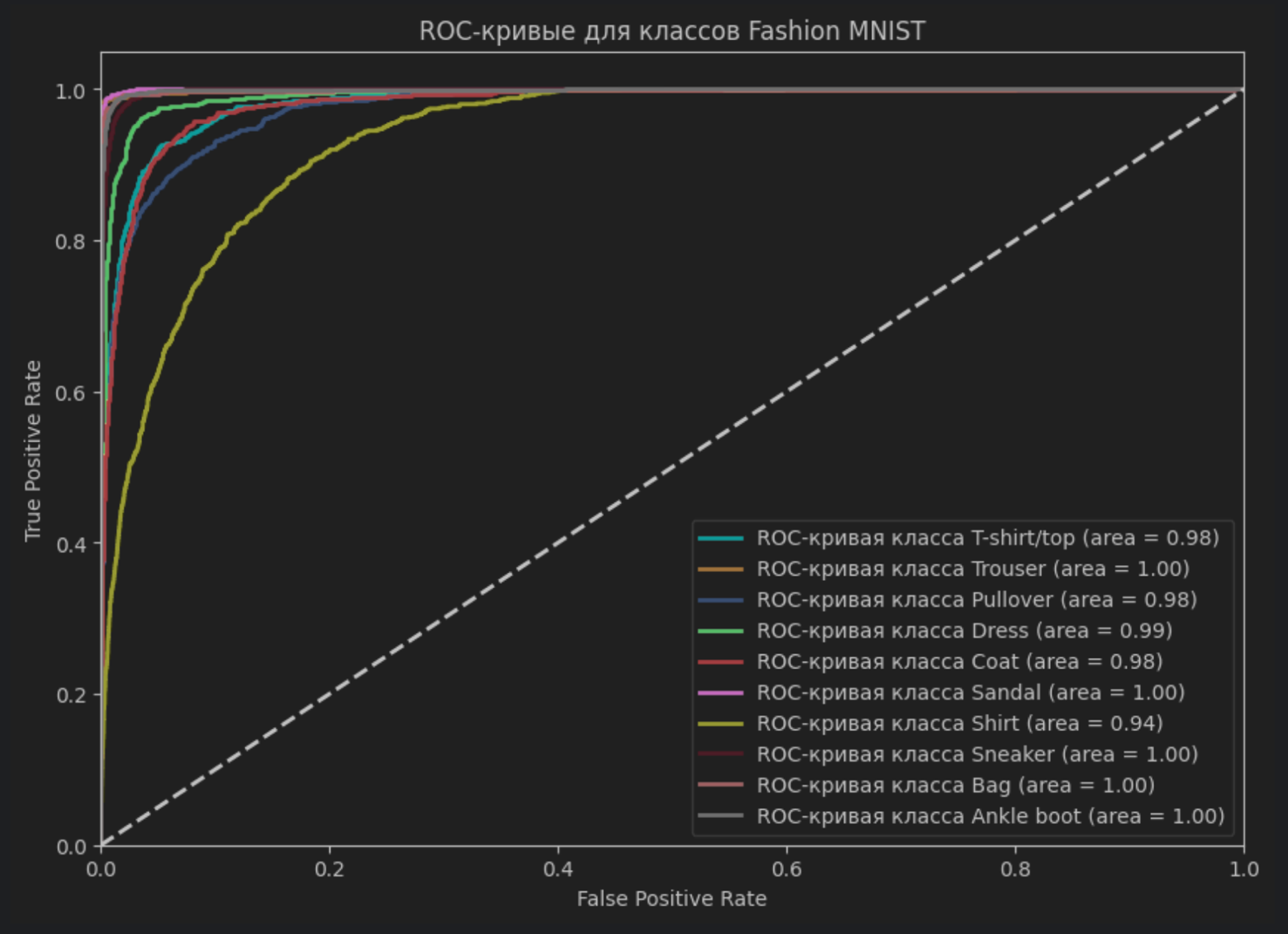
Таким образом, на основании представленных метрик можно сделать вывод о том, что сверточная нейронная сеть продемонстрировала высокую эффективность в решении задачи классификации изображений одежды, несмотря на ограниченный объем входных данных.

## Визуализация ROC-кривых для категорий товаров датасета Fashion MNIST

Для более детальной оценки работы модели были построены ROC-кривые для каждого из 10 классов датасета Fashion MNIST. ROC-кривые (Receiver Operating Characteristic) представляют собой инструмент для оценки качества классификации и позволяют определить баланс между чувствительностью (True Positive Rate) и специфичностью (False Positive Rate) модели.

В нашем случае каждая кривая демонстрирует, насколько точно модель классифицирует изображения для каждого конкретного класса. Площадь под кривой (AUC) является метрикой, которая оценивает качество классификации для каждого класса — чем ближе AUC к 1, тем лучше модель справляется с классификацией.

Рассмотрим результаты для каждого из классов, визуализация построения ROC-кривых для категорий товаров датасета Fashion MNIST показана на рисунке 14:

Рисунок 14 - ROC-кривые для категорий товаров датасета Fashion MNIST.

- Класс T-shirt/top (футболка/топ): AUC = 0.98. Это указывает на высокую точность модели при распознавании данной категории. Модель уверенно различает футболки и топы среди других категорий, однако есть небольшой процент ошибок.

- Класс Trouser (брюки): AUC = 1.00.

Этот результат свидетельствует о том, что модель идеально классифицирует брюки, не допуская ложных срабатываний. Это может быть связано с тем, что брюки легко отличаются от других категорий, таких как свитеры или платья, по своим характерным признакам.

- Класс Pullover (свитер): AUC = 0.98. Модель хорошо справляется с классификацией свитеров, хотя возможны некоторые ошибки в случаях, когда свитеры визуально схожи с другими категориями одежды.

- Класс Dress (платье): AUC = 0.99. Платья также были классифицированы с высокой точностью, что указывает на четкие различия между платьями и другими категориями.

- Класс Sandal (сандалии): AUC = 1.00. Модель прекрасно распознает сандалии, что объясняется их визуальной уникальностью по сравнению с другими типами обуви и одежды.

- Класс Sneaker (кроссовки): AUC = 1.00. Кроссовки были классифицированы с такой же точностью, как и сандалии, что подтверждает высокую уверенность модели в распознавании этой категории.

- Класс Bag (сумка): AUC = 1.00. Сумки, благодаря своей форме и текстуре, легко отличимы от остальных предметов одежды, что обеспечивает столь высокую точность.

- Класс Ankle boot (ботинки): AUC = 1.00. Этот результат также демонстрирует идеальную классификацию ботинок.

Стоит отметить, что для таких классов, как Shirt (рубашка), AUC немного ниже и составляет 0.94. Это может свидетельствовать о том, что модель иногда путает рубашки с футболками или свитерами, поскольку эти категории могут иметь схожие визуальные характеристики.

Таким образом, ROC-кривые подтверждают высокую точность работы модели по большинству категорий. Высокие значения AUC, близкие к 1, говорят о том, что модель не только хорошо классифицирует изображения, но и демонстрирует надежные результаты по каждому классу, минимизируя ложные срабатывания.

## Сравнение с альтернативными моделями

При решении задачи классификации товаров важно учитывать особенности различных алгоритмов. Для задачи классификации изображений, таких как датасет Fashion MNIST, были рассмотрены несколько подходов, включая сверточные нейронные сети (CNN) и традиционные алгоритмы, такие как XGBoost и случайный лес.

Модель CNN показала высокую точность на тестовых данных – 90,2%, что делает ее более эффективной для работы с изображениями по сравнению с классическими методами машинного обучения. Это обусловлено тем, что сверточные сети способны эффективно обрабатывать пространственную структуру изображений, выявляя паттерны на различных уровнях абстракции. В то время как алгоритмы, такие как XGBoost, хорошо работают с табличными данными, их применение для изображений требует дополнительных шагов по извлечению признаков, что снижает их эффективность в подобных задачах.

## Применение модели в реальных условиях

Модель сверточной нейронной сети, обученная на данных Fashion MNIST, может применяться для автоматизации процессов в различных отраслях. Например, в сфере электронной коммерции она может быть использована для автоматической классификации товаров по изображениям. В крупных интернет-магазинах, где ассортимент товаров постоянно расширяется, автоматическая классификация на основе изображений поможет сократить трудозатраты на ручную разметку данных, повысить точность поиска и формирования предложений для клиентов.

Модель также может найти применение в системах визуального контроля качества, где требуется определять дефекты продукции по изображениям. В таких задачах важна высокая точность и устойчивость модели к различным искажениям изображений, что также подтверждается результатами обучения.

## Проблемы и ограничения модели

Несмотря на достигнутую высокую точность (90.2%), существует ряд ограничений, с которыми сталкивается данная модель. Анализ ROC-кривых (см. рисунок 14) показывает, что не все классы распознаются с одинаковой эффективностью. Хотя для некоторых категорий, таких как "Кроссовки" и "Сумки", показатель AUC (площадь под кривой) достигает 1.00, для других классов, например, "Рубашки", AUC составляет 0.94, что указывает на проблемы с точностью распознавания.

Другим ограничением является различие между потерями на обучающих и валидационных данных. На графике видно (см. рисунок 12, 13), что с увеличением количества эпох, модель достигает высокой точности на обучении, однако на валидационных данных потери остаются несколько выше. Это может свидетельствовать о частичном переобучении модели, что требует дополнительной оптимизации.

Кроме того, было применено ограничение на размер выборки, при котором использовалось 10 000 изображений. Увеличение объема данных могло бы улучшить обобщающую способность модели, однако, даже с ограниченной выборкой, результаты остаются на достаточно высоком уровне.

## Перспективы улучшения модели

Для улучшения производительности модели существует несколько путей. Во-первых, можно расширить тренировочный набор данных и применить методы аугментации данных, такие как повороты, масштабирование и изменение яркости изображений. Это позволит модели лучше обрабатывать разнообразные вариации данных и уменьшить вероятность переобучения.

Во-вторых, можно рассмотреть более сложные архитектуры сверточных нейронных сетей, такие как ResNet или EfficientNet, которые показали высокие результаты в классификации изображений. Эти архитектуры обладают способностью глубже анализировать изображения, что может еще больше повысить точность модели.

Дополнительная настройка гиперпараметров, таких как скорость обучения, размер пакета и количество эпох, может также улучшить сходимость модели и сократить потери на валидации. Применение регуляризации, например Dropout или L2-регуляризации, также может способствовать улучшению обобщающей способности модели.

## Визуализация результатов

На представленных графиках (см. рисунок 12, 13) можно наблюдать, как изменяются точность и потери модели на этапах обучения и валидации.

Видно, что по мере увеличения числа эпох точность модели на тренировочных данных возрастает, достигая значения 94% на 10-й эпохе. При этом, точность на валидации также увеличивается, но не столь резко, достигая максимума в 90%. Этот разрыв указывает на необходимость дополнительных мер для предотвращения переобучения.

ROC-кривые, построенные для каждого класса, демонстрируют, что модель хорошо распознает большинство классов. Для категорий "Кроссовки", "Сандалии", "Сумки" и "Ботинки" модель достигает почти идеальных показателей с AUC равным 1.00. Это свидетельствует о том, что модель уверенно классифицирует эти категории. Однако для таких классов, как "Футболка/Топ" и "Пуловер", AUC ниже – 0.98, что требует дальнейшего улучшения.

# **ВЫВОДЫ**

В ходе исследования нами были использованы разнообразные методы машинного обучения для решения задач классификации и прогнозирования потребительского поведения. Основной акцент был сделан на работе с двумя разными типами данных — структурированными данными по покупкам из онлайн-ретейла и изображениями товаров из датасета Fashion MNIST. Для каждого типа данных мы адаптировали подходы к их обработке и построению моделей, что позволило выявить важные зависимости и особенности поведения моделей на различных наборах данных.

## 1. Простые зависимости

Во многих случаях, модели машинного обучения, такие как RandomForest и XGBoost, смогли выявить простые линейные и нелинейные зависимости между характеристиками товаров и результатами классификации. Например, для классификации покупок на основе данных онлайн-ретейла, такие признаки, как дата заказа или идентификатор товара, оказывали непосредственное влияние на предсказания модели. Эти модели демонстрировали высокий уровень точности за счет своей способности обрабатывать как простые линейные зависимости, так и более сложные взаимосвязи. Так, случайный лес показал отличные результаты благодаря построению множества независимых деревьев решений, которые эффективно справляются с задачами классификации, основанными на простых зависимостях в данных.

## 2. Нелинейные зависимости

Во многих случаях, особенно в задачах классификации изображений, зависимости между признаками могут носить сложный нелинейный характер. Это было особенно заметно при работе с изображениями из датасета Fashion MNIST, где традиционные линейные модели не показали бы столь высоких результатов. Для решения этой задачи была применена сверточная нейронная сеть (CNN), способная эффективно обнаруживать сложные паттерны в изображениях, что привело к высоким значениям точности (около 90%). Этот результат подчеркивает важность использования глубоких нейронных сетей для решения задач, где простые зависимости между признаками оказываются недостаточными.

## 3. Скрытые зависимости

На протяжении экспериментов было выявлено несколько скрытых зависимостей, которые на первый взгляд не были очевидны. Применение методов визуализации и анализа данных позволило обнаружить корреляции, не выявляемые при обычном осмотре данных. Например, при анализе данных по продажам товаров в онлайн-ретейле выявилась сильная сезонная зависимость, проявляющаяся в росте продаж в конце года (ноябрь-декабрь). В случае с изображениями, использование сверточных нейронных сетей позволило выделить скрытые пространственные зависимости, такие как форма и текстура объектов, которые трудно было бы выявить с использованием традиционных методов анализа. ROC-кривые и метрики качества (AUC) продемонстрировали высокую точность классификации большинства классов, что также подтверждает наличие глубоких скрытых закономерностей в данных.

## 4. Общие выводы по применяемым моделям

Применение различных моделей машинного обучения, таких как случайный лес, XGBoost и сверточные нейронные сети, позволило решить задачи классификации с высокой точностью и полнотой. Каждый метод показал свои сильные стороны: модели на основе деревьев решений лучше справлялись с предсказанием на структурированных данных, а нейронные сети показали высокую эффективность в задачах, требующих анализа изображений. Важной особенностью стала возможность гибко настраивать параметры моделей для улучшения их производительности с помощью методов, таких как GridSearch и кросс-валидация.

Таким образом, данное исследование продемонстрировало важность выбора подходящих методов машинного обучения в зависимости от природы данных и задач.Оценка моделей показала, что использование комбинации различных подходов, включая методы работы с пропусками в данных и визуализацию зависимостей, позволяет получать более точные результаты и выявлять скрытые закономерности, что является важным шагом к более глубокому пониманию потребительского поведения и повышению точности прогнозов.

# **СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**

В ходе выполнения работы использовались следующие источники:

1. Гудфеллоу И., Бенджио Й., Курвил А. Глубокое обучение. – Москва: Издательство «ДМК Пресс», 2018.

2. Хасти Т., Тибширани Р., Фридман Дж. Элементы статистического обучения. – Москва: Издательство «МЦНМО», 2015.

3. Бишоп К. М. Pattern Recognition and Machine Learning. – Springer, 2006.

4. Касперский А., Новиков Д. Искусственный интеллект: основы, методы и перспективы. – Москва: Либроком, 2020.

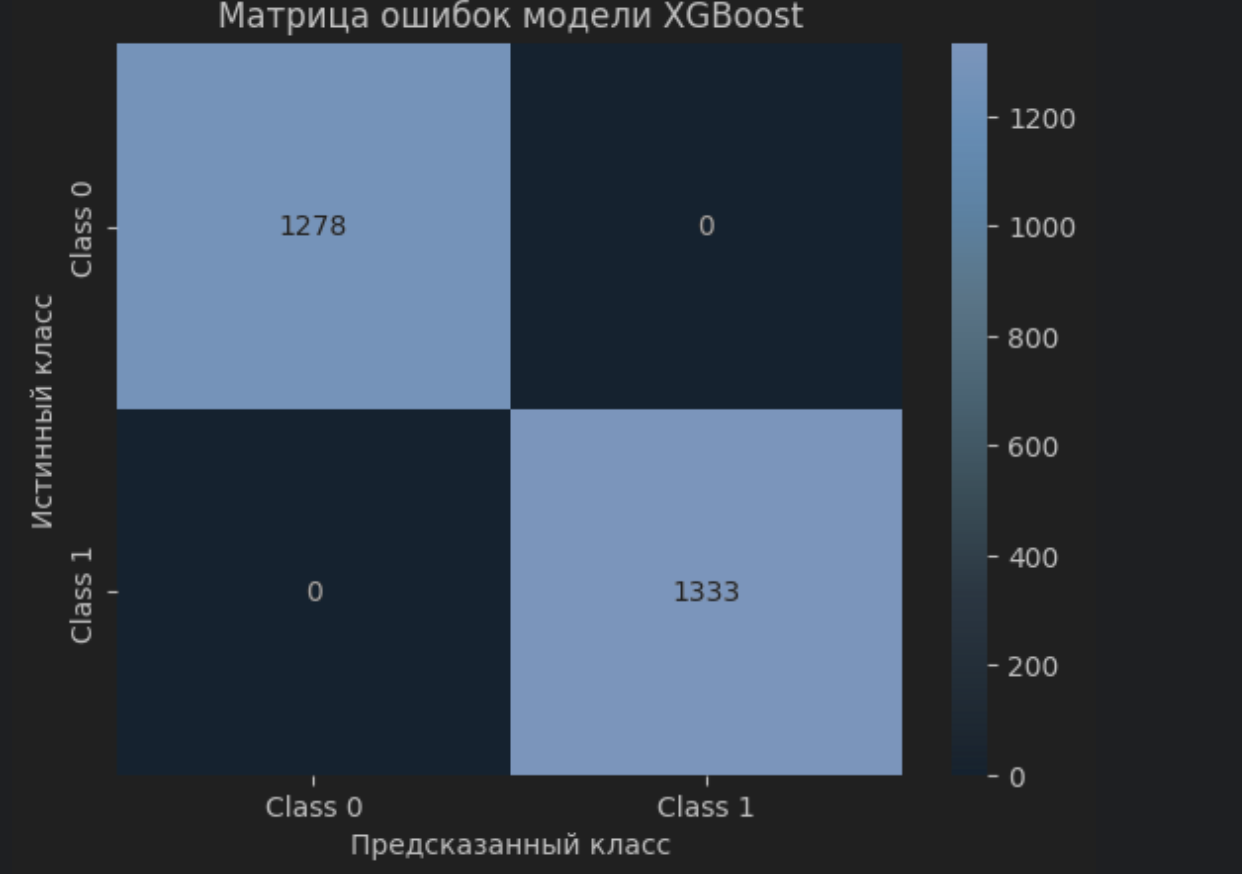
5. Стэнфордский курс по машинному обучению от Andrew Ng. URL: https://www.coursera.org/learn/machine-learning.

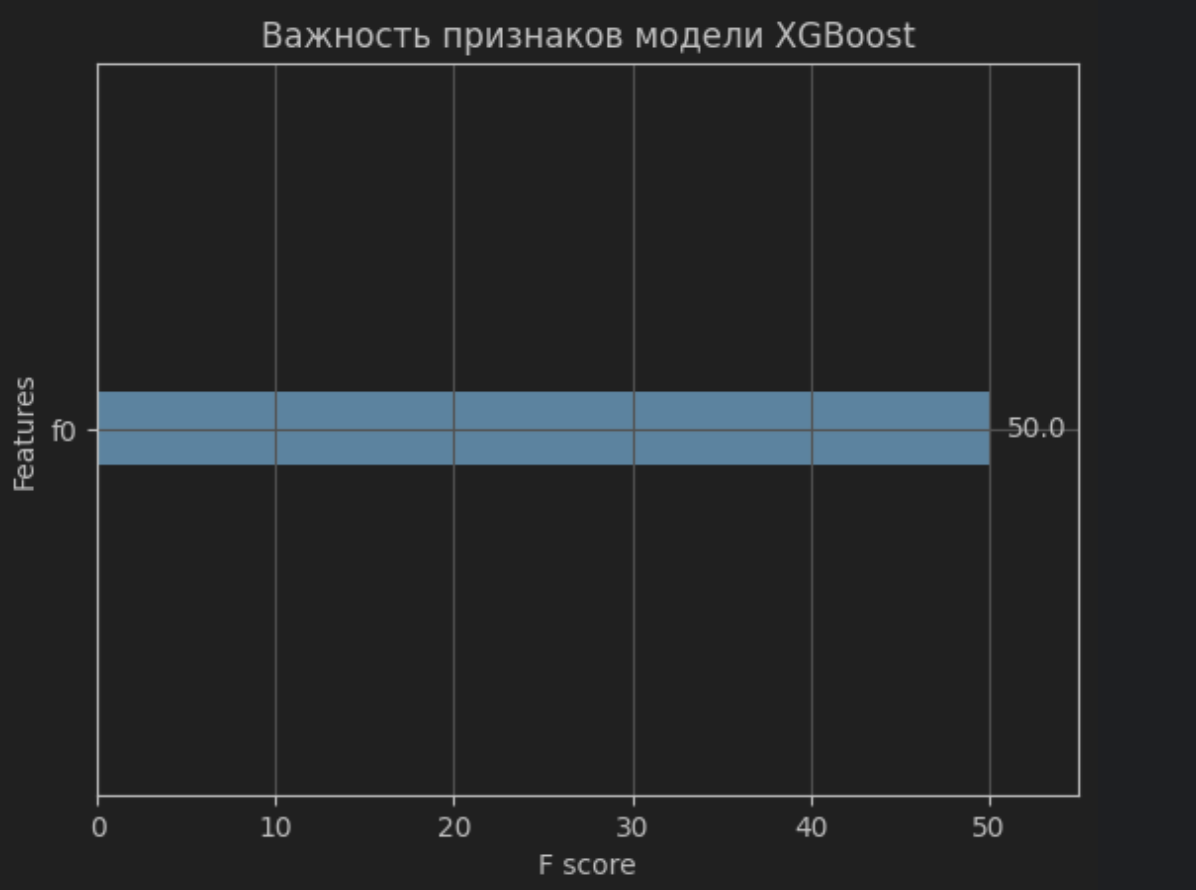
6. Официальная документация библиотеки TensorFlow. URL: https://www.tensorflow.org/.

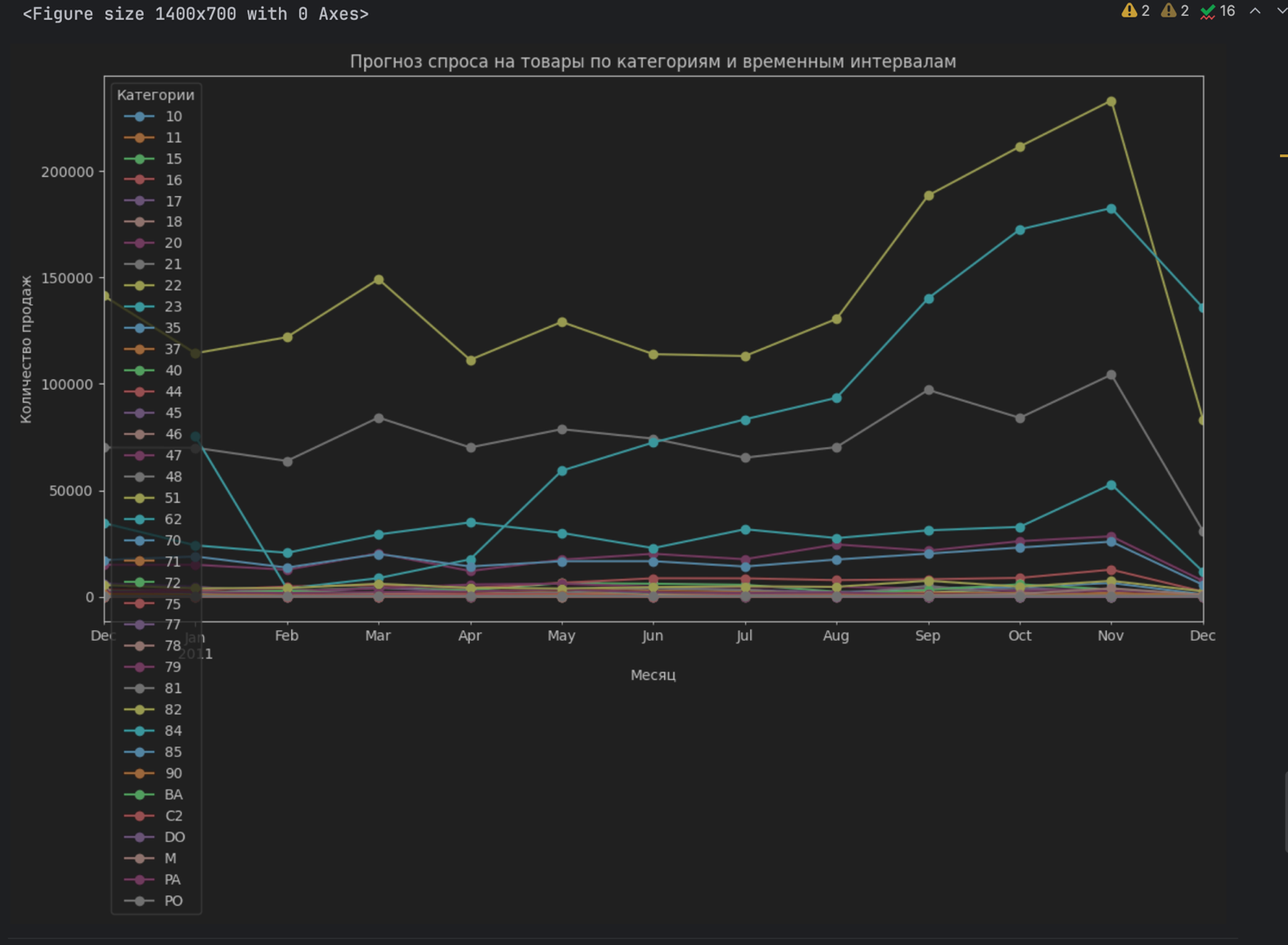
7. Kaggle: платформа для работы с данными. URL: https://www.kaggle.com/.

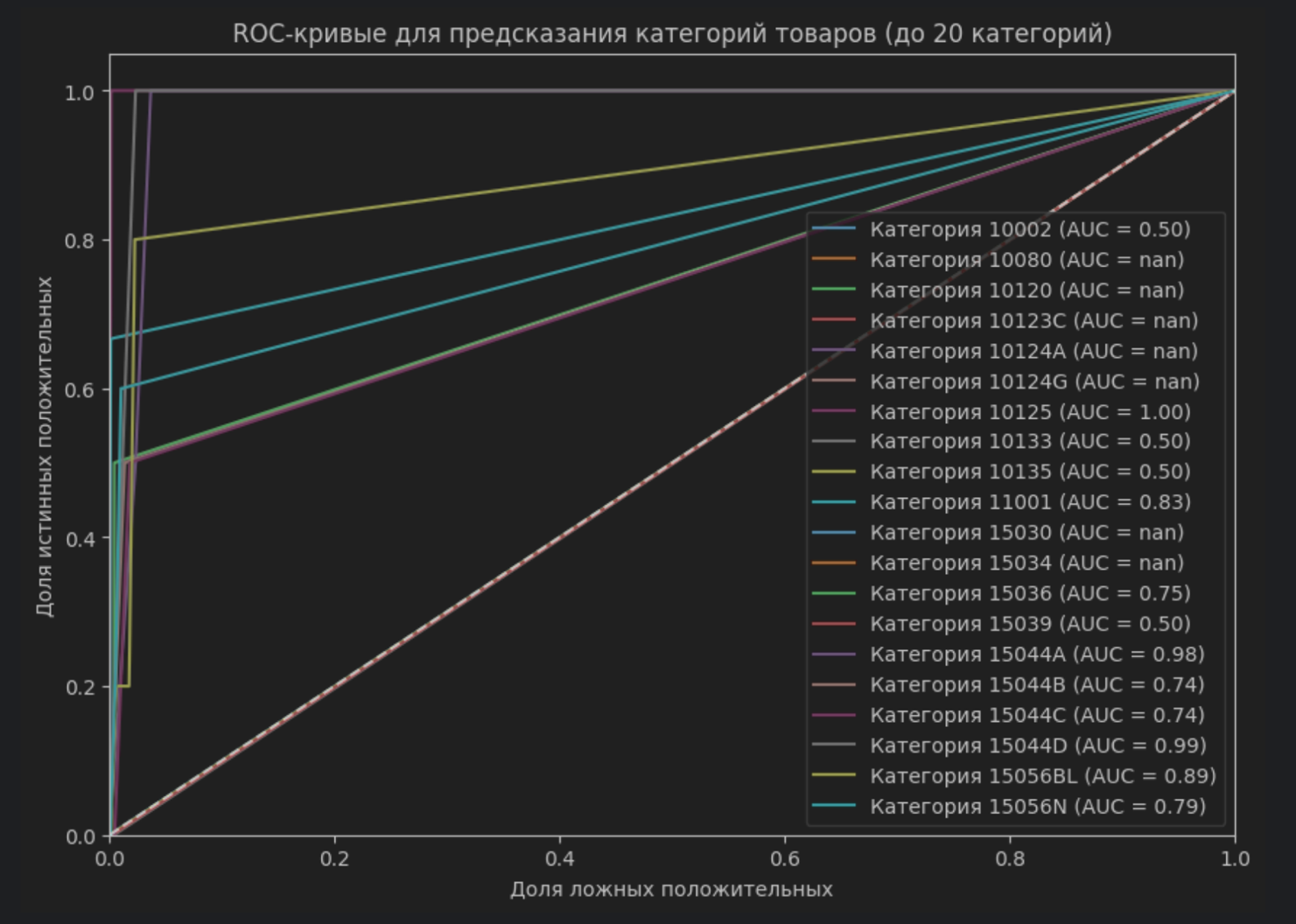
# **ПРИЛОЖЕНИЯ**

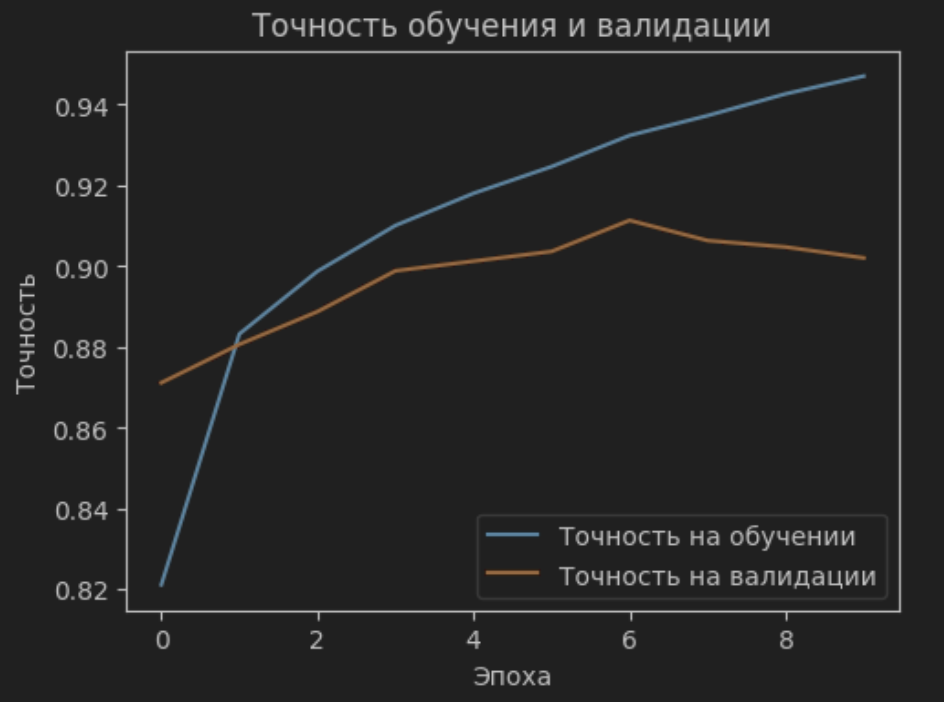
## Визуализация данных

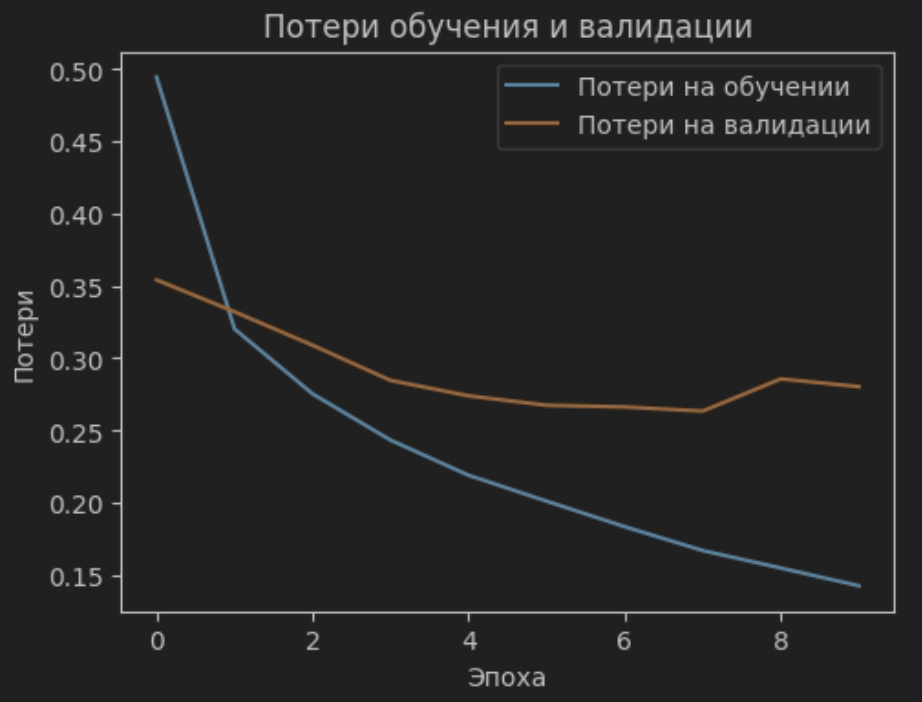
Рисунок 7 - Визуализации матрицы ошибок.

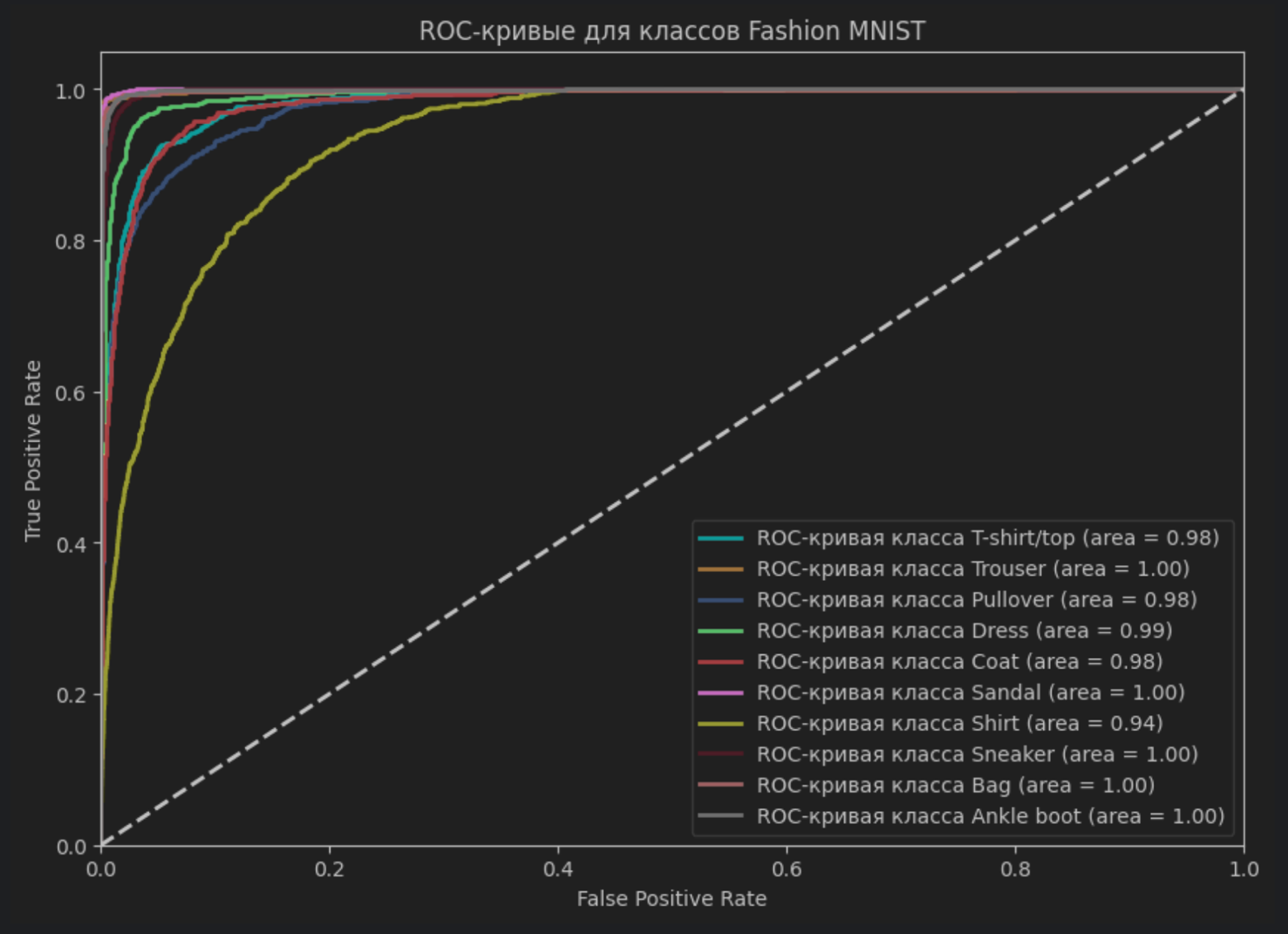
Рисунок 8 – Важность признаков модели XGBoost.

Рисунок 9 - Прогноз спроса на товары по категориям и временным интервалам.

Рисунок 10 - ROC-кривые для предсказания категорий товаров.

Рисунок 12 – Точность обучения и валидации.

Рисунок 13 – Потери обучения и валидации.

Рисунок 14 - ROC-кривые для категорий товаров датасета Fashion MNIST.

## Код

Код используемый в ходе выполнения дипломной работы:

* practice\_part\_02.ipynb
* practice\_part\_03.ipynb

## Проект на GitHub:

<https://github.com/Anton-Savenkov/diplom_project_01.git>