**Министерство науки и высшего образования Российской Федерации**

**федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Российский экономический университет имени Г. В. Плеханова»**

Кафедра информатики

**Выпускная квалификационная работа**

**по программе профессиональной переподготовке**

на тему «Прогнозирование времени доставки с использованием методов машинного обучения»

Выполнила:

Бойкевич Ксения Александровна

15.11д-БИТЦ10/21б

3 курс, ВШКМиС

Преподаватель:

ст. преп. Савинова Виктория Михайловна

Москва 2024

**Понимание бизнес-целей**

* 1. **Понимание бизнеса**

В быстро развивающемся мире онлайн-торговли, где клиенты привыкли к мгновенному удовлетворению своих потребностей, скорость и надежность доставки стали ключевыми факторами конкуренции. Компании, способные оптимизировать свои логистические процессы и предоставить клиентам точную информацию о времени доставки, получают неоспоримое преимущество. Это особенно актуально в сфере доставки продуктов питания, где спрос на быстрое и качественное обслуживание высок.

Одной из ключевых задач в этой области является разработка системы прогнозирования времени доставки, учитывающей параметры заказа, местоположение клиента, а также характеристики курьера. Использование данных о заказах, курьерах и клиентах позволит оптимизировать маршруты, прогнозировать время доставки и повысить эффективность работы службы доставки. Кроме того, анализ данных может помочь определить проблемные зоны в работе курьеров, улучшить управление заказами и, как результат, повысить удовлетворенность клиентов и рентабельность бизнеса.

В качестве примера можно привести платформу доставки "Porter" в Индии, которая предоставляет услуги доставки продуктов питания от различных ресторанов. Компания обладает обширной базой данных, включающей информацию о заказах, курьерах, ресторанах и клиентах. Эта информация может быть использована для совершенствования логистических процессов, повышения точности прогнозирования времени доставки и улучшения качества предоставляемых услуг.

В связи с предпосылками возникают следующие задачи:

1. Повышение точности прогнозирования времени доставки, чтобы уменьшить время доставки и повысить удовлетворенность клиентов.

2. Оптимизация логистических процессов, таких как маршрутизация, управление запасами и распределение ресурсов, что свою очередь может уменьшить время доставки и снизить затраты.

3. Выявление проблемных зон в работе курьеров и предложение решения для повышения их эффективности.

4. Улучшение управления заказами, таких как прогнозирование спроса и планирование запасов.

5. Улучшение своих услуг и повышение конкурентоспособности на рынке.

Критерий успеха – рост рентабельности продаж.

* 1. **Доступные ресурсы**

Для успешной реализации проекта необходимы следующие категории

специалистов: аналитик данных, бизнес-аналитик, специалист по базам данных,

руководитель проекта.

Заказчик располагает всем необходимым оборудованием для проведения

анализа данных.

* 1. **Риски**

1. Несоблюдение сроков проекта

2. Риск неплатежеспособности заказчика

3. Риск нехватки и неполноты данных

4. Риск несоответствия полученных результатов требованиям заказчика

* 1. **Ограничения**

Ограничение сроков: 6 месяцев. Ставки по сотрудникам:

Аналитик данных – 1 ставка.

Бизнес-аналитик – 1 ставка.

* 1. **Цели исследования данных**

Исследование направлено на создание модели, которая позволит предсказывать время доставки. Для этого будет проведен анализ имеющихся данных, включающий визуализацию влияния различных факторов на скорость доставки и построение таблиц с описанием данных. Затем, на основе полученного понимания, будут построены и сравнены несколько моделей регрессии, таких как линейная регрессия, дерево решений, случайный лес, метод ближайшего соседа и XGBoost. Цель - выбрать модель, которая наиболее точно и стабильно предсказывает время доставки на основе тестовых данных.

* 1. **Критерии успешности изучения данных**

Качество модели определяется с использованием коэффициента детерминации (R2), точность модели определяется на основании средней относительной ошибки

(MAPE).

Границы значений метрик: R2 должен быть больше либо равен 0.8, MAPE не более 10%.

**Начальное изучение данных**

* 1. **Сбор данных**

**Внутренние данные** – идентификатор рынка, время создания заказа, фактическое время доставки, идентификатор магазина, основная категория магазина, протокол заказа, общее количество товаров в заказе, сумма заказа без учета доставки, количество уникальных товаров в заказе, минимальная цена товара в заказе, максимальная цена товара в заказе, общее количество курьеров, находящихся на смене, общее количество курьеров, занятых выполнением заказов, общее количество заказов в ожидании доставки.

(https://www.kaggle.com/datasets/ranitsarkar01/porter-delivery-time-estimation/data)

**Внешние данные** – не требуются

**Дополнительные данные** – не требуются

* 1. **Описание данных**

Объем данных – 21,1+MB

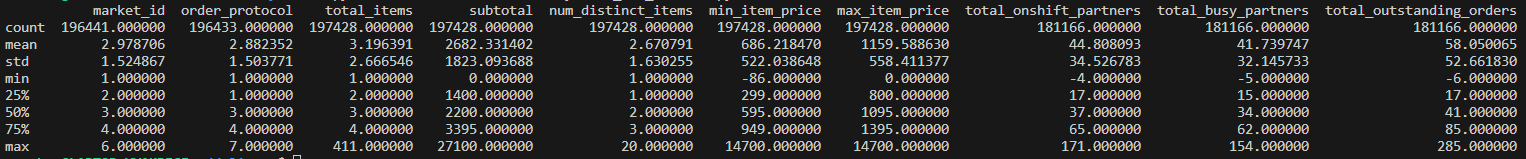
Типы, виды данных и схемы кодирования

|  |  |  |  |
| --- | --- | --- | --- |
| **Наименование** | **Тип данных** | **Вид данных** | **Схема кодирования** |
| market\_id | float | непрерывный | - |
| created\_at | object | дискретный | время |
| actual\_delivery\_time | object | дискретный | время |
| store\_id | object | дискретный | - (столбец будет удален) |
| store\_primary\_category | object | дискретный | - (столбец будет удален) |
| order\_protocol | float | непрерывный | - |
| total\_items | int | непрерывный | - |
| subtotal | int | непрерывный | - |
| num\_distinct\_items | int | непрерывный | - |
| min\_item\_price | int | непрерывный | - |
| max\_item\_price | int | непрерывный | - |
| total\_onshift\_partners | float | непрерывный | - |
| total\_busy\_partners | float | непрерывный | - |
| total\_outstanding\_orders | float | непрерывный | - |

Формат данных – файл csv, разделитель – “,”.

* 1. **Исследование данных**

Описательная статистика:

  
• Средний заказ: состоит из 3-х товаров, имеет стоимость около 2682, содержит 2-3 различных товара.

• Цены: минимальная цена товара колеблется в районе 299, максимальная - 1395, причем есть записи с отрицательной минимальной ценой (что может быть ошибкой в данных).

• Курьеры: в среднем доступно 45 курьеров, из которых около 42 заняты выполнением заказов, а 58 заказов ожидают обработки.

• Разброс данных: стандартное отклонение говорит о том, что данные распределены неравномерно - есть как совсем маленькие, так и очень большие заказы, с различным количеством товаров и курьеров.

Пропущенные значения:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Дубликаты:



Матрица корреляции:

Изображение выглядит как снимок экрана, прямоугольный, Прямоугольник, шаблон

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, диаграмма, Прямоугольник

Автоматически созданное описание

В датасете содержатся выбросы.

**Подготовка данных**

В процессе подготовки данных для модели машинного обучения были предприняты следующие шаги:

1. Предобработка:

• Удалены столбцы 'store\_id' и 'store\_primary\_category', поскольку они были слишком информативными и могли привести к переобучению модели, когда модель запоминает слишком много деталей из обучающей выборки и не может обобщить полученные знания на новые данные.

• Столбцы 'created\_at' и 'actual\_deivery\_time' преобразованы в формат даты и времени.

• Создана новая функция 'time\_taken(mins)' для вычисления времени доставки в минутах.

• Удалены исходные столбцы 'created\_at' и 'actual\_deivery\_time', так как они больше не нужны.

2. Очистка данных:

• Применено правило трех сигм для удаления выбросов.

• Удалены пропуски.

• Данные в наборе нормализованы (метод: MinMaxScaler).

3. Исправление ошибок:

• Выявлено и исправлено несколько ошибок в описательной статистике:

* В столбце 'min\_item\_price' минимальное значение было равно -86, что невозможно для цены. Все отрицательные значения были удалены.
* Аналогично, в столбцах 'total\_outstanding\_order', 'total\_busy\_partners' и 'total\_onshift\_partners' были обнаружены отрицательные значения, которые были удалены, поскольку они не могут быть отрицательными.

4. Разделение данных:

• Данные разделены на тестовую и обучающую выборки для дальнейшего обучения и оценки модели (в соотношении 70% на 30%)

5. Удаление нерелевантных столбцов (с корреляцией меньше заданного порога):

* Были удалены столбцы, у которых связь со столбцом «time\_taken(mins)» меньше 0.05

**Моделирование**

Были построены следующие модели:

\_\_\_\_\_\_\_\_\_\_\_ linear regression \_\_\_\_\_\_\_\_\_\_\_\_

MAPE:0.28085078505340305

r2\_score:0.17095912559134097

\_\_\_\_\_\_\_\_\_\_\_ KNN \_\_\_\_\_\_\_\_\_\_\_\_

MAPE:0.31527806085418547

r2\_score:-0.06366503235658327

\_\_\_\_\_\_\_\_\_\_\_ Decision tree \_\_\_\_\_\_\_\_\_\_\_\_

MAPE:0.3086957119795695

r2\_score:-0.03273804992953733

\_\_\_\_\_\_\_\_\_\_\_ Random forest \_\_\_\_\_\_\_\_\_\_\_\_

MAPE:0.26685128667713687

r2\_score:0.23673131476878906

\_\_\_\_\_\_\_\_\_\_\_ XGBoost \_\_\_\_\_\_\_\_\_\_\_\_

MAPE:0.262789519282393

r2\_score:0.25211687031308194

**Оценка результатов**

Наилучшие показатели точности и качества показали модели Random Forest и XGBoost. Однако, ни в одном из случаев не были достигнуты необходимые значения коэффициента детерминации и средней относительной ошибки (MAPE). Модели не способны точно спрогнозировать время доставки товаров.

Рекомендации для заказчика:

Для улучшения точности моделей необходимо обратить внимание на несколько аспектов. В первую очередь, важно проверить качество данных. Следует убедиться, что данные полные, актуальные и не содержат ошибок. Кроме того, стоит изучить влияние других факторов, которые могут существенно влиять на время доставки, и включить их в модель. Погодные условия в день доставки, расстояние до заказчика, количество заказов, выполненных до текущего, вес заказа и тип используемого транспортного средства – все эти параметры могут помочь моделям лучше понять и предсказать время доставки.

**Внедрение**

Построенные модели Random Forest и XGBoost показали наилучшие результаты. Однако, несмотря на это, критерии успешности моделей, такие как коэффициент детерминации и средняя относительная ошибка (MAPE), не соответствуют требованиям. Таким образом, модели не могут быть внедрены для принятия решений и выполнения прогноза по ранее не встречающимся данным из-за возможных ошибок.

В будущем требуется включение в анализ более значимых факторов, влияющих на время доставки, с корреляцией Пирсона не менее 0,5. Такие факторы могут включать погодные условия в день доставки, расстояние до заказчика, количество заказов, выполненных до текущего, вес заказа и тип используемого транспортного средства. Включение этих дополнительных параметров поможет моделям лучше понимать и предсказывать время доставки, что может существенно повысить их точность и пригодность для реального применения.

**Приложение 1.** Код программы Python для проведенного анализа

import pandas as pd

import seaborn as sns

import numpy as np

import matplotlib.pyplot as plt

from sklearn.linear\_model import LinearRegression

from sklearn.neighbors import KNeighborsRegressor

from sklearn.tree import DecisionTreeRegressor

from sklearn.ensemble import RandomForestRegressor

from xgboost import XGBRegressor

from sklearn.metrics import r2\_score, mean\_absolute\_percentage\_error

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import LabelEncoder, MinMaxScaler

from sklearn.preprocessing import StandardScaler

from scipy.stats import zscore

def normal(df):

    for column in df.columns:

        enc = MinMaxScaler()

        enc.fit(df[[column]])

        df[column] = enc.transform(df[[column]])

    return df

df = pd.read\_csv('/home/meowksu/delivery/dataset.csv')

print(df.info())

print(df.head(5))

print("Описательная статистика:")

print(df.describe())

print("\nПропущенные значения:")

print(df.isnull().sum())

print("\nДубликаты:")

duplicates = df.duplicated()

print(f"Количество строк с дубликатами: {duplicates.sum()}")

# Обработка пропущенных значений

print(df.isnull().sum())

df.dropna(inplace=True)

print(df.isnull().sum())

# Изменение формата данных на date\_time

df['created\_at'] = pd.to\_datetime(df['created\_at'])

df['actual\_delivery\_time'] = pd.to\_datetime(df['actual\_delivery\_time'])

# Вычисление времени доставки в минутах

df['time\_taken(mins)'] = (df['actual\_delivery\_time'] - df['created\_at']).dt.total\_seconds() / 60

# Удаление ненужных столбцов

df.drop(['created\_at', 'actual\_delivery\_time'], axis=1, inplace=True)

df.drop(['store\_id', 'store\_primary\_category'], axis=1, inplace=True)

# Применение правила трех сигм для удаления выбросов

threshold = 3

cols\_to\_check = [col for col in df.columns if df[col].dtype!= 'object']

for col in cols\_to\_check:

    df = df[np.abs(zscore(df[col])) < threshold]

# Масштабирование данных с помощью StandardScaler

scaler = StandardScaler()

scaled\_df = scaler.fit\_transform(df)

scaled\_df = pd.DataFrame(scaled\_df, columns=df.columns)

# Разделение данных на признаки (X) и целевую переменную (y)

X = df.drop(columns = ["time\_taken(mins)"], axis = 1)

y = df["time\_taken(mins)"]

# Удаление нерелевантных столбцов (с корреляцией меньше заданного порога)

threshold = 0.05

correlation\_matrix = df.corr()['time\_taken(mins)']

irrelevant\_columns = correlation\_matrix[abs(correlation\_matrix) <= threshold].index

df.drop(irrelevant\_columns, axis=1)

# Разделение данных на обучающую (70%) и тестовую (30%) выборки

X\_train,X\_test,y\_train,y\_test = train\_test\_split(X,y,test\_size = 0.3,random\_state=0)

print(X\_train.shape, y\_train.shape)

print(X\_test.shape, y\_test.shape)

# Обучение и оценка различных моделей машинного обучения

# Линейная регрессия

reg = LinearRegression()

reg.fit(X\_train,y\_train)

y\_pred = reg.predict(X\_test)

print('\_\_\_\_\_\_\_\_\_\_\_ linear regression \_\_\_\_\_\_\_\_\_\_\_\_')

print('MAPE:{}'.format(mean\_absolute\_percentage\_error(y\_test,y\_pred)))

print('r2\_score:{}'.format(r2\_score(y\_test,y\_pred)))

# KNN

reg = KNeighborsRegressor(n\_neighbors=5)

reg.fit(X\_train,y\_train)

y\_pred = reg.predict(X\_test)

print('\_\_\_\_\_\_\_\_\_\_\_ KNN \_\_\_\_\_\_\_\_\_\_\_\_')

print('MAPE:{}'.format(mean\_absolute\_percentage\_error(y\_test,y\_pred)))

print('r2\_score:{}'.format(r2\_score(y\_test,y\_pred)))

# Дерево решений

reg = DecisionTreeRegressor(min\_samples\_leaf=7,

min\_samples\_split=7, criterion='squared\_error')

reg.fit(X\_train,y\_train)

y\_pred = reg.predict(X\_test)

print('\_\_\_\_\_\_\_\_\_\_\_ Decision tree \_\_\_\_\_\_\_\_\_\_\_\_')

print('MAPE:{}'.format(mean\_absolute\_percentage\_error(y\_test,y\_pred)))

print('r2\_score:{}'.format(r2\_score(y\_test,y\_pred)))

# Случайный лес

reg = RandomForestRegressor(min\_samples\_leaf=7, min\_samples\_split=7, criterion='squared\_error')

reg.fit(X\_train,y\_train)

y\_pred = reg.predict(X\_test)

print('\_\_\_\_\_\_\_\_\_\_\_ Random forest \_\_\_\_\_\_\_\_\_\_\_\_')

print('MAPE:{}'.format(mean\_absolute\_percentage\_error(y\_test,y\_pred)))

print('r2\_score:{}'.format(r2\_score(y\_test,y\_pred)))

# XGBoost

xgb=XGBRegressor()

xgb.fit(X\_train,y\_train)

y\_pred=xgb.predict(X\_test)

print('\_\_\_\_\_\_\_\_\_\_\_ XGBoost \_\_\_\_\_\_\_\_\_\_\_\_')

print('MAPE:{}'.format(mean\_absolute\_percentage\_error(y\_test,y\_pred)))

print('r2\_score:{}'.format(r2\_score(y\_test,y\_pred)))