Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет имени Г.В. Плеханова»

**МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ.**

Специальность: 09.02.07 Информационные системы и программирование

Квалификация: Программист

ОТЧЕТ ПО ДИСЦИПЛИНЕ:

ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РАЗРАБОТКИ ПО

Листов: 45

|  |  |
| --- | --- |
| Выполнил студент  Группы П50-6-20  Лебедев В. С.  «\_\_\_\_» \_\_\_\_\_\_\_\_\_ 2023 года | Проверил преподаватель  \_\_\_\_\_\_\_\_\_\_\_М.Н. Гацкан  «\_\_\_\_» \_\_\_\_\_\_\_2023 года |

Москва 2023

### **ОГЛАВЛЕНИЕ**

[Практическая работа №1 3](#_Toc137672924)

[Практическая работа №2 5](#_Toc137672925)

[Практическая работа №3. 7](#_Toc137672926)

[Практическая работа №4. 21](#_Toc137672927)

[Практическая работа №5. 26](#_Toc137672928)

[Практическая работа №6. 34](#_Toc137672929)

[Практическая работа №7. 42](#_Toc137672930)

[Практическая работа №8. 48](#_Toc137672931)

[Практическая работа №8. 48](#_Toc137672932)

# Практическая работа №1

Цель работы: создать калькулятор на языке программирования python, который может совершать такие действия с числами, как: сложение, вычитание, умножение, деление и ввод количества чисел.

Процесс выполнения работы:

1. Создание переменной colvo, в которую пользователь будет вводить количество чисел.

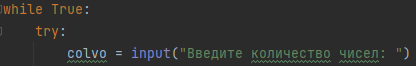


Рисунок 1 (Блок кода 1)

1. Объявление переменной oper, в которую пользователь вводит желаемую операцию.



Рисунок 2 (Блок кода 2)

1. Далее в цикле for, который будет повторяться пока числа не закончатся производить с ними действия, введённые пользователем.

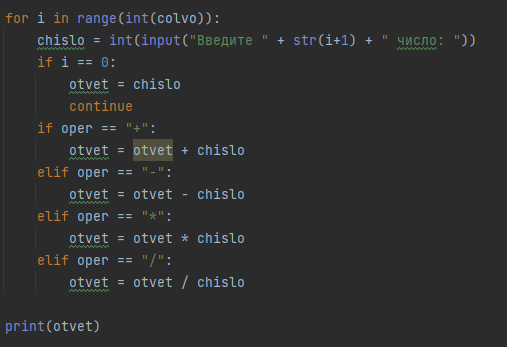


Рисунок 3 (Блок кода 3)

Результат работы программы:

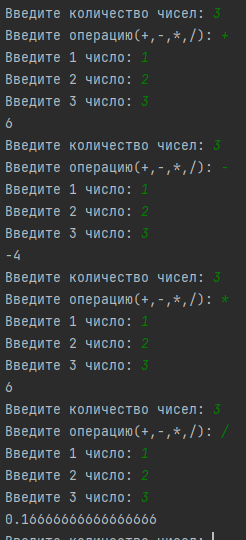


Рисунок 4 (Результат)

Вывод: в ходе практической работы был разработан калькулятор на языке программирования python, который может совершать такие действия с числами, как: сложение, вычитание, умножение, деление.

# Практическая работа №2

Цель работы: составить алгоритм и создать программу подсчета суммы цифр всех дат для каждого месяца всех дней в году.

Процесс выполнения работы:

1. Импортировал библиотеку для подсчета дней



Рисунок 5 (Блок кода 1)

1. Объявление переменной year, в которую пользователь вводит желаемый год, и переменная result, в которую будут складывается значения.



Рисунок 6 (Блок кода 2)

1. Далее в цикле for, который будет повторяться пока не закончатся месяцы, и в другом цикле for мы проходимся по дням месяца.

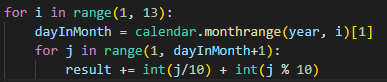


Рисунок 7 (Блок кода 3)

1. Вывод результата.



Рисунок 8 (Блок кода 4)

Результат работы программы:

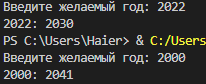


Рисунок 9 (Результат)

Вывод: в ходе практической работы был составлен алгоритм и создана программа подсчета суммы цифр всех дат для каждого месяца всех дней в году.

# Практическая работа №3.

Цель работы: создать программу мини-бизнес на языке программирования python, в которой должно присутствовать: сбор своего блюда, регистрация/авторизация с подтверждением кода по почте, реализация пользователя с балансом и ролью (Администратор/Пользователь), акция, скидка за попадание в блюдо таракана, чек, история покупок и карта лояльности.

Процесс выполнения работы:

1. Написания SQL-скрипта создание скрипта базы данных, с такими таблицами, как: User, которая будет хранить данные пользователя, Order – которая будет хранить данные о заказе пользователя, Ingredients – будет хранить данные об ингредиентах, Ingredients-Order – таблица-связь, которая будет хранить ключи таблиц заказов и ингредиентов.

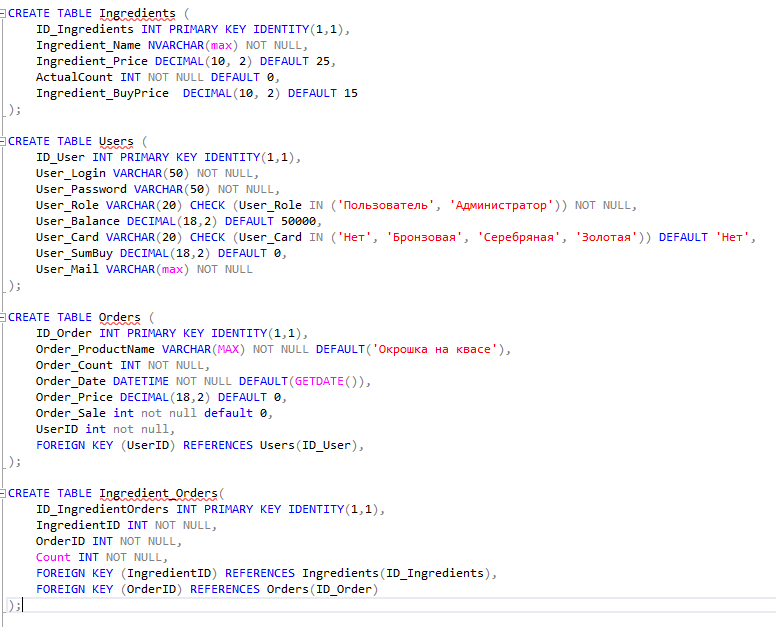


Рисунок – Код SQL-скрипта

1. Импорт всех нужных библиотек для работы, pyodbc – работа с БД, smtplib – отправка кода на почту в момент авторизации/регистрации, os – вызов системных функций, в программе используется для очистки консоли, click – ожидание нажатия клавиши пользователя, random – генерация случайных значений, decimal – преобразования, json – для импорта почты и пароля (чтобы акк через гитхаб не украли).

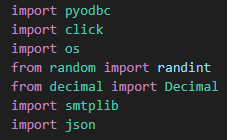


Рисунок – Импорты библиотек

1. Объявление переменных и подключение к базе данных.

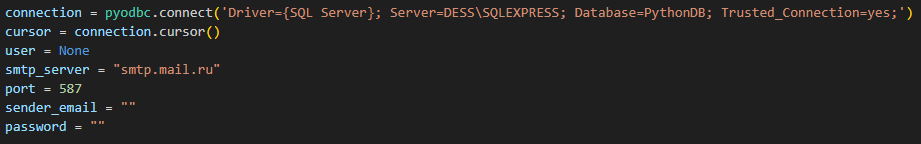


Рисунок – Переменные

1. Метод отправки сообщения пользователю.

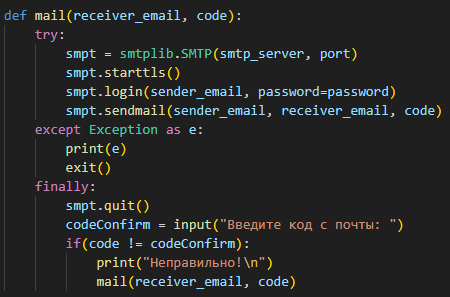


Рисунок – Метод mail

1. Метод выбора пользователя по логину.

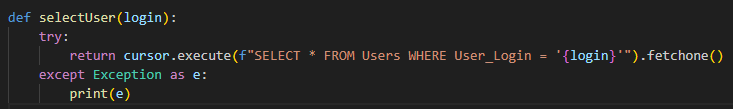


Рисунок – Метод selectUser

1. Метод регистрации пользователя.



Рисунок – Метод toRegister

1. Метод авторизации пользователя.

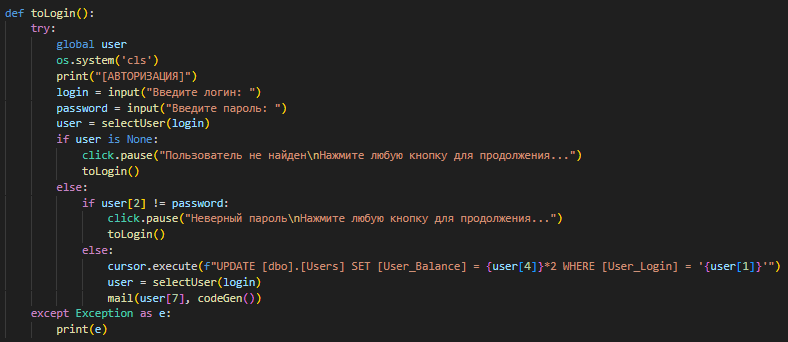


Рисунок – Метод toLogin

1. Метод выбора ингредиента.

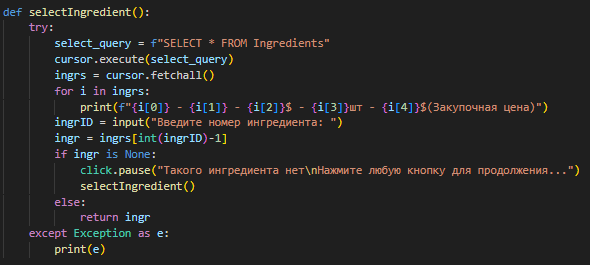


Рисунок – Метод selectIngredient

1. Метод закупки ингредиента администратором.

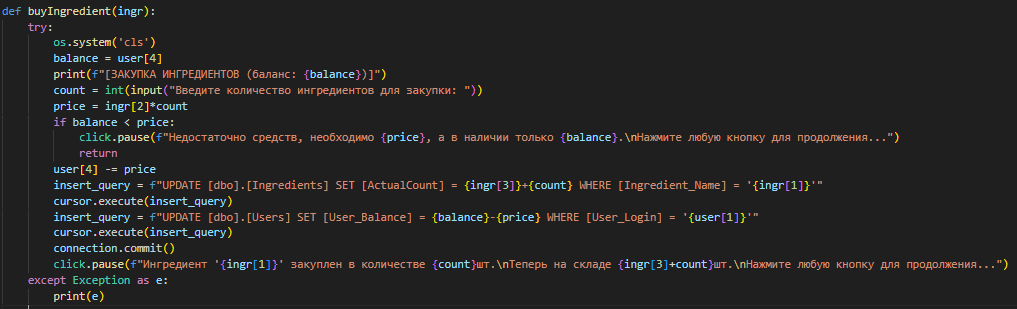


Рисунок – Метод buyIngredient

1. Метод изменение данных игредиента.

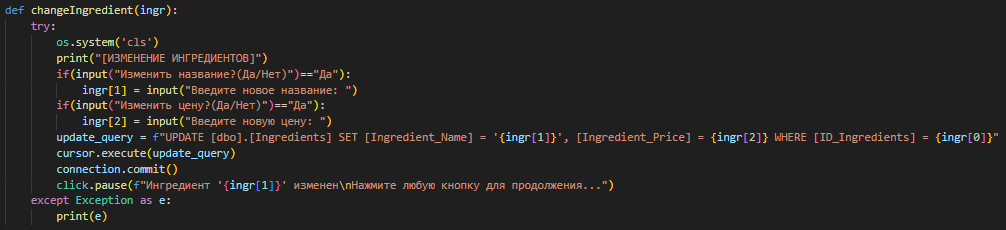


Рисунок – Метод changeIngredient

1. Метод удаления ингредиента

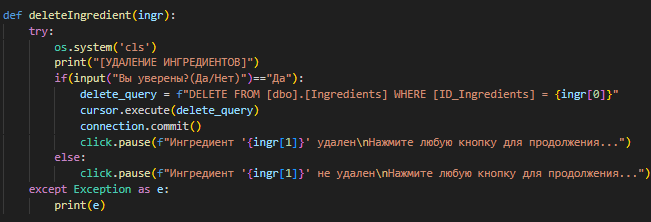


Рисунок – Метод deleteIngredient

1. Метод добавления ингредиента.

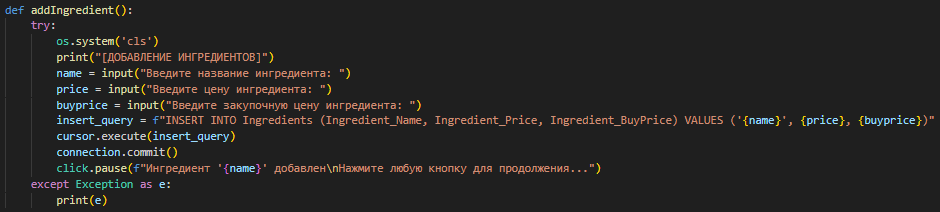


Рисунок – Метод addIngredient

1. Метод просмотра истории пользователя.

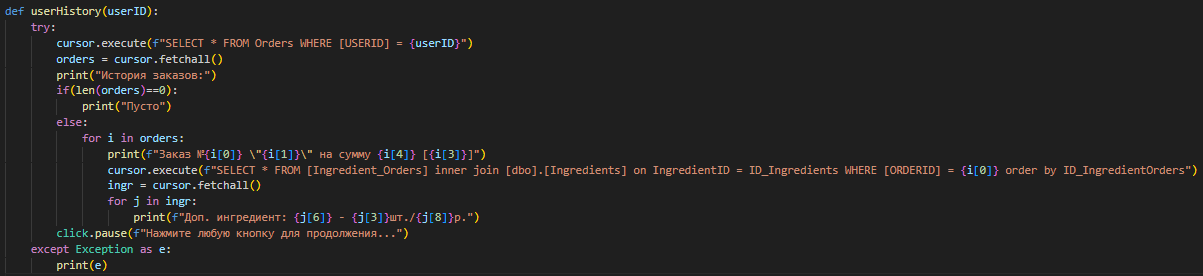


Рисунок – метод userHistory

1. Метод реализации интерфейса администратора.

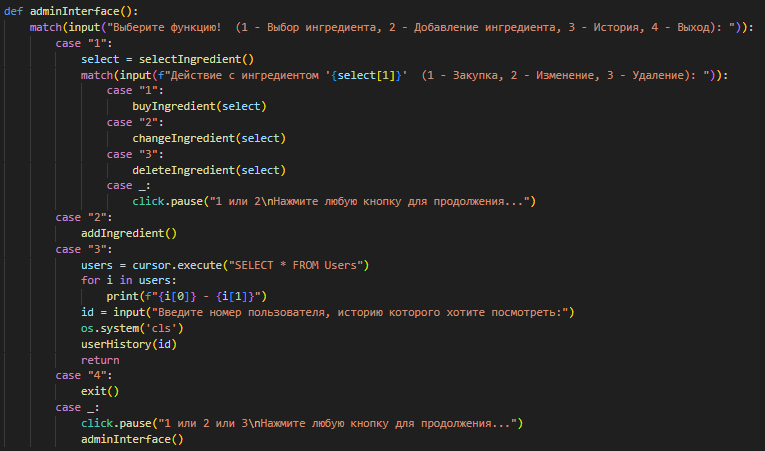


Рисунок – Метод adminInterface

1. Вспомогательные методы для расчёта цены ингредиентов, количества на складе ингредиентов и расчёта количества бесплатных бургеров по акции, запроса на «Да», расчёта скидки на заказ.

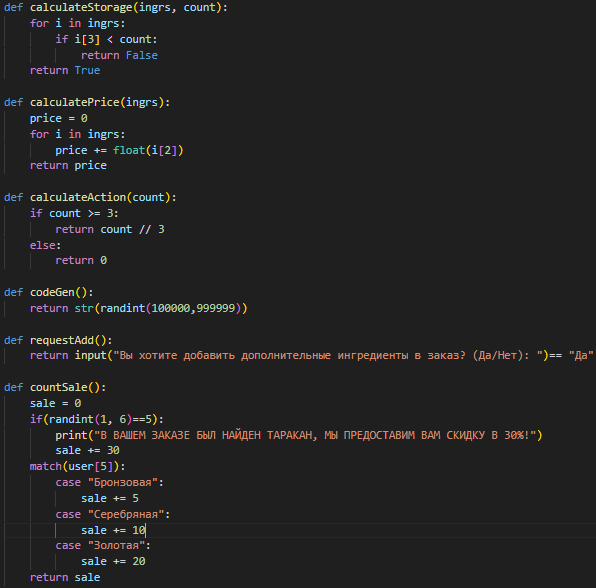


Рисунок – Вспомогательные методы

1. Метод совершения сбора и совершения заказа.



Рисунок – Метод productBuy

1. Метод реализации интерфейса пользователя.

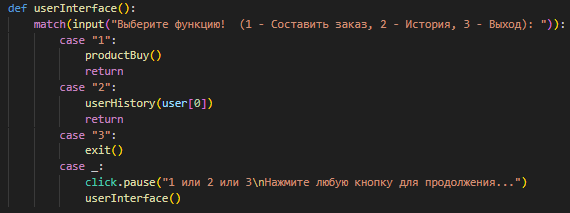


Рисунок – Метод userInterface

1. Написание основной точки входа в программу.

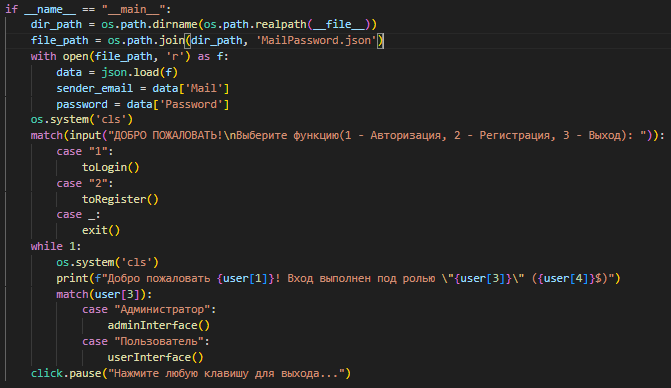


Рисунок – Функция «main»

Результат работы программы:

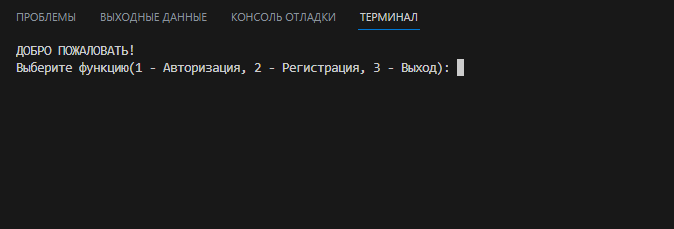


Рисунок – Приветствие и выбор функции

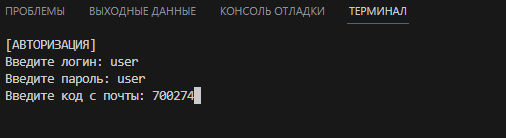


Рисунок – Авторизация

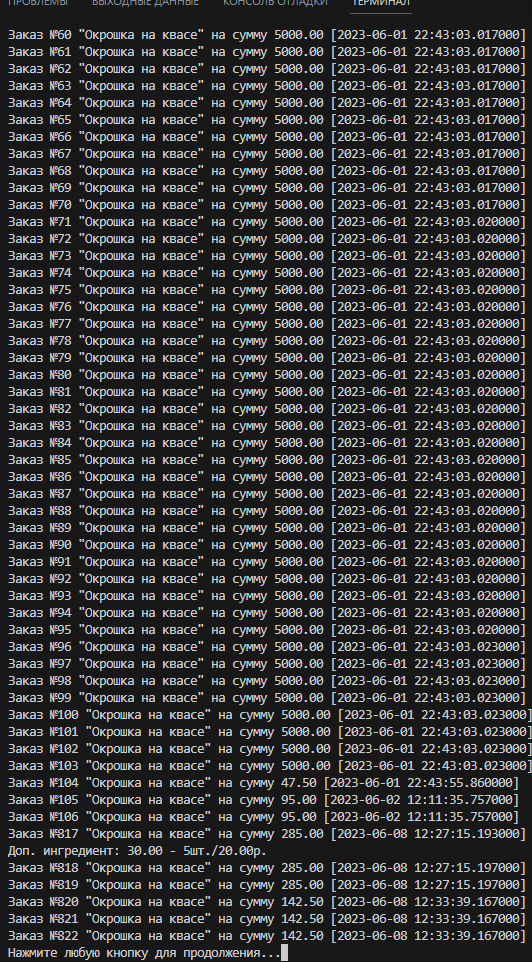


Рисунок – Отображение истории

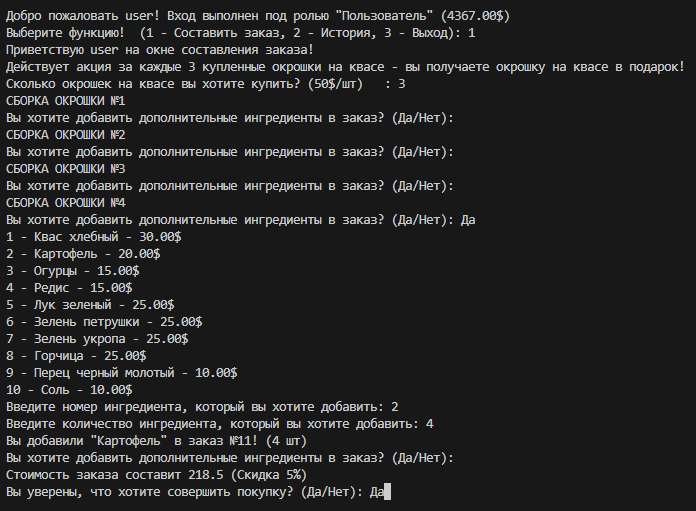


Рисунок – Составление заказа

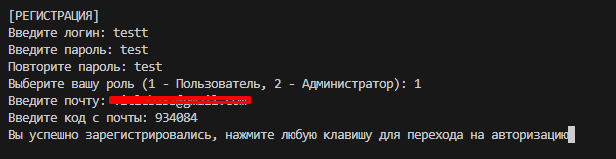


Рисунок – Регистрация

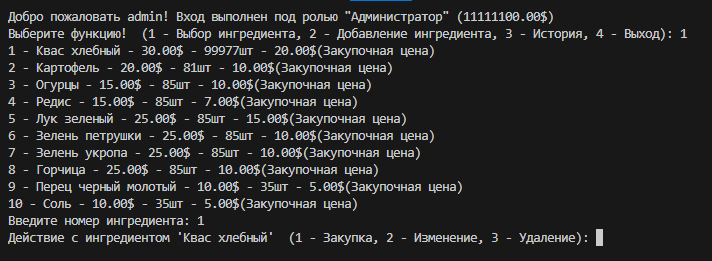


Рисунок – Выбор ингредиента

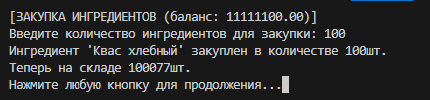


Рисунок – Закупка ингредиентов

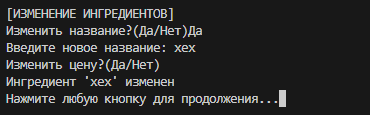


Рисунок – Изменение ингредиентов

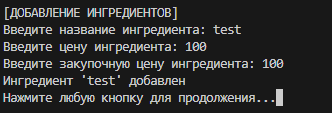


Рисунок – Добавление ингредиента

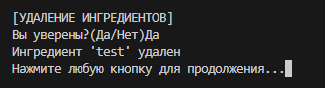


Рисунок – Удаление ингредиента

Вывод: в ходе практической работы была создана программа «мини-бизнес» на языке программирования python, в которой присутствует: сбор своего блюда, регистрация/авторизация с подтверждением кода по почте, реализация пользователя с балансом и ролью (Администратор/Пользователь), акция, скидка за попадание в блюдо таракана, чек, история покупок и карта лояльности.

# Практическая работа №4.

Цель работы: провести разведочный анализ данных по датасету.

Процесс выполнения работы:

1. Загрузка и импорт необходимых для дальнейшей работы с датасетами библиотек pandas и numpy.

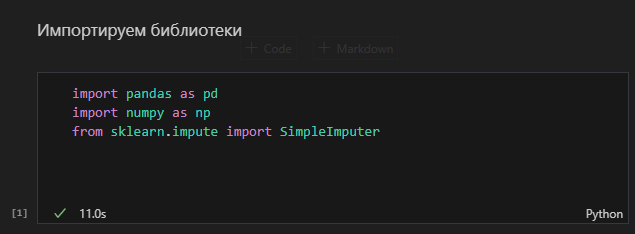


Рисунок – Импорт библиотек

1. Загрузка датасета считыванием csv файла с вариантом 14 в переменную df.

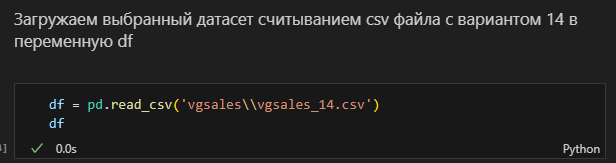


Рисунок – Считывание датасета

1. Вывод количества строк и колонок.

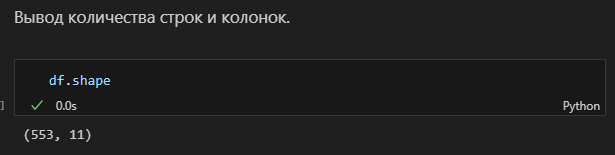


Рисунок – Вывод строк и колонок

1. Вывод первый пяти строк и последней строки.

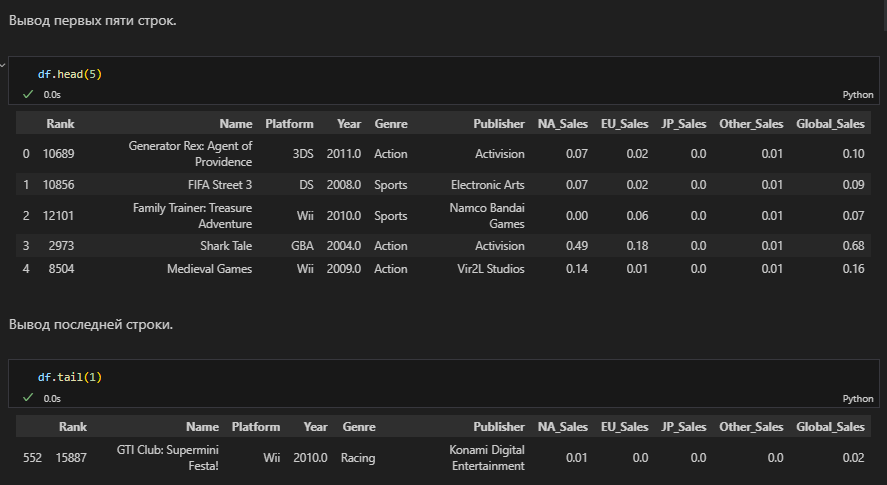


Рисунок – Вывод первых и последних строк

1. Вывод информации и статистики о датасете.

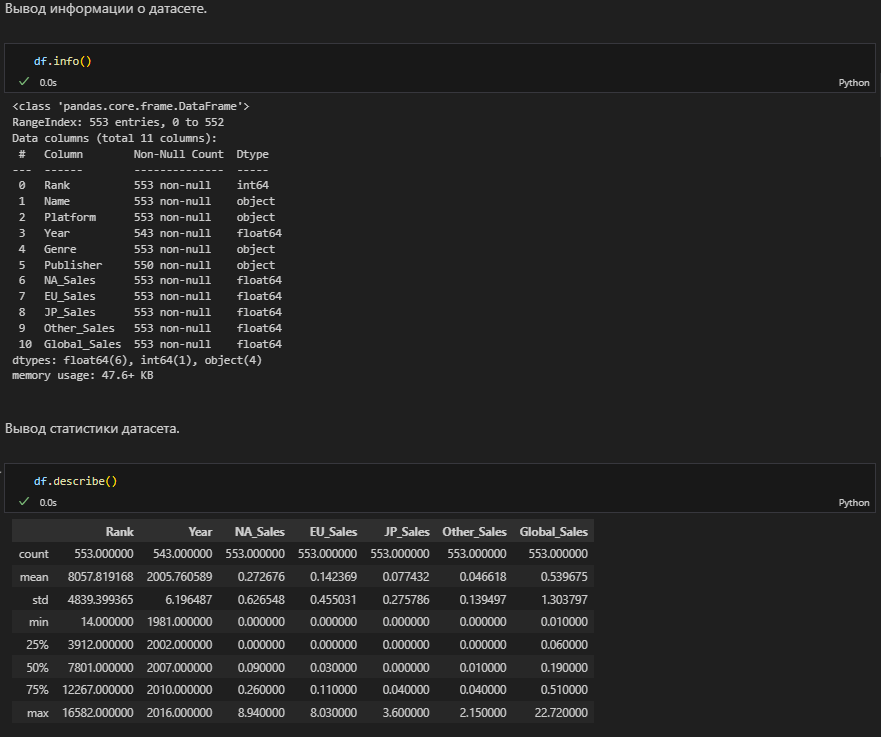


Рисунок – Вывод информации и статистики

1. Вывод количества и процентов уникальных игр.

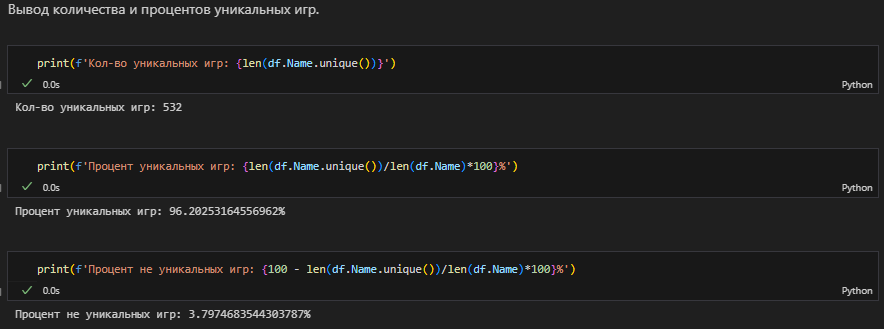


Рисунок – Вывод количества и процентов уникальных игр

1. Поиск строк только со столбцом "PC".

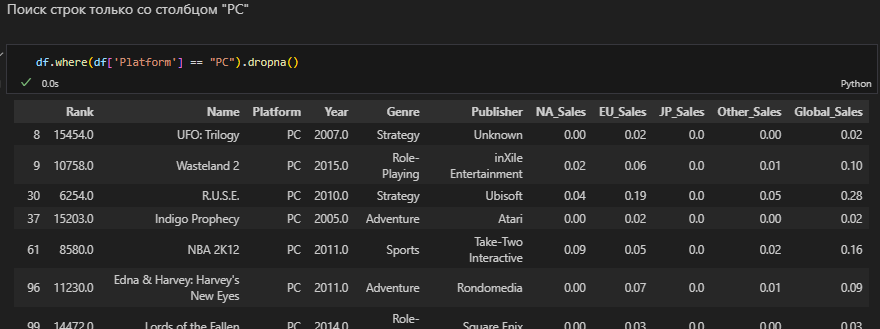


Рисунок – Поиск по платформе

1. Создание новой переменной test, ей присваиваем датасет из переменной df, но уже с дропом дубликатов в колонках "Name" и "Platform".

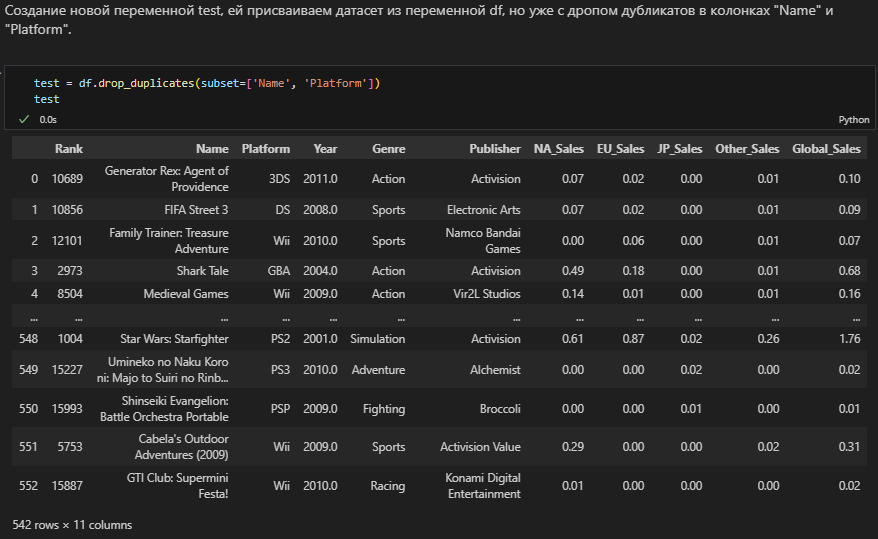


Рисунок – Удаление дупликатов

1. Замены колонки Platform с значений PC на значения каммпуктер.

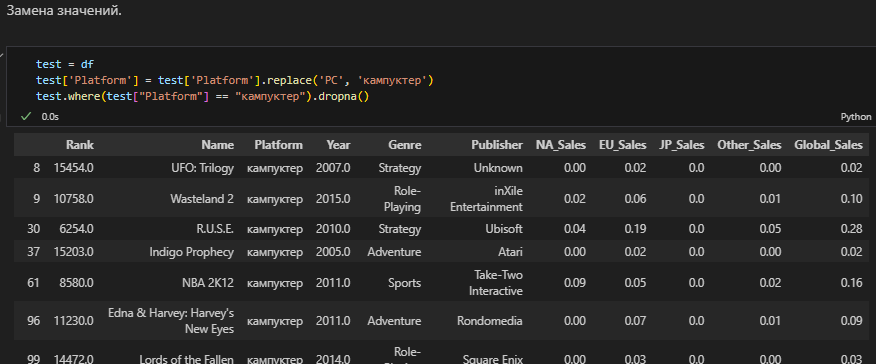


Рисунок – Замена значений

1. В качестве целевой переменной я выбрал столбец "Name", поскольку данный столбец хранит в себе уникальные значения.

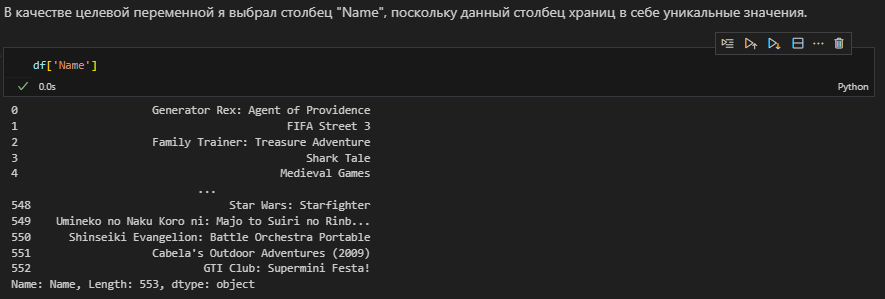


Рисунок – Выбор и обоснование признака

1. Использование простого подстановщика, чтобы заполнить все недостающие значение в столбце Year набора данных.

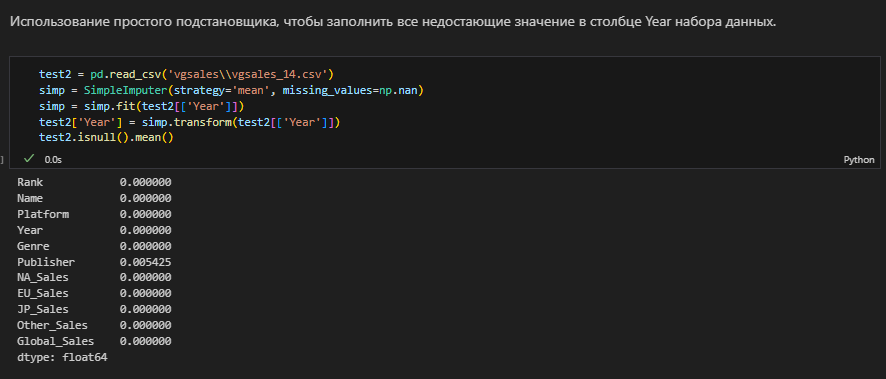


Рисунок – Использование подстановщика

1. Сохранение датасета.

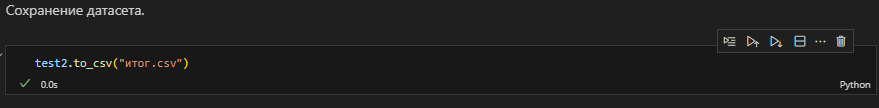


Рисунок – Сохранение датасета

Вывод: в ходе выполнения практической работы был произведен разведочный анализ данных по датасету.

# Практическая работа №5.

Цель работы: Составить графики, проанализировать их и провести кластеризацию.

Процесс выполнения работы:

1. Установка библиотеки seaborn и добавление библиотек seaborn и pyplot для построения графиков, загрузка датасета.

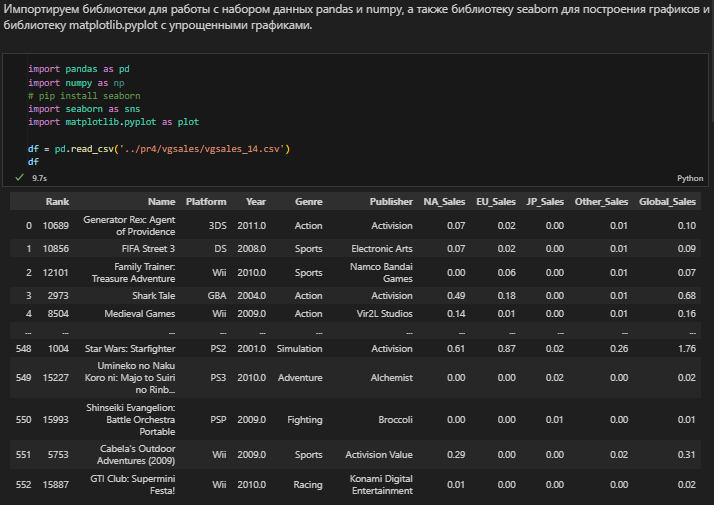


Рисунок – Подключение библиотек и считывание датасета

1. Создание графика countplot.

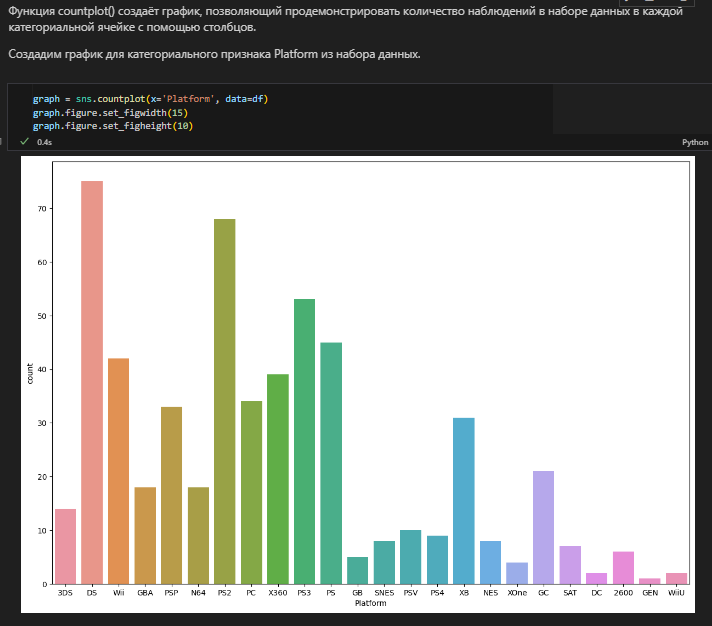


Рисунок – График countplot

1. Создание графика barplot.

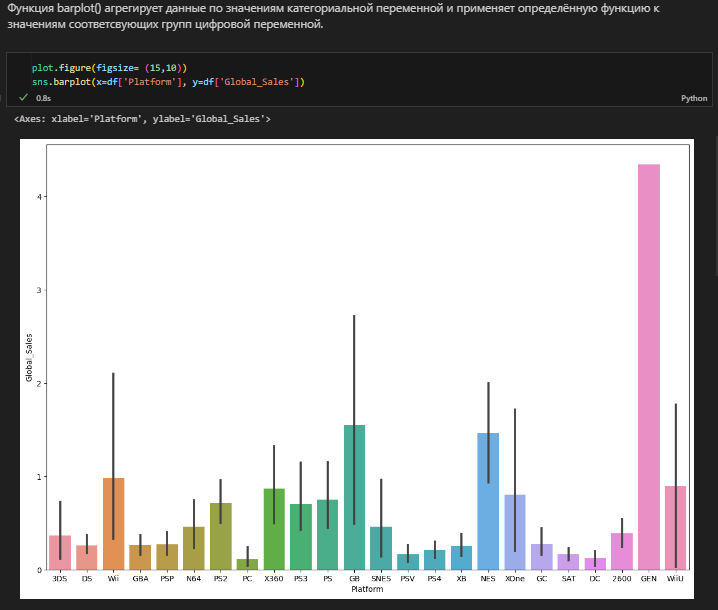


Рисунок – График barplot

1. Создание графика boxplot.

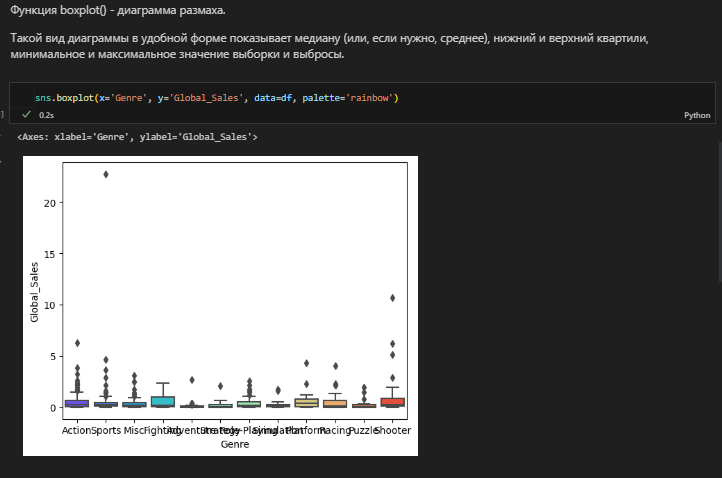


Рисунок – График boxplot

1. Создание графика heatmap.



Рисунок – График heatmap

1. Создание графика pairplot.

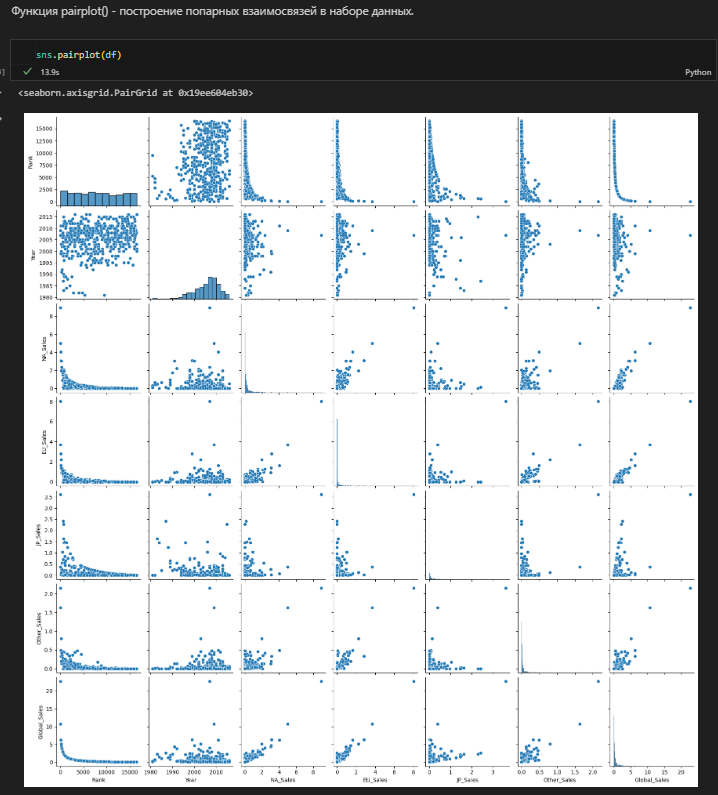


Рисунок – График pairplot

1. Создание графика pairgrid.

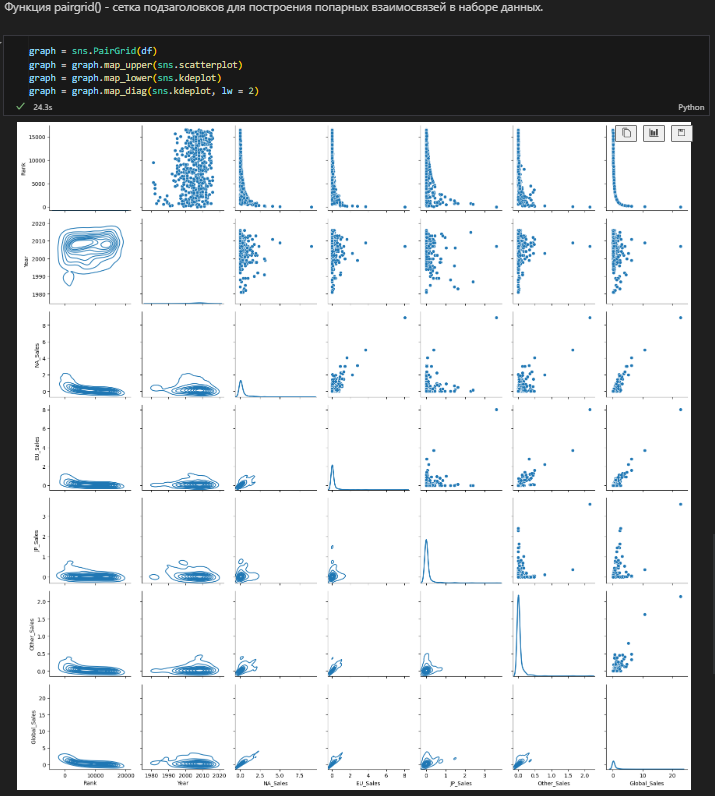


Рисунок – График pairgrid

1. Кластеризация числовых значений.

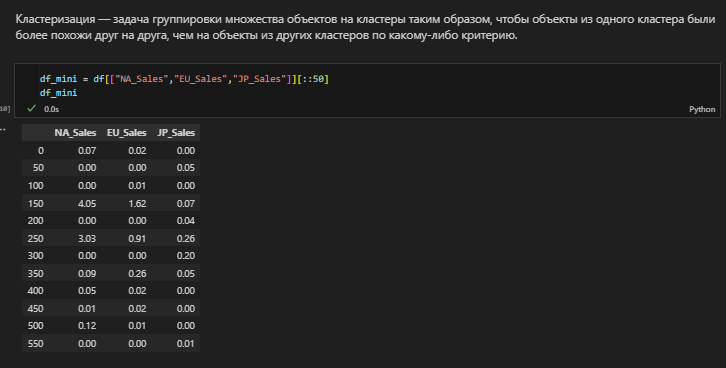


Рисунок – Кластеризация данных

Вывод: в ходе выполнения практической работы были составлены графики, проанализированы и проведена кластеризация.

# Практическая работа №6.

Цель работы: Преобразовать признаки и разбить датасет на обучающую и тестовую выборку.

Процесс выполнения работы:

1. Импорт файла и библиотек.

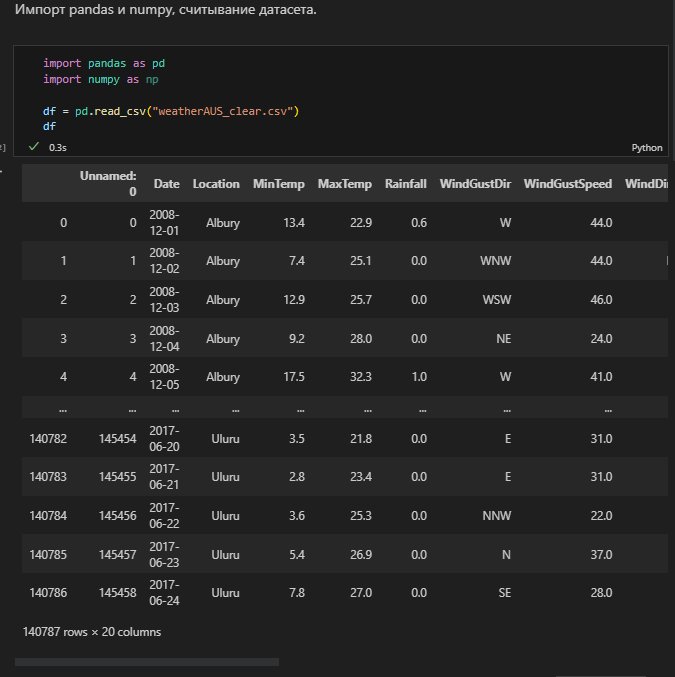


Рисунок – Импорт файла и библиотек

1. Работа с OrdinalEncoder.

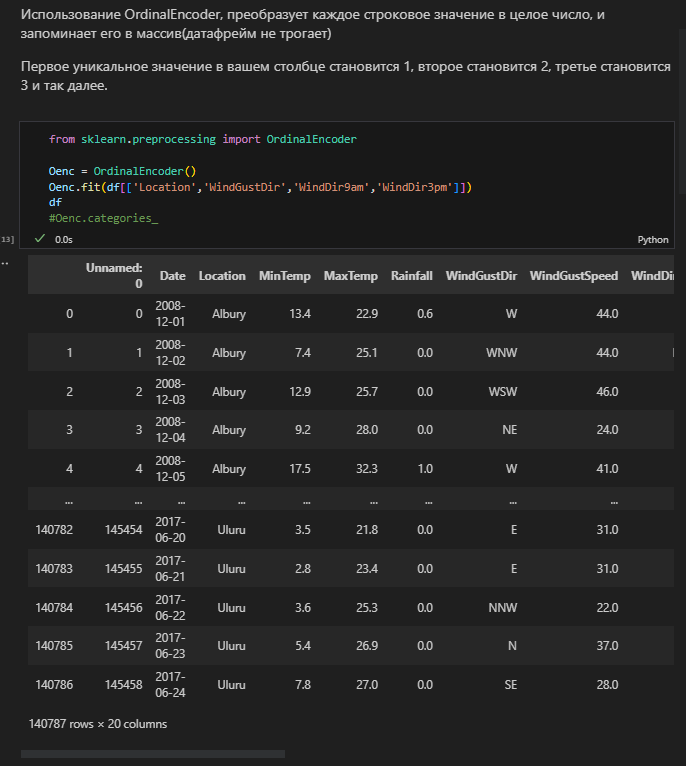


Рисунок – Работа OrdinalEncoder (1)

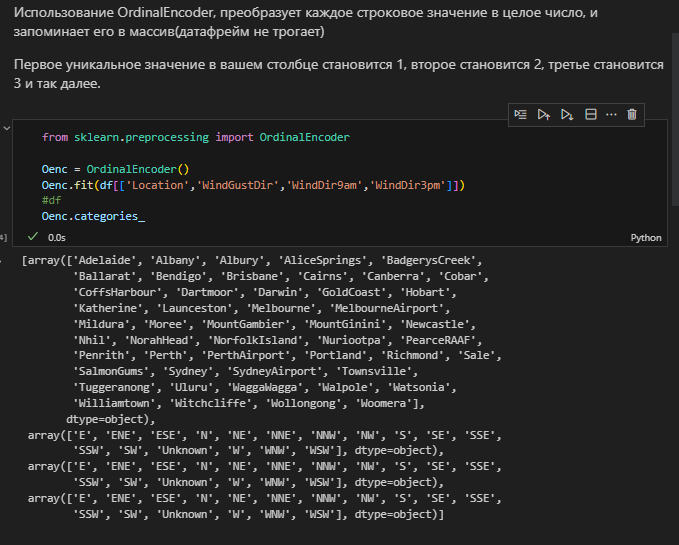


Рисунок – Работа OrdinalEncoder (2)

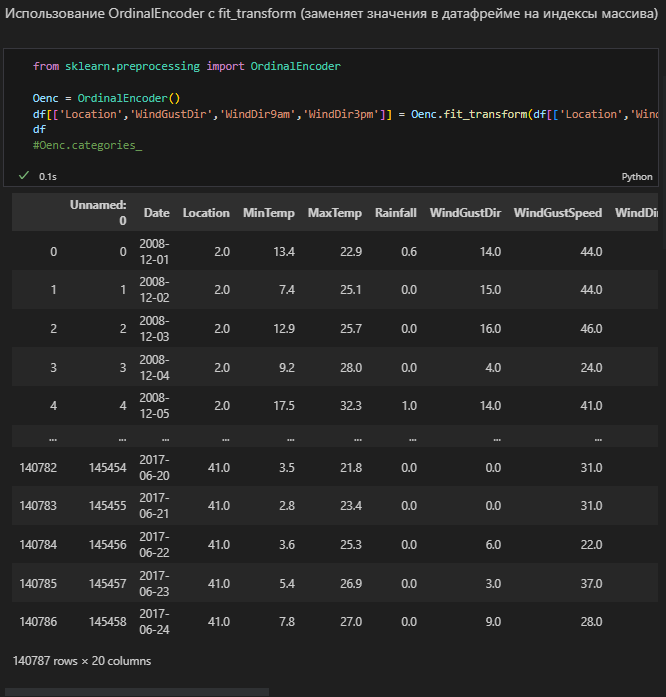


Рисунок – Работа OrdinalEncoder (3)

1. Использование LabelEncoder.

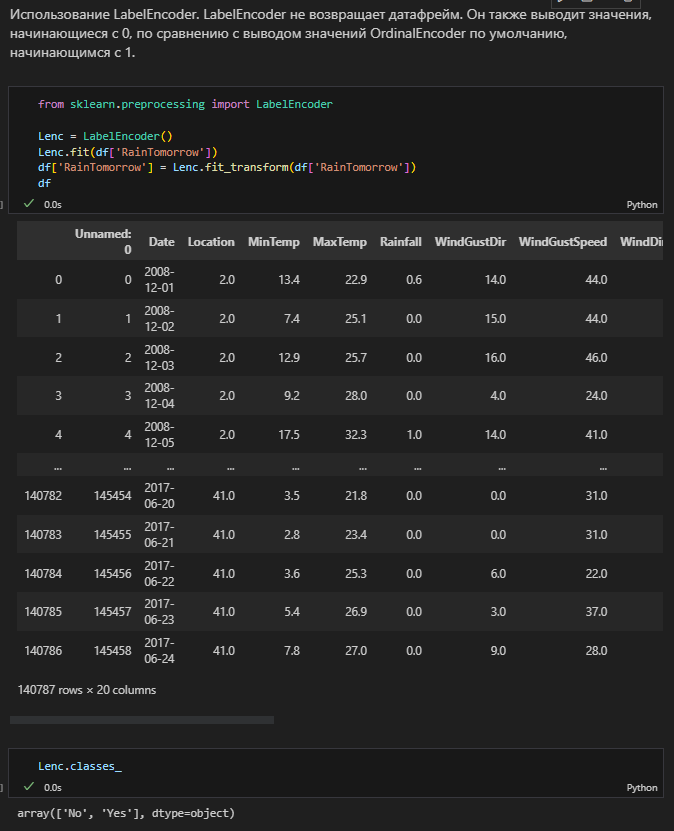


Рисунок – Работа LabelEncoder

1. Использование OneHotEncoder.

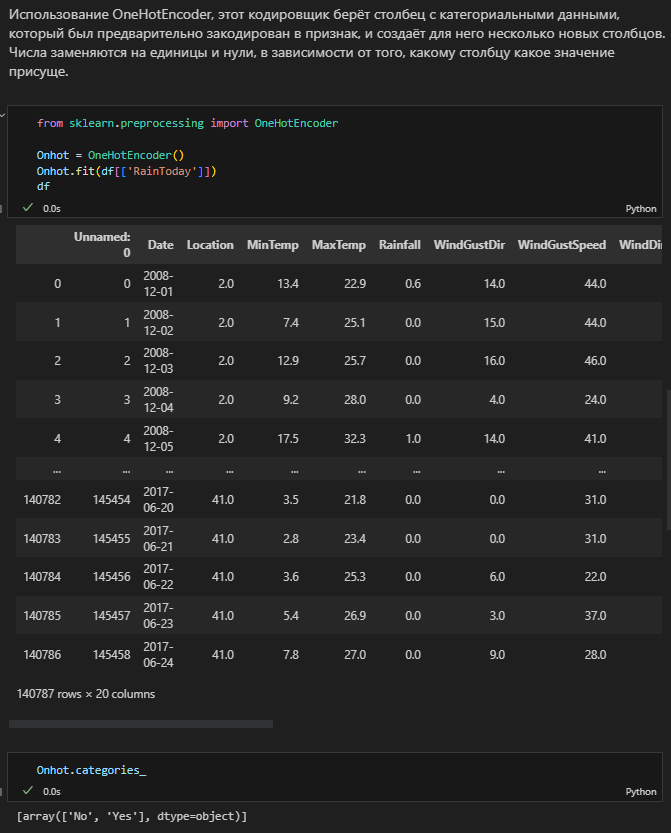


Рисунок – Работа OneHotEncoder

1. Использование Normalizer.

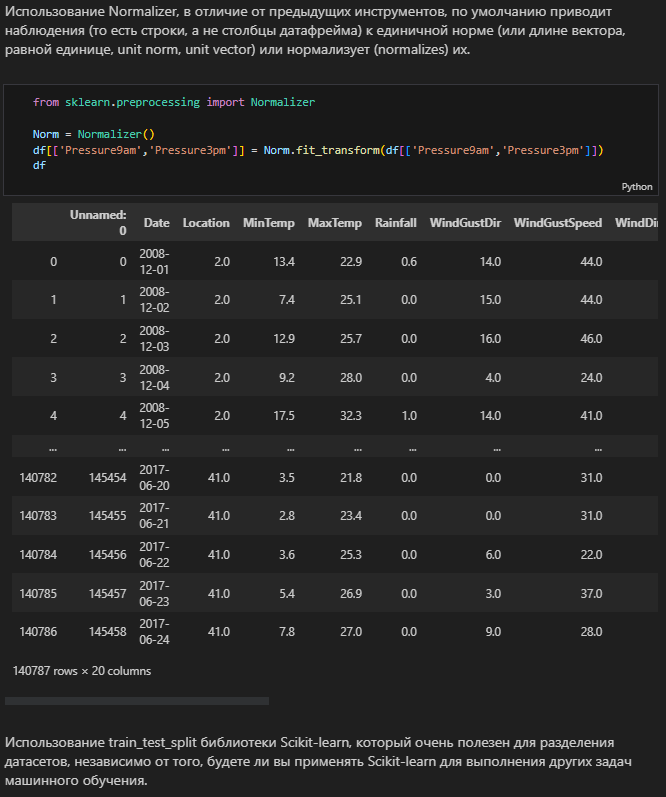


Рисунок – Работа Normalizer

1. Разделение датасета на обучающую и тестовую выборку.

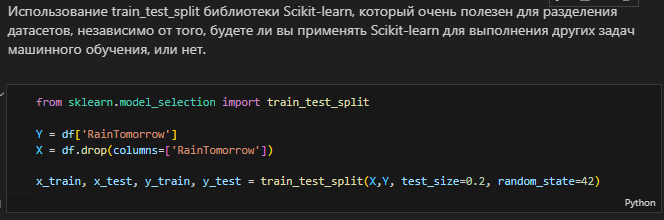


Рисунок – Работа train\_test\_split

Вывод: в ходе выполнения практической работы были преобразованы признаки и разбит датасет на обучающую и тестовую выборку.

# Практическая работа №7.

Цель работы: обучить модель "Дерево решений" и проверить её точность.

Процесс выполнения работы:

1. Импорт всех необходимых библиотек для работы.

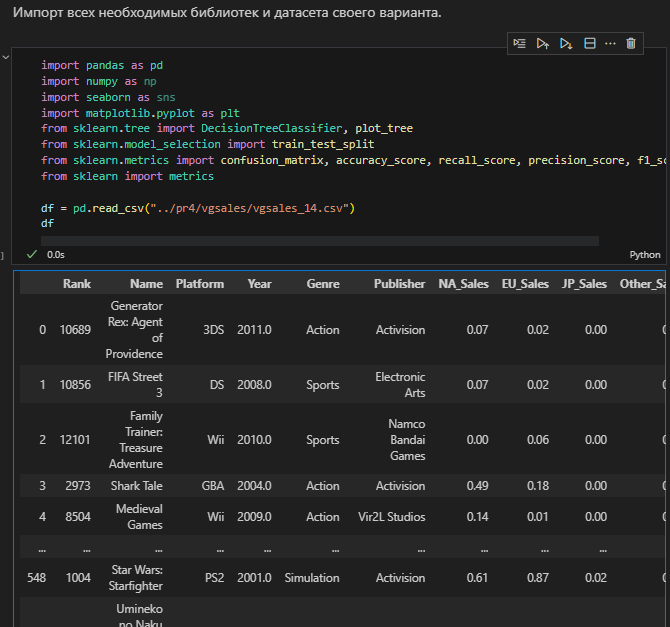


Рисунок – Импорт библиотек

1. Преобразование датасета.

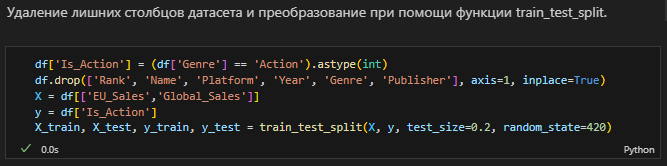


Рисунок – Преобразование датасета

1. Код для построения дерева решений с выбранными размерами и параметрами.

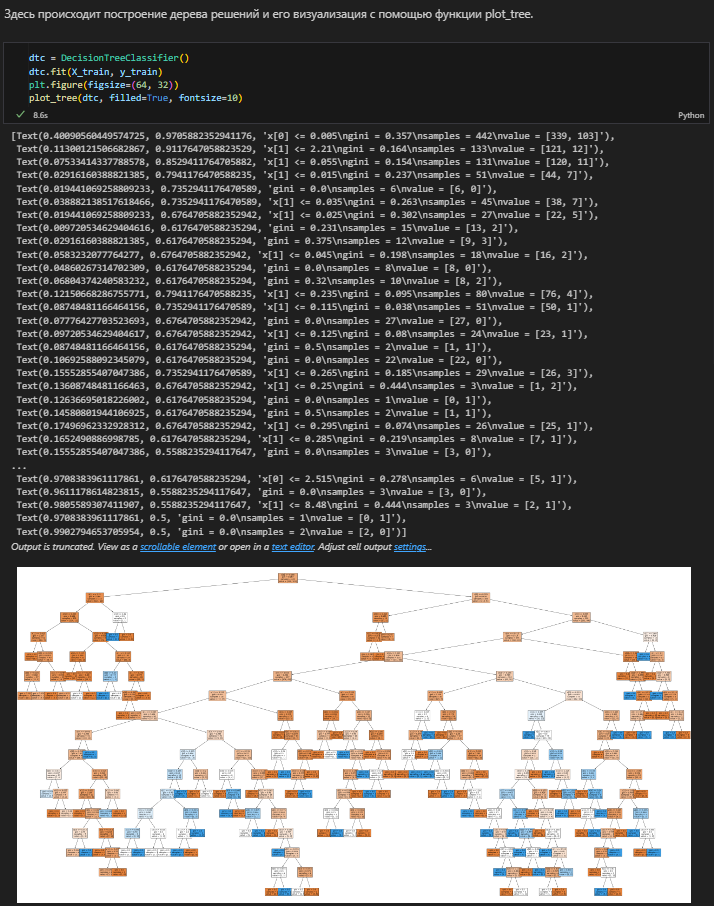


Рисунок – Код построения

1. Вычисление метрик графика.

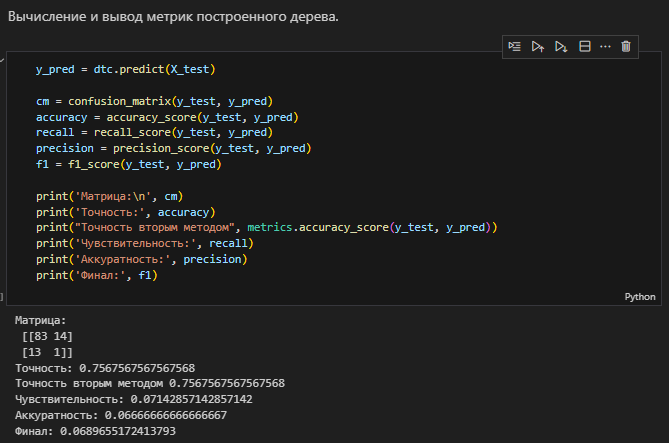


Рисунок – Метрики

1. Код для построения второго дерева, но уже с параметрами.

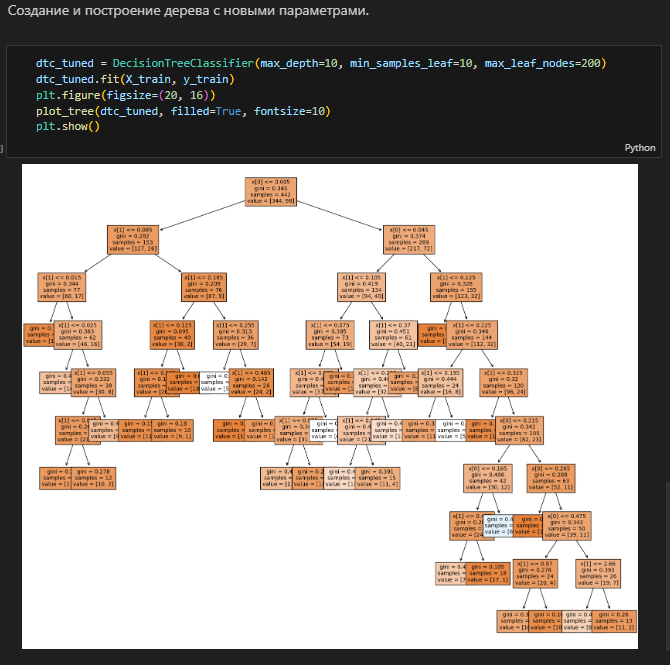


Рисунок – Код создания второго дерева

1. Метрики для второго дерева.

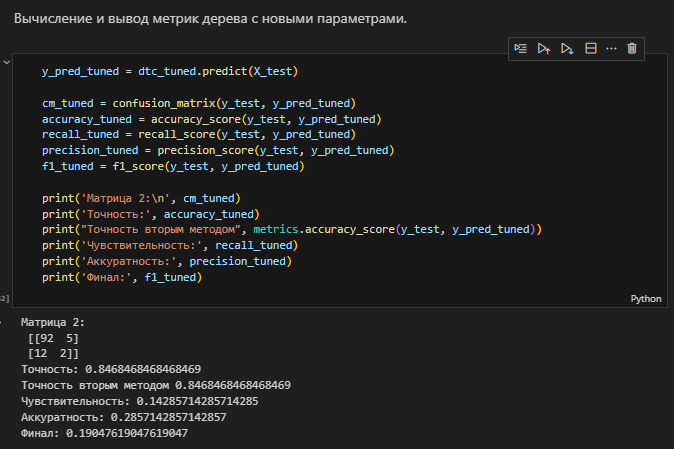


Рисунок – Метрики второго дерева

Вывод: в ходе выполнения практической работы было обучено две модели «Дерево решений» и проверены их метрики, из-за того, что у первой модели не было гиперпараметров их точность отличается.

# Практическая работа №8.

Цель работы: ознакомится с моделями классификации и регрессии.

Процесс выполнения работы:

1. Импорт всех необходимых библиотек для работы и датасета.

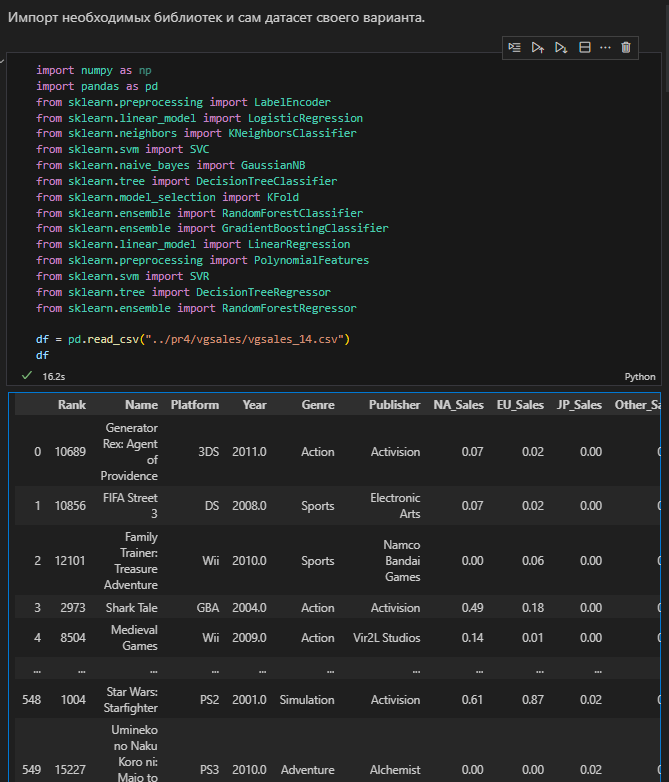


Рисунок 72 – Импорт библиотек и датасета

1. Удаление всех нечисловых столбцов датасета.

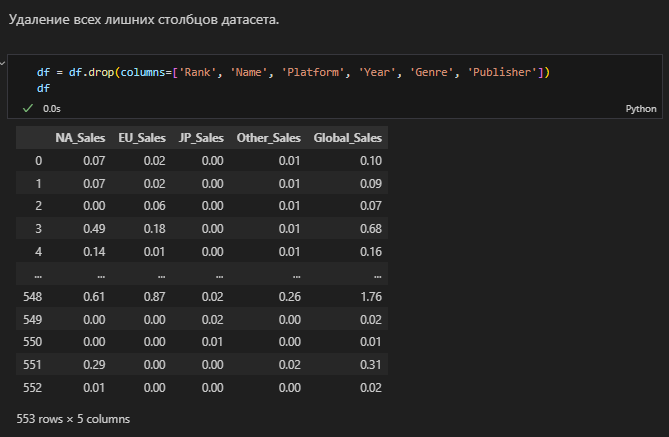


Рисунок 73 – Удаление столбцов

1. Преобразование датасета в целые числа в стобце "Global\_Saldes" при помощи LabelEncoder().

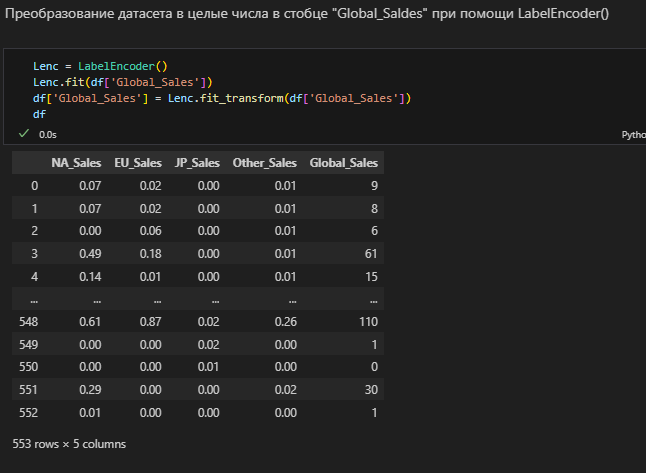


Рисунок 74 – Преобразование при помощи LabelEncoder

1. Создание экземпляров классов для дальнейшей работы с ними.

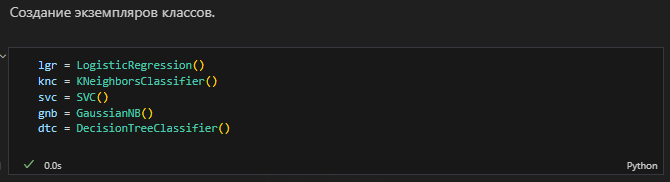


Рисунок 75 – Экземпляры классов

1. Разбивка данных датасета на X и Y, создание экземпляра класса KFold с параметрами.

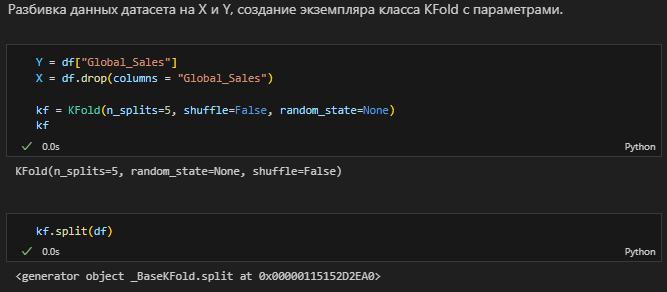


Рисунок 76 – Разбивка данных датасета

1. Обучение.

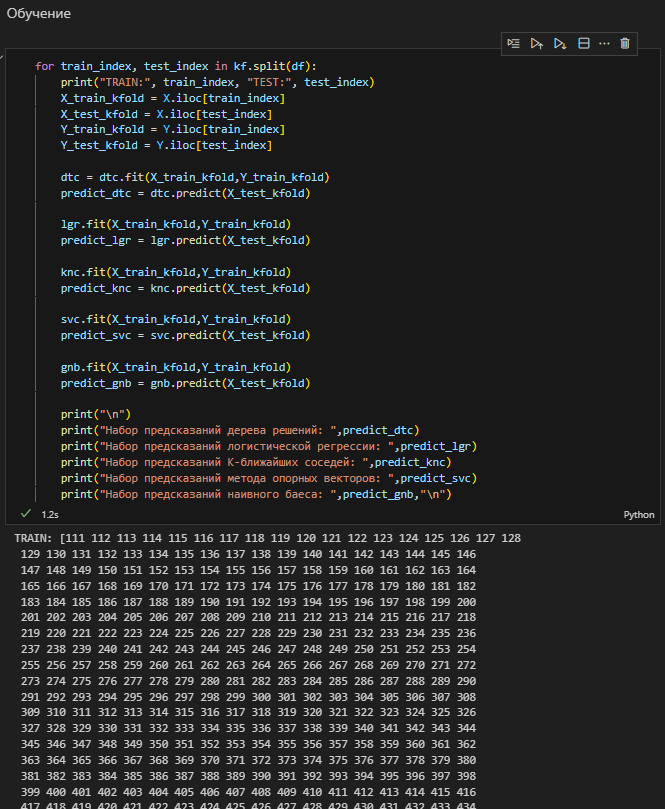


Рисунок 77 – Обучение.

1. Создание экземпляров и преобразование столбца «EU\_Sales».

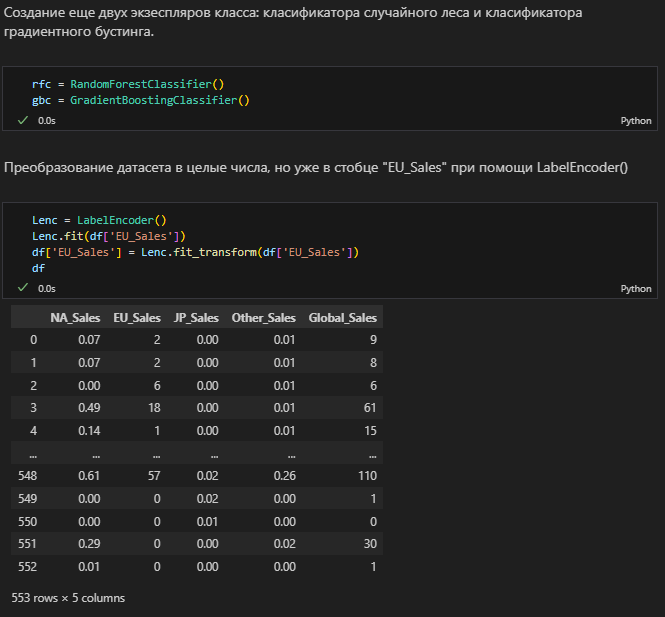


Рисунок 78 – Создание и преобразование

1. Повторный вызов тех же моделей для обучения, но уже для других типов классификаций.

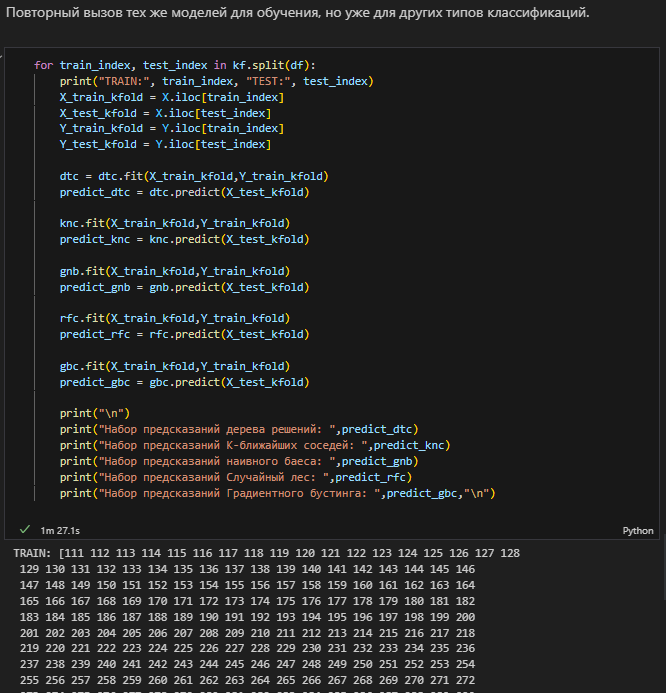


Рисунок 79 – Обучение 2

1. Создание экземпляров линейной, полиноминальной регрессии, регрессии опорных векторов, дерева решений и случайного леса.

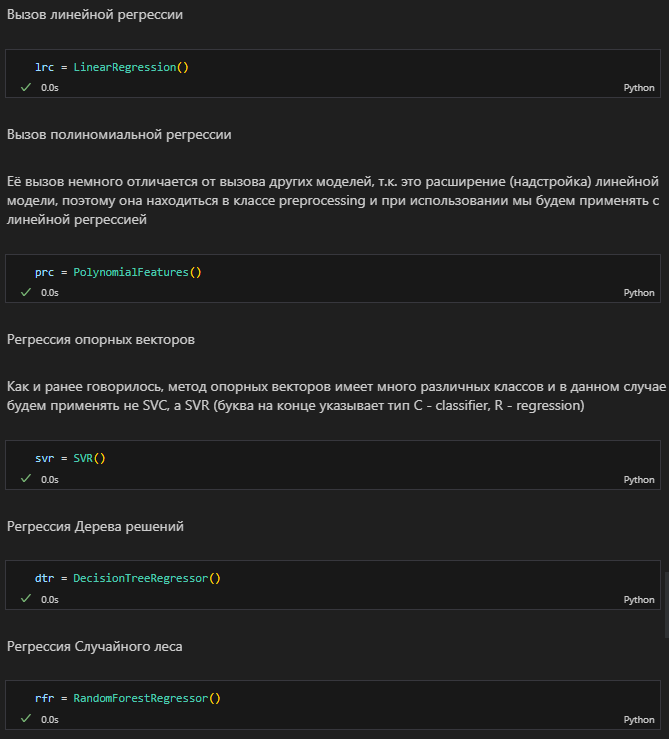


Рисунок 80 – Создание экземпляров регрессий

1. Преобразование датасета в целые числа, но уже в стобце "NA\_Sales" при помощи LabelEncoder() и разбивка на Y и X.

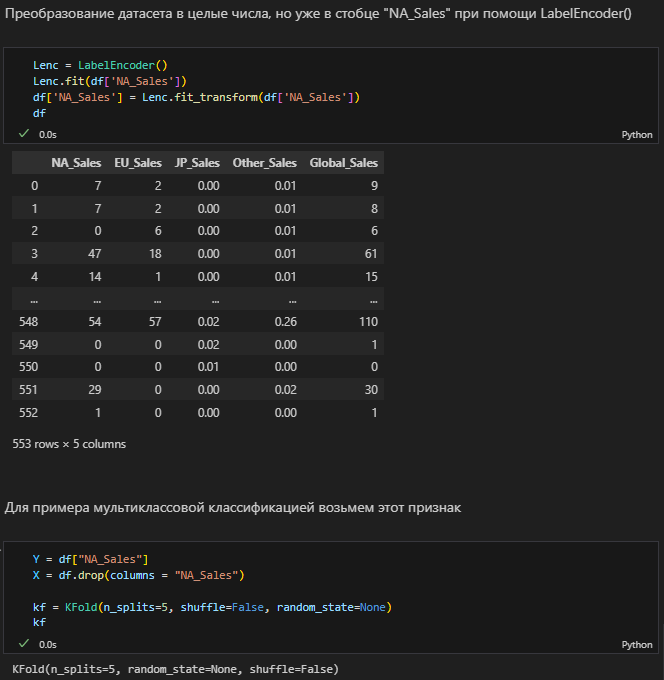


Рисунок 81 – Преобразование и разбивка

1. Обучение тоже, что было выше, но с параметрами, которые остались.

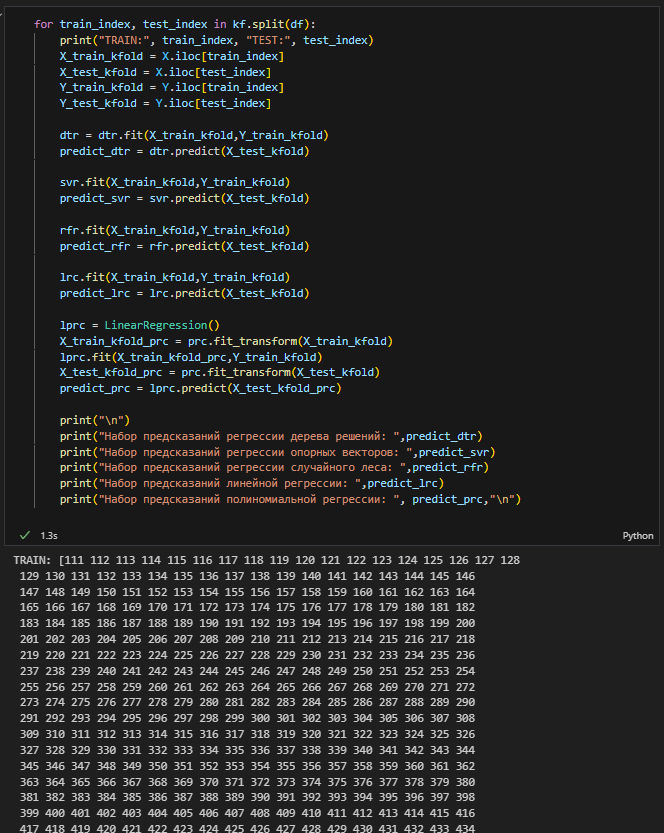


Рисунок 83 – Обучение 3

Вывод: в ходе выполнения практической работы были разобраны на примере регрессии и классификации.