Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

|  |
| --- |
| Институт Высоких Технологий |
| наименование института |
| Кафедра автоматизированных систем |
| наименование кафедры |

**Лабораторная работа**

по дисциплине:

|  |
| --- |
| **Технологии искусственного интеллекта** |
| Анализ данных с использованием Python |

(Наименование темы)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил студент группы: | ИСТбп-16 |  |  |  | Литвинов В.А. |
|  |  |  |  |  | Фамилия Имя Отчество |
|  |  |  |  |  |  |
| Проверил: |  |  |  |  | Пестерев Д.В. |
|  | должность |  | подпись |  | Фамилия Имя Отчество |

Иркутск 2020 г

ОГЛАВЛЕНИЕ

[**ВВЕДЕНИЕ** 3](#_Toc27478467)

[**1.** **Постановка задачи и исходные данные** 4](#_Toc27478468)

[**2.** **Результат работы программы** 5](#_Toc27478469)

[**3.** **Листинг реализации генетического алгоритма** 8](#_Toc27478470)

[**ЗАКЛЮЧЕНИЕ** 11](#_Toc27478471)

# **ВВЕДЕНИЕ**

Язык программирования Python в последнее время все чаще используется для анализа данных, как в науке, так и коммерческой сфере. Этому способствует простота языка, а также большое разнообразие открытых библиотек.

В данной лабораторной работы мы разберем простой пример исследования и классификации данных с использованием некоторых библиотек на Python. Для исследования, нам понадобится выбрать интересующий нас набор данных (DataSet). DataSet обычно представляет собой файл с таблицей в формате JSON или CSV.

Наша цель будет не получить исчерпывающие ответы на главный вопрос жизни, вселенной и всего такого, а показать простоту обработки достаточно большого объема данных средствами Python.

1. **Постановка задачи и исходные данные**

Необходимо произвести анализ произвольного датасета по нескольким критериям.

Для выполнения работы мной был выбран датасет “Продажа видеоигр”, который содержит список видеоигр с объемом продаж более 100 000 копий по следующим характеристикам:

* Ранг - Рейтинг общих продаж
* Название - Название игры
* Платформа - платформа выпуска игр (т.е. ПК, PS4 и т. Д.)
* Год - год выпуска игры
* Жанр - Жанр игры
* Издатель - издатель игры
* NA\_Sales - Продажи в Северной Америке (в миллионах)
* EU\_Sales - Продажи в Европе (в миллионах)
* JP\_Sales - Продажи в Японии (в миллионах)
* Other\_Sales - Продажи в остальном мире (в миллионах)
* Global\_Sales - общий объем продаж по всему миру.

# **Результат работы программы**

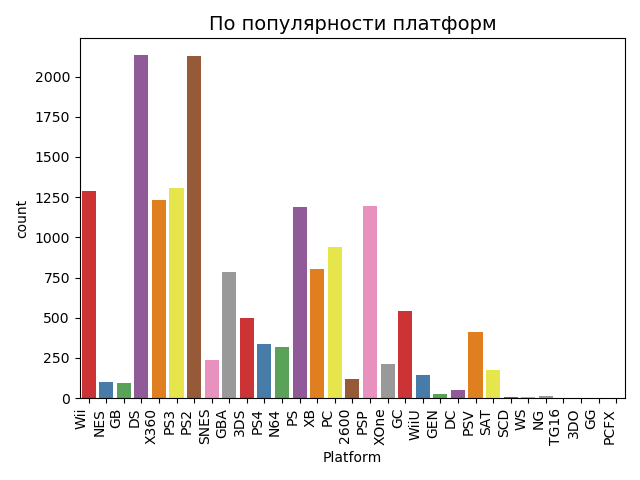


Рисунок 1 – Платформы игр

Согласно данному графику можно сделать вывод что большинство игр с объемом продаж выходили на платформу DS и на втором месте PS2.

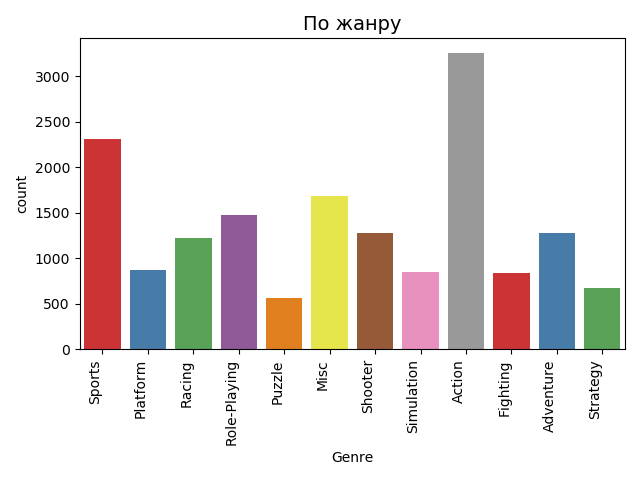


Рисунок 2 – Кол-во игр по жанрам

По этому графику можно сделать вывод о том, что большинство игр в жанре – Action. И на втором месте Sports.

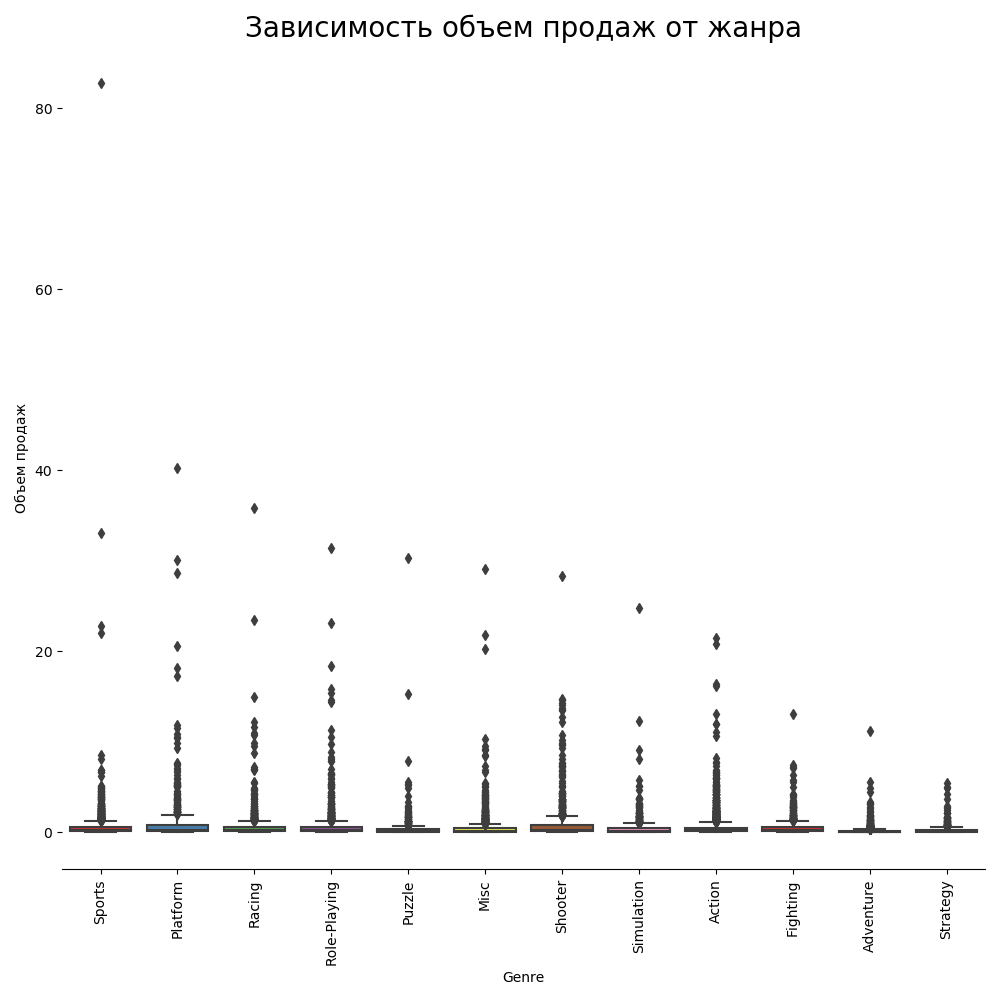


Рисунок 3 – Зависимость объема продаж от жанра

Исходя из данных наибольшее количество проданных игр соответствует категории Sports.

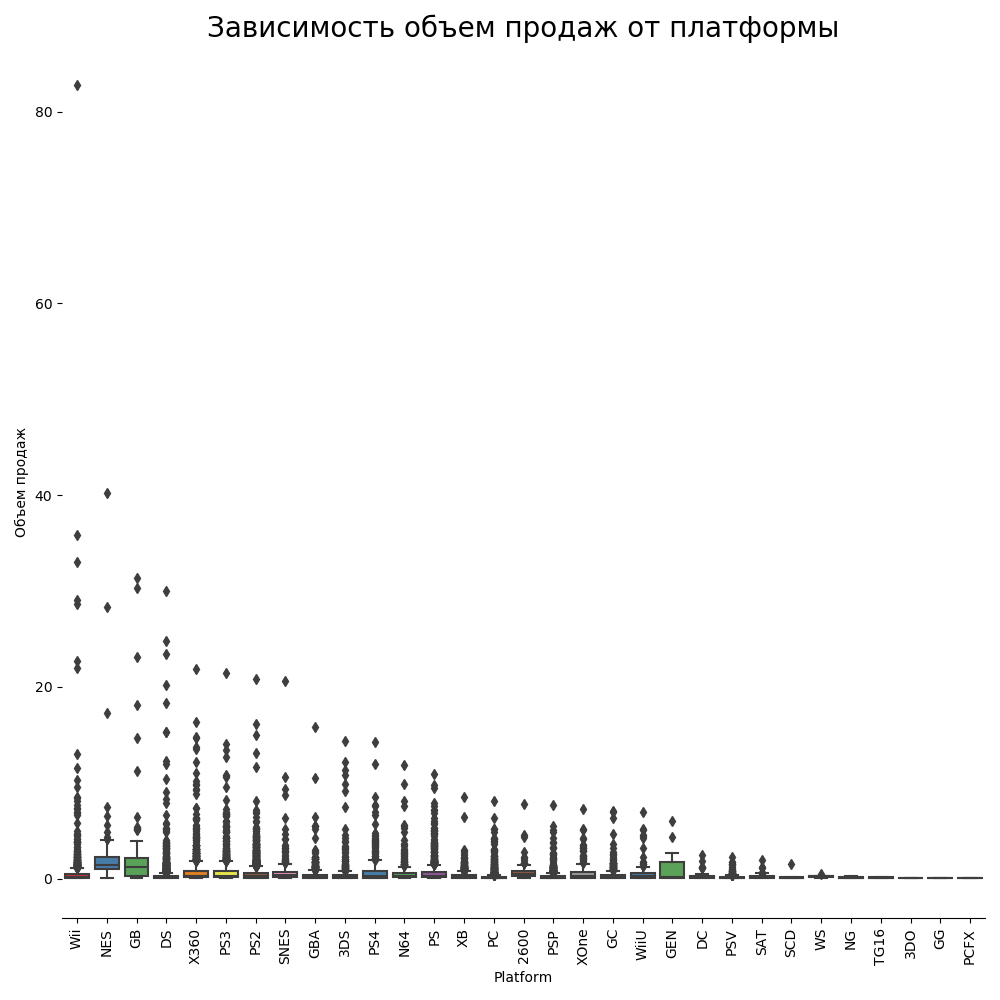


Рисунок 4 – График зависимости объема продаж от жанра

Исходя из данных наибольшее количество проданных игр на платформе Wii.

# **Листинг реализации генетического алгоритма**

|  |
| --- |
| 1. import sys; sys.path 2. import pandas as pd 3. import numpy as np 4. import seaborn as sns 5. import matplotlib.pyplot as plt 6. from matplotlib import cm 7. from mpl\_toolkits.mplot3d import Axes3D 8. PLOT\_LABEL\_FONT\_SIZE = 12 9. def getColors(n): 10. COLORS = [] 11. cm = plt.cm.get\_cmap('hsv', n) 12. for i in np.arange(n): 13. COLORS.append(cm(i)) 14. return COLORS 15. def dict\_sort(my\_dict): 16. keys = [] 17. values = [] 18. my\_dict = sorted(my\_dict.items(), key=lambda x:x[1], reverse=True) 19. for k, v in my\_dict: 20. keys.append(k) 21. values.append(v) 22. return (keys,values) 23. data = pd.read\_csv('./vgsales.csv', escapechar='`', low\_memory=False) 24. data.head() 25. data.dropna(how ='any', inplace = True) 26. total = data.isnull().sum().sort\_values(ascending=False) 27. percent = (data.isnull().sum()/data.isnull().count()).sort\_values(ascending=False) 28. missing\_data = pd.concat([total, percent], axis=1, keys=['Total', 'Percent']) 29. missing\_data.head(6) 30. g = sns.countplot(x="Platform",data=data, palette = "Set1") 31. g.set\_xticklabels(g.get\_xticklabels(), rotation=90, ha="right") 32. plt.title('По популярности платформ',size = 14) 33. plt.tight\_layout() 34. plt.show(); 35. g = sns.countplot(x="Genre",data=data, palette = "Set1") 36. g.set\_xticklabels(g.get\_xticklabels(), rotation=90, ha="right") 37. plt.title('По жанру',size = 14) 38. plt.tight\_layout() 39. plt.show(); 40. g = sns.catplot(x="Genre",y="Global\_Sales",data=data, kind="box", height = 10 ,palette = "Set1") 41. g.despine(left=True) 42. g.set\_xticklabels(rotation=90) 43. g = g.set\_ylabels("Объем продаж") 44. plt.title('Зависимость объем продаж от жанра',size = 20) 45. plt.tight\_layout() 46. plt.show(); 47. g = sns.catplot(x="Platform",y="Global\_Sales",data=data, kind="box", height = 10 ,palette = "Set1") 48. g.despine(left=True) 49. g.set\_xticklabels(rotation=90) 50. g = g.set\_ylabels("Объем продаж") 51. plt.title('Зависимость объем продаж от платформы',size = 20) 52. plt.tight\_layout() 53. plt.show(); |

## **ЗАКЛЮЧЕНИЕ**

В результате выполнения работы был проведён анализ набора данных с использованием Python. Получены навыки работы с библиотеками анализа данных и построения графиков.