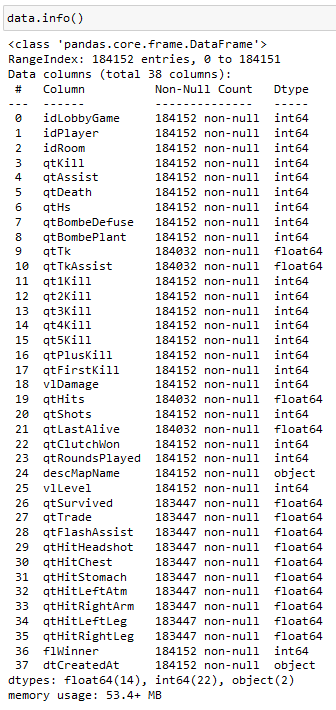
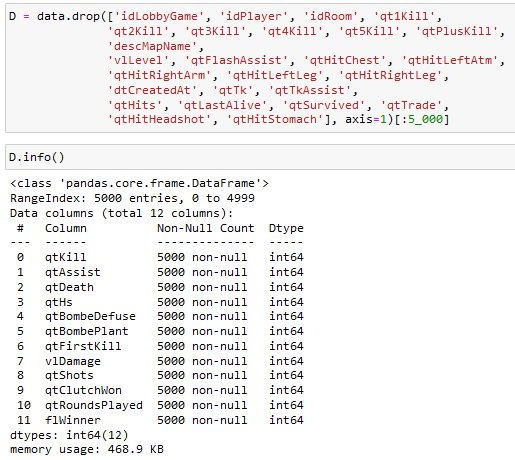
# ВЫПОЛНЕНИЕ ЗАДАЧ

Целью работы было изучить данные из многомерного датасета, используя t-SNE, а также UMAP.

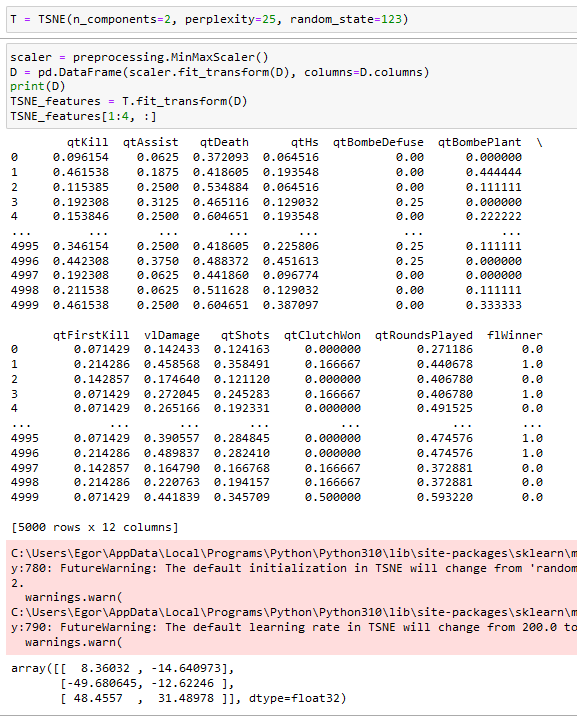
С целью выполнения задачи был найден датасет, содержащий данные в рамках турнира по кс го, в котором, с помощью вышеназванных библиотек, было решено построить визуализацию, разделив данные по параметру названия карты, на которой эти данные были запечетлены.



**Рисунок 1 – Проверка датасета**



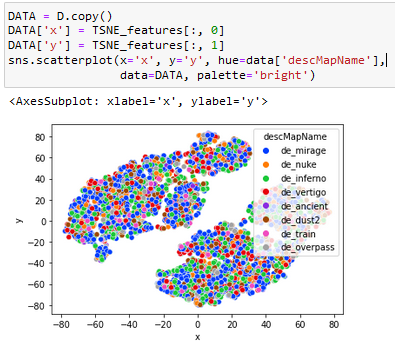
**Рисунок 2 — Создание датасета с нужными параметрами**



**Рисунок 3 — Обучение модели TSNE библиотеки sklearn**

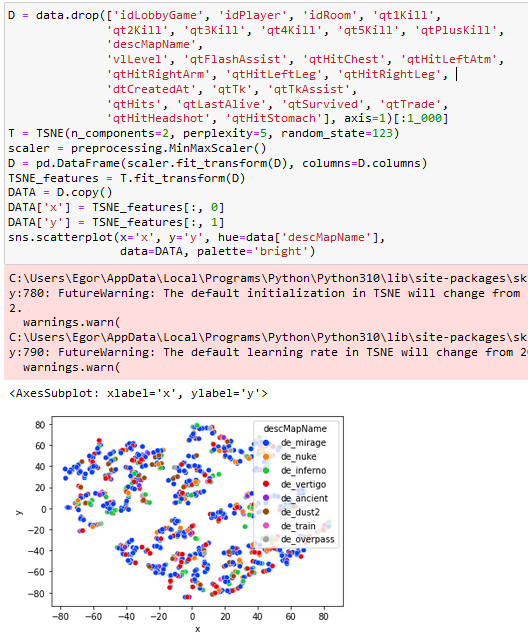
Было решено взять параметры по типу урона, количества попаданий, выигрышей на карте, исключив такие чрезмерные параметры, как, например, попадание в левую ногу, правую руку и так далее.

В результате seaborn.scatterplot построил следующий график.

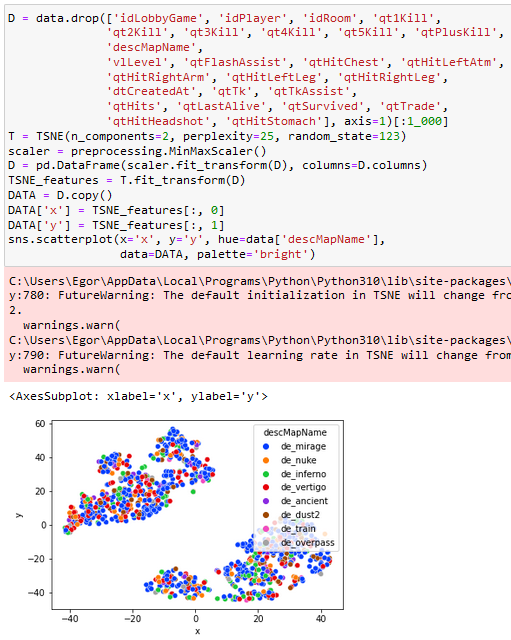


**Рисунок 4 — scatterplot с перплексией в 25**

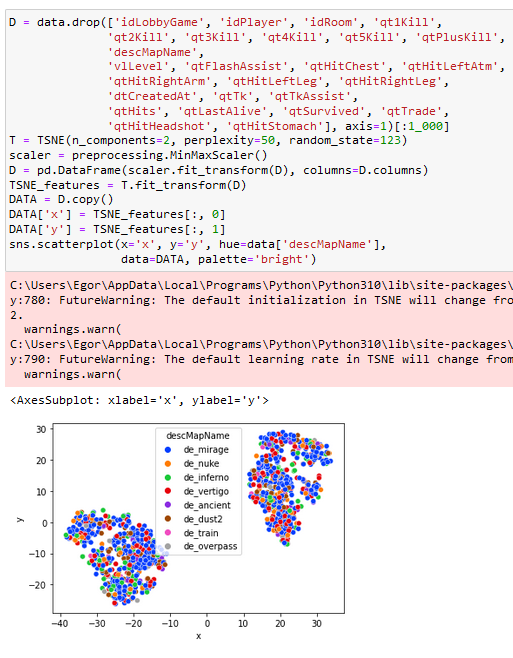
Как мы можем видеть, данные из датасета не очень хорошо показывают себя после обучения. Попробуем построить графики с различным показателем перплексии, сократив количество элементов в выборке до 1000 вместо 5000.



**Рисунок 5 — График с перплексией в 5**



**Рисунок 6 — График с персплексией в 25**

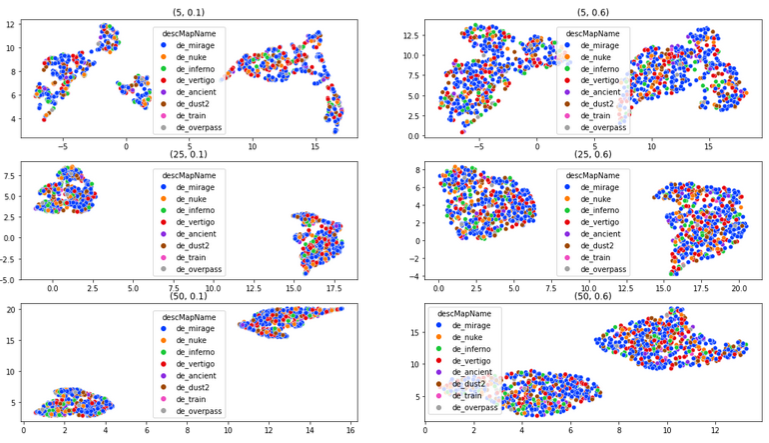


**Рисунок 7 — График с перплексией**

Попробуем доказать несостоятельность связанности данных с помощью umap.

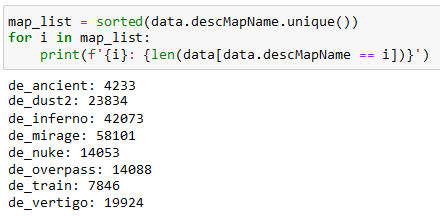


**Рисунок 8 — Код для обучения модели umap и вывода 6 графиков с различными показателями кучности и перплексии**



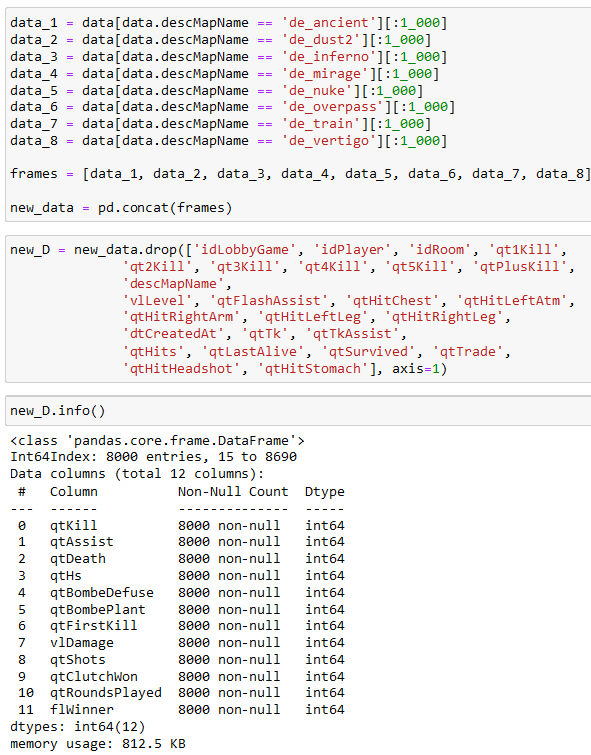
**Рисунок 9 — Графики umap**

Похоже, данные никак не коррелируют между собой. Возможно, дело в неравномерности в количество сыгранных карт.

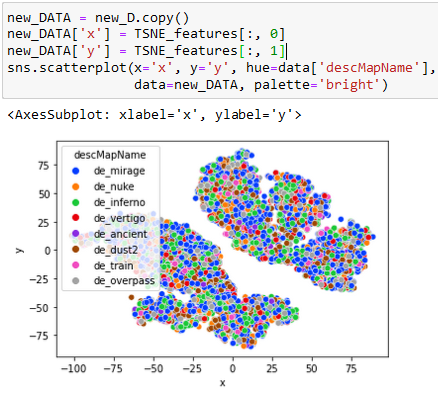


**Рисунок 10 — Количество записей по различным картам**

Как оказалось, разница между записями по картам неравномерна. Попробуем сделать случайную выборку по 1000 записей от каждой карты и построим новый график по модели.



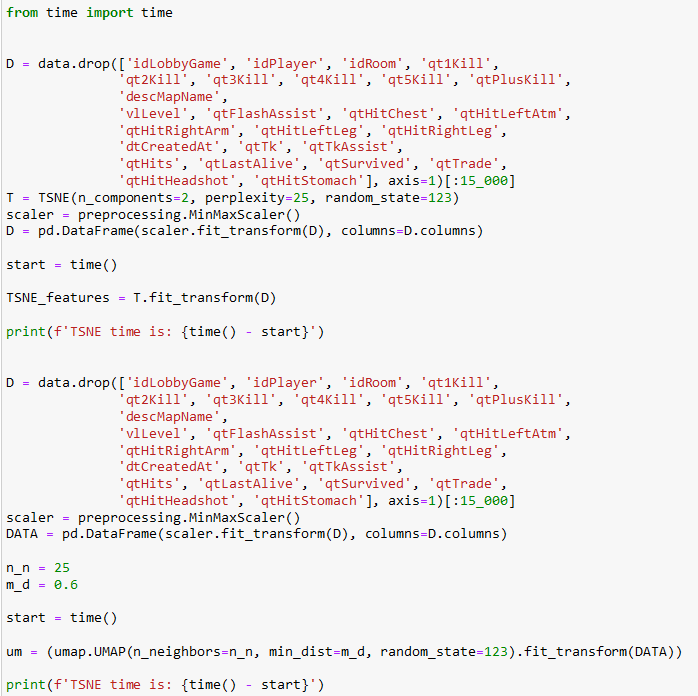
**Рисунок 11 — Новый датасет**



**Рисунок 12 — График с перплексией в 25**

По графику делаем вывод, что данные не подходят для создания модели t-SNE.

Попробуем выяснить время работы алгоритмов для выборки в 15 тысяч элементов из генеральной совокупности записей.



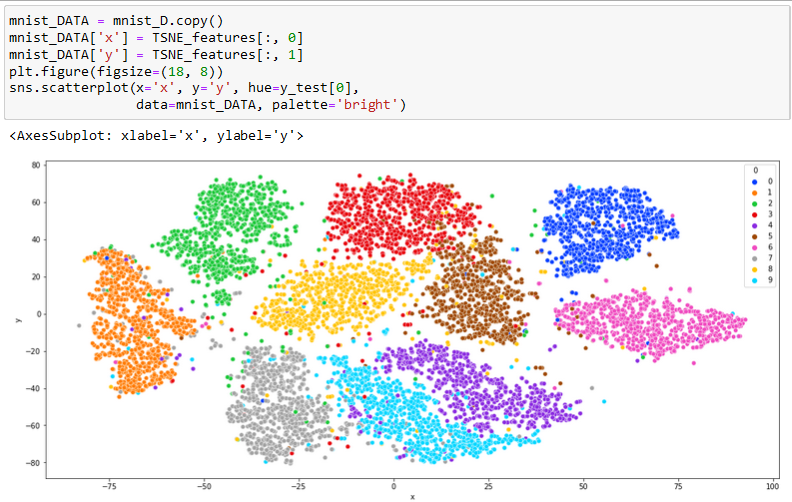
**Рисунок 13 — Код для сравнения алгоритмов**



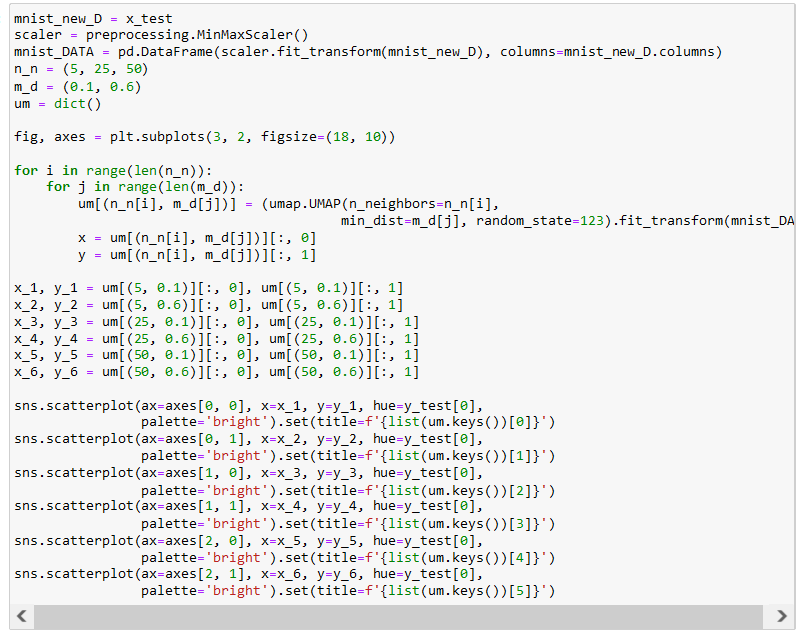
**Рисунок 14 — Сравнение времени исполнения алгоритмов umap и t-SNE**

Umap оказался более, чем в четыре раза быстрее, чем t-SNE.

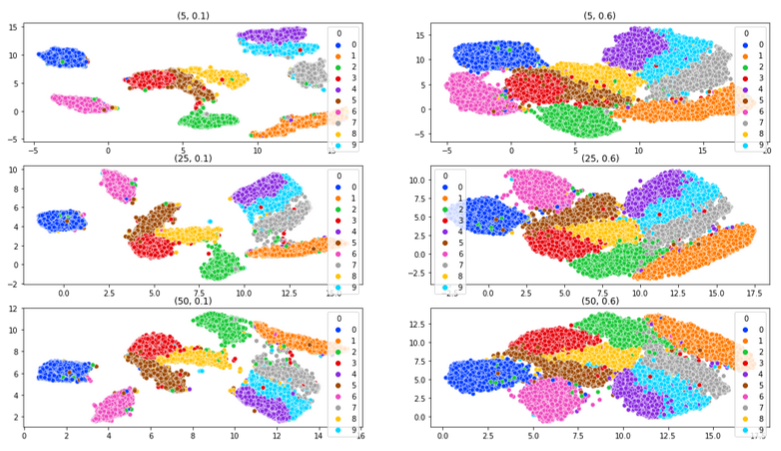
Попробуем оценить алгоритмы, используя известный набор данных mnist, содержащий рукописные цифры. В данном случае графики получились понятные, а «соседи» находятся в различных областях.



**Рисунок 16 — t-SNE график с перплексией в 25**



**Рисунок 17 — Код для отображения графиков umap**



**Рисунок 18 — графики umap**

Нас интересует на Рисунке 18 график (25, 0.6) с перплексией в 25. Этот график отличается от графика, построенного с помощью t-SNE, что дает нам сделать вывод о разном принципе работы алгоритмов.

# ВЫВОДЫ

В ходе выполнения работы были изучены библиотеки matplotlib, seaborn, sklearn. Выполнены задания, включающие в себя изучение датасета путем создания графиков через кластеризацию данных. Сделан вывод, что данные не связаны между собой и методы umap и TSNE не подходят для cs\_go датасета, но прекрасно себя чувствуют в рамках датасета mnist.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Python Documentation [Электронный ресурс] - <https://www.python.org/>
2. Jupyter Documentation [Электронный ресурс] - <https://docs.jupyter.org/en/latest/index.html>
3. Pandas Documentation [Электронный ресурс] - <https://pandas.pydata.org/docs/getting_started/index.html>
4. Matplotlib Documentation [Электронный ресурс] - <https://matplotlib.org/>
5. Plotly Documentation [Электронный ресурс] - <https://plotly.com/python/>
6. Numpy Documentation [Электронный ресурс] - <https://numpy.org/doc/>
7. Stats Documentation [Электронный ресурс] - <https://docs.scipy.org/doc/scipy/reference/stats.html>