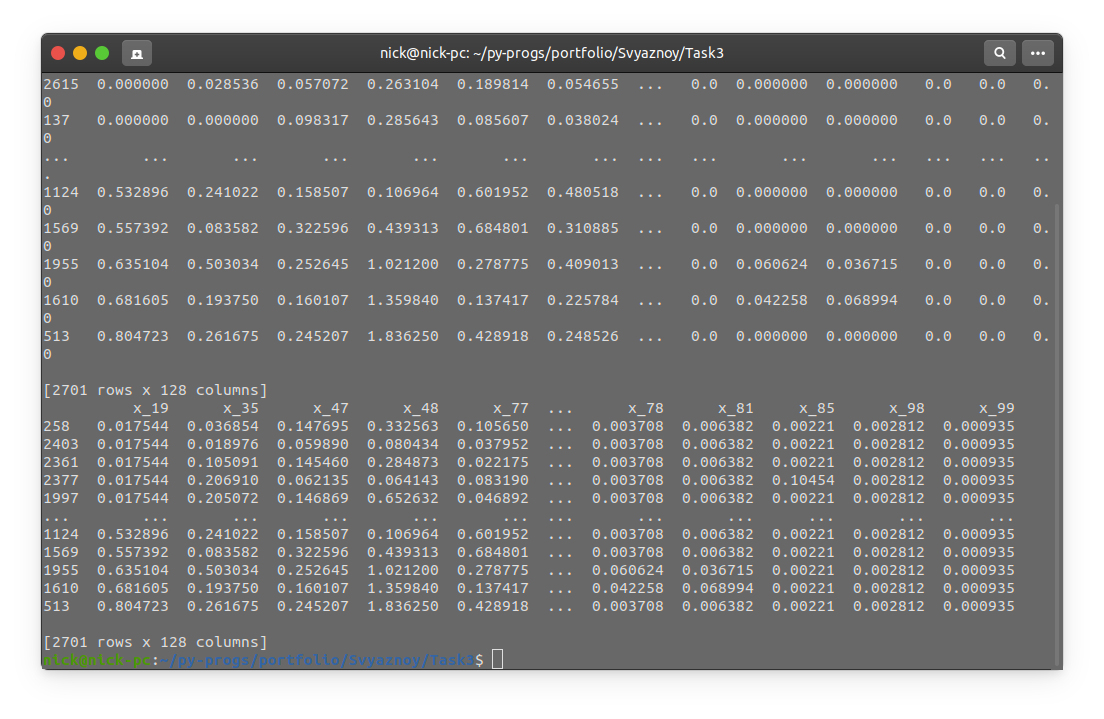
***Отчет по Task3***

**Первый этап** — подготовка данных. Преобразование данных из .txt в .csv точно такое же, как и во втором задании. Далее возпользуемся ProfileReport, чтобы получить детальную информацию по датасету. Можно видеть, что во всех столбцах есть пропуски, и во многих из них нули составляют очень большу часть от всего столбца — от 60% до 98%. Однако есть и столбцы, в которых пропущено всего 3-9% данных. Учитывая эти факты, а также то, все 128 предикторов — непрерывные величины, а не категориальные, то удаление строк или столбцов с нулями приведет к потере почти всех данных, поэтому можно попробовать следующий алгоритм: на начальном этапе оставить только те столбцы, где кол-во пропущенных данных не превышает 10%, и заполнить эти пропуски (средним, методом соседей, линейной регрессией или еще каких образом); на следующем этапе добавлять столбцы, где пропущенных данных больше и заполнять пропуски теми же методами; на последнем же этап добавить столбцы, в которых пропусков больше 90%, и вместо нулей вставить средние значение по соотвествующим столбцам. Изучив отчет по данному датасету, оказалось, что в каждом столбце есть нули. Все столбцы разбиты на 5 групп: <10%, 10-30%, 30-60%, 60-90%, >90%, по кол-ву в них нулей. **Второй этап** — заполнение пропусков. Пропуски в первой группе будем заполнять средним значением соотвествующего столбца. Таких столбцов мало, и число пропусков мало, по сравнению с другими, поэтому после заполнения эти столбы вполне сгодятся в качестве базы. В пятой группе пропуски также будем заполнять средними значениями, так как их слишком много, а заполнение их, основываясь на заполнении столбцов из други групп, может повлечь за собой накопление излишнего шума и ошибки. При работе со второй группой каждый столбец из нее будет прогнозируемой величиной, а столбцы из первой группы — предикторами. Соотвественно номера строк в столбцах из второй группы, не содержащие нулей, будут обучающими. Будем последовательно обучать модель на каждой паре «столбцы первой группы»-«столбец из второй группы». На этом этапе в качестве модели будем использовать *MLPRegressor*, которая хороша в аппроксимации нелинейных ф-ий. Поиск по сетке гиперпараметров использовать не будем, так как он только увеличит время ожидания, но даже без него точность будет куда выше, чем при заполнении средними значениями. С группами 3 и 4 будем действовать аналогичными образами, дополняя предикторы столбцами из групп, в которых не осталось нулей. Естественно, это ведет к некоторому накоплению ошибки, потому 5 группа была заполнена именно средними значениями, так как доля данных, доступных для обучения, крайне мала (в 5 группе менее 10% строк, не содержащих нулей). Как можно видеть из фото ниже, сортировки не нарушили целостности в строках, то есть на одних и тех же номерах строк значения предикторов, которые были в изначальных данных, остались на своих местах.



Проанализировав окончательные данные (ф-ия *GetInfo()*), заметил, что некоторые предикторы стали сильно коррелировать, поэтому, зная их контекст и назначение в случае реальной задач, некоторые из них можно исключить. Скажу честно, что не знаю, существует ли какая-то метрика качества в подобных задачах, но считаю, что придуманный метод весьма хорош и оправдан, особенно использование *MLP*, вместо *k*-средних