Техническое задание по курсовому проекту Слепова Артемия, Б21-221.

(обычным шрифтом дана та часть задания, которая будет «на релизе», а курсивом – то, что постараюсь сделать, ~~если не задушит 32-я кафедра~~ если будут силы).

Дана таблица с данными о крупнейших капиталистах мира, столбцы с их показателями и данными о странах, в которых они оперируют. Число столбцов фиксированное, так же, как и число строк (для большего быстродействия я решил использовать array, а не vector. *Но можно чуть поднапрячься и сделать выделение памяти под датасет динамическим*, *тем более что только так возможно будет работать с по-настоящему огромными файлами).*

Данные берутся из датасета Billionaires Statistics Dataset (год сбора данных: 2023) (<https://www.kaggle.com/datasets/nelgiriyewithana/billionaires-statistics-dataset/data>).

Из него выбрасываются столбцы personName (имя миллиардера для анализа не важно), source (дублирование данных), industries (дублирование данных), organization (название организации не вано), birthdate (день рождения не важен), lastName (дублирование данных), firstName (дублирование данных), date (дата сбора статистики не важна), state (дублирование данных), residenceStateRegion (район региона, в котором находится штаб-квартира, не важен), birthyear (дублирование данных), birthMonth (дублирование данных), birthday (дублирование данных), latitude\_country (местонахождение страны не важно), longitude\_country (местонахождение страны не важно).

Остальные признаки участвуют в модели полноправно.

В программе при запуске начинает работу функция для парсинга того, что подано в csv-файле.

Каждый из столбцов разделяется и список признаков данного сэмпла подаётся на вход в следующую функцию, которая конструирует признаки (нужно будет придумать способ, как сопоставить сферам деятельности буржуев числа, как проделать то же со странами, в которых они работают*, попробовать учесть также и город, в котором находится штаб-квартира их компаний; превратить и города в числа*). Какое-то время будет потрачено на изучение и применение one-hot кодирования.

Имена будут служить лишь для визуализации данных. Очевидно, имя как таковое (за происхождение богатств в таблице отвечает отдельный столбец) не будет влиять на предсказания модели. *Хорошо бы реализовать функцию, которая бы убирала из таблицы ненужный столбец и возвращала бы измененную его версию.*

*Для разведочного анализа данных можно написать анализ с помощью критерия Пирсона корреляции между признаками попарно.*

**Конечная цель программы – сгруппировать олигархов по близости их параметров, оценить, какие сценарии становления олигархами чаще всего реализуются**. Для этого воспользуемся картой Кохонена – нейронной сетью обучения без учителя. Несомненным плюсом данного подхода является то, что не потребуется определять заранее критерии «похожести» олигархов с помощью аппарата статистики.

Будет реализован класс «узел», там будет вектор из весов размерности признаков и вектор координат узла на карте.

Число узлов фиксированное, оптимальное число будет найдено в процессе совершенствования нейросети.

Начальные веса задаются произвольно.

Затем в цикле перебираются признаки, по ним находится лучшая единица соответствия, изменяется вес этого узла и его соседей с цель приближения к начальным данным.

Далее подсчитывается суммарная ошибка, цикл прекращается при заданной точности карты.

*В идеале можно изучить CUDA и воспользоваться ею для ускорения векторных вычислений на кластере.*

Наконец, программа записывает в файл все необходимые данные для дальнейшей визуализации с помощью Python, которая, в свою очередь, может быть выполнена уже на персональном компьютере.

В ходе нескольких запусков программы будет выявлено близкое к оптимальному число узлов для карты.

Предполагается, что для обучения будут «использованы» не все олигархи. Это нужно для проверки корректности работы кода. Таким образом, на этой части данных будем тестировать сеть, а после отладки - включим её в анализ.

Во избежание переобучения будем подбирать подходящую величину σ, отвечающую за число соседних узлов, «задетых» сдвигом узла-победителя. Подбирать будем, постоянно меняя трейновую часть датасета на тестовую и наоборот.

На выходе получаем графические отображения «близких» по признакам олигархов, градации того, какие параметры чаще всего встречаются, *гифку того, как менялось взаимное расположение узлов карты в ходе обучения.*

Для чего это может пригодиться? Выявлять выбивающися из общего ряда капиталистов, так как карта не сможет классифицировать набор данных, связанный с ними.

Кроме того, если я правильно понимаю алгоритм данного типа нейросетей, то можно будет предсказывать состояние человека по набору параметров, аналогичному тому, что есть в датасете и ставить точку, соответствующую этому человеку на карту. То есть, можно будет ~~наглядно демонстрировать школьным стартаперам и криптоинвесторам, в чём они неправы~~ сохранять деньги наивных сограждан от необдуманных кредитов на бизнес :)