Зображення, що містить текст

Автоматично згенерований опис

Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

**Лабораторна робота №6**

Прикладні задачі машинного навчання

**Тема:** Прикладна задача машинного навчання

Виконав Перевірив:

студент групи ІП-11: Нестерук А. О

Панченко С. В.

Київ 2023

Зміст

[1 Мета лабораторної роботи 6](#__RefHeading___Toc239_811161302)

[2 Завдання 7](#__RefHeading___Toc241_811161302)

[3 Виконання 8](#__RefHeading___Toc243_811161302)

[3.1 Генерування набору даних 8](#__RefHeading___Toc245_811161302)

[3.2 Навчання моделі 14](#__RefHeading___Toc247_811161302)

[4 Висновок 21](#__RefHeading___Toc249_811161302)

# Мета лабораторної роботи

Навчитись розв’язувати прикладні задачі машинного навчання.

# Завдання

1. Створити, навчити і апробувати багатошарову нейронну мережу з прямою передачею сигналу для ухвалення рішення про зарахування до Університету абітурієнтів, які здали вступні іспити з математики, англійської та української мови.

Правила прийому наступні:

1. Рейтинг абітурієнтів формується за формулою 0,4 БМ+0,3БА+0,3БУ, де БМ-бал з іспиту з математики, БА-бал з іспиту з англійської мови, БУ-бал з іспиту з української мови.
2. Мінімальний прохідний бал на вступ 160 для абітурієнтів без пільг.
3. З математики для абітурієнтів без пільг мінімальний бал іспиту не може бути менший 140 балів.
4. Абітурієнти, які мають пільги, зараховуються при мінімумі 120 балів з усіх іспитів і їх рейтинг не може бути меншим ніж 144 бали
5. Університет може прийняти на навчання 350 абітурієнтів, з них не більше 10% це абітурієнти з пільгами.
6. Статистика минулих років показує, що в середньому до Університету подають документи 1500 абітурієнтів.

# Виконання

## Генерування набору даних

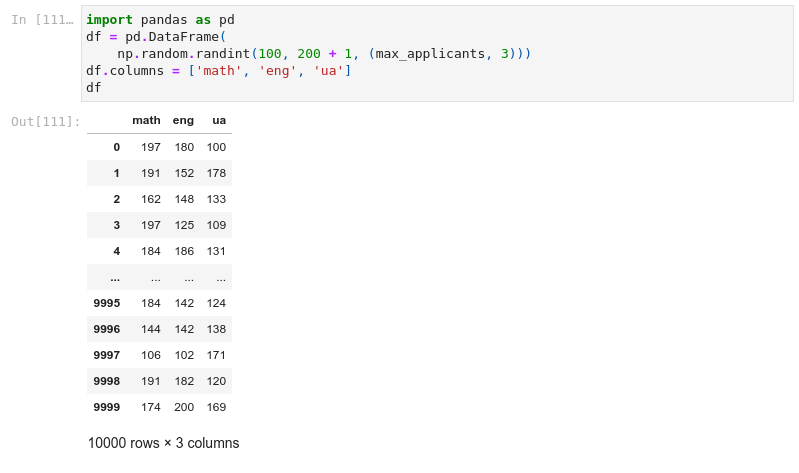
Згенеруємо набір даних результатів вступу учнів до вищих навчальних закладів.

Статистика минулих років показує, що в середньому до Університету подають документи 1500 абітурієнтів.

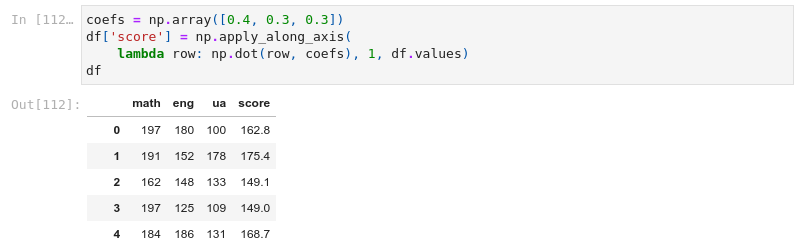
Для початку імпортуємо модуль NumPy. Кількість згенерованих значення визначенється за допомогою змінної max\_applicants.

  
  
Рисунок 3.1 - Визначення загальної кількості абітурієнтів

Згенеруємо значення в проміжку від 100 до 200 за допомогою numpy.random.random\_integers. Надамо набору даних назви стовпців.

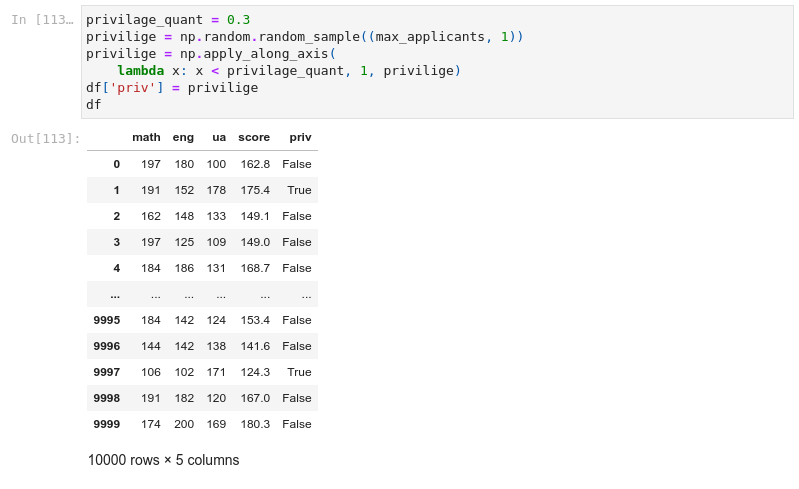
  
  
Рисунок 3.2 - Список балів абітурієнтів

Додамо поле, що визначає рейтинг. Рейтинг абітурієнтів формується за формулою 0,4БМ + 0,3БА + 0,3БУ, де БМ-бал з іспиту з математики, БА-бал з іспиту з англійської мови, БУ-бал з іспиту з української мови.

  
  
Рисунок 3.3 - Додавання поля середнього балу до балів студента

Згенеруємо стовпчик про наявність пільг в абітурієнта.

Очевидно, що студентів з пільгами не може бути багато, то згенеруємо випадкові значення у проміжку від 0 до 1. Нехай значення, що менші privilage\_quant, визначають, чи абітурієнти мають пільги.

  
  
Рисунок 3.4 - Додавання поля пільг до балів студента

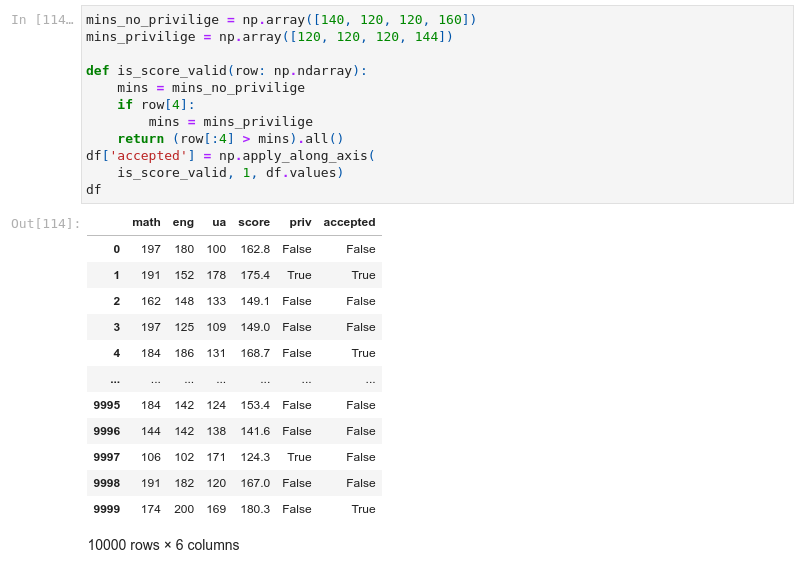
Утворимо список значень результатів прийняття абітурієнта до освітнього закладу.

Мінімальний прохідний бал на вступ 160 для абітурієнтів без пільг.

З математики для абітурієнтів без пільг мінімальний бал іспиту не може бути менший 140 балів.

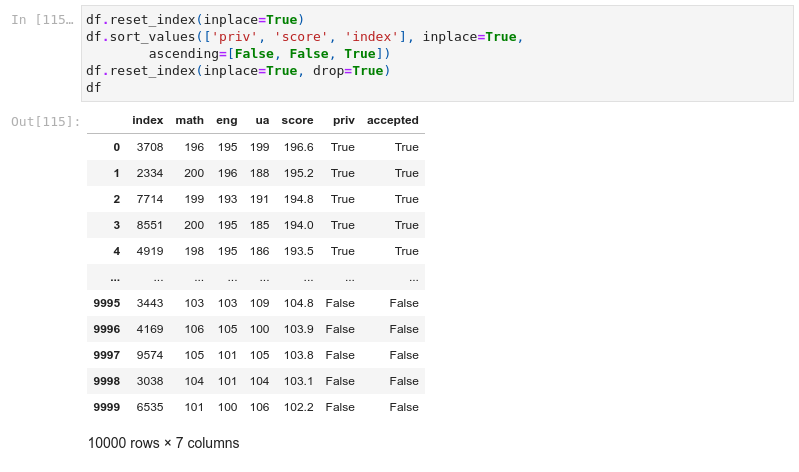
Абітурієнти, які мають пільги, зараховуються при мінімумі 120 балів з усіх іспитів і їх рейтинг не може бути меншим ніж 144 бали.

Університет може прийняти на навчання 350 абітурієнтів, з них не більше 10% це абітурієнти з пільгами.

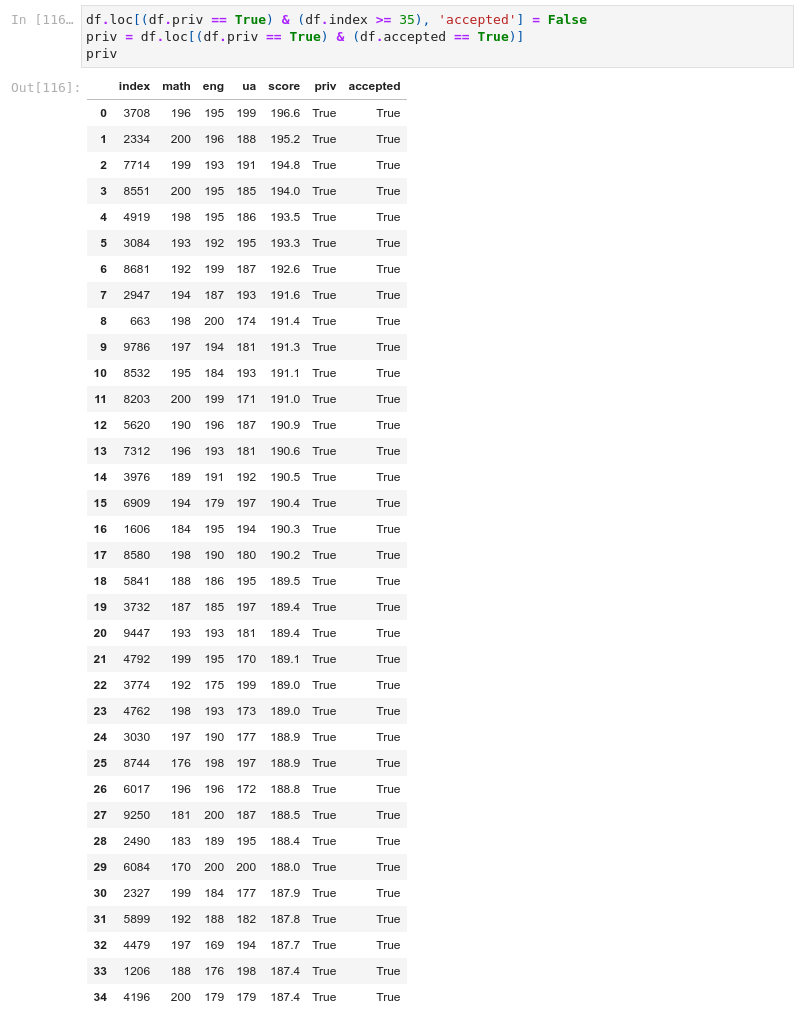
  
  
Рисунок 3.5 - Визначення прийняття за кількістю балів

Оскільки порядок подачі документів абітурієнтами має значення, то додамо стовпець index, де чим він менший, тим раніше абітурієнт подав документи. Зробимо це за допомогою методу reset\_index.

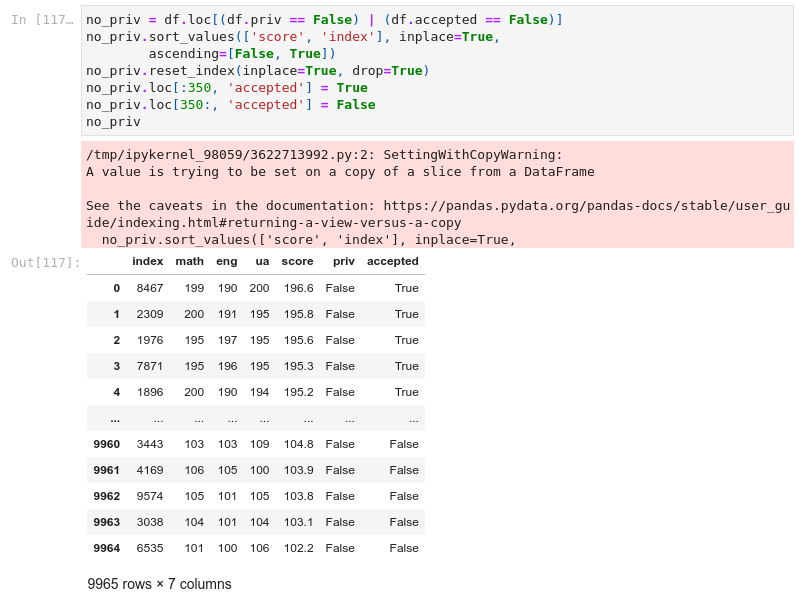
Утворимо відсортований набір даних за стовпцями в такому порядку used\_priv, score. Та визначимо ліміт прийнятих абітурієнтів. Для яких місць не залишиться, тим виставиться False у полі accepted.

  
  
Рисунок 3.6 - Індексація та сортування абітурієнтів

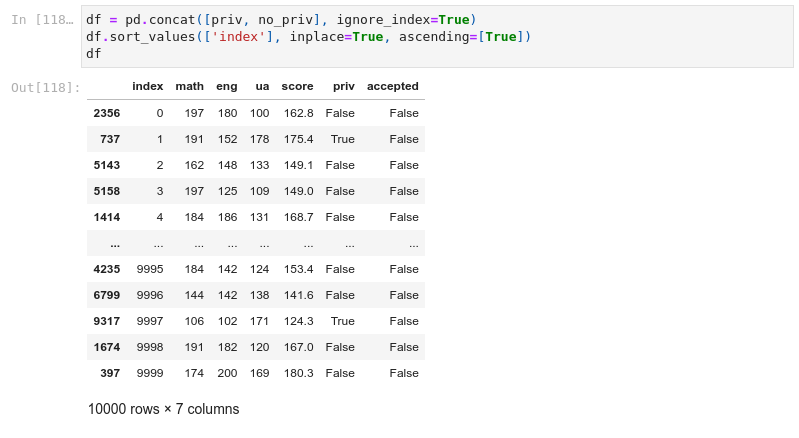
Визначення студентів з пільгами, що пройшли відбір. Виставлення студентам з пільгами, які не увійшли до перших 35-ти людей, параметру accepted = False.

  
  
Рисунок 3.7 - Перші 35 студентів з пільгами, що пройшли конкурс

Визначення результатів студентів без пільг та з пільгами, що не пройшли конкурс.

  
  
Рисунок 3.8 - Результати вступу

Відсортуємо увесь набір даних за стовпцем index, щоб повернути його до того порядку, з якого починали.

  
  
Рисунок 3.9 - Сортування за стовпцем index

Очистимо оперативну пам'ять

  
  
Рисунок 3.10 - Очищення пам'яті

## Навчання моделі

Навчання моделі будемо проводити в циклі за допомогою перебору параметрів шарів.

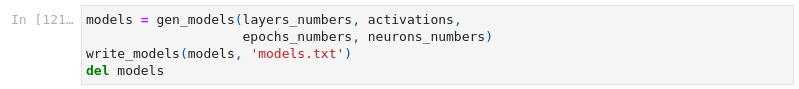
Модель завжди буде компілюватися з optimizer = 'adam', який працює на основі градієнтного спуску, та функцією втрат loss = 'binary\_crossentropy', оскільки ці параметри гарно працюють для бінарної класифікації.

Оскільки навчання моделі забирає багато часу та ресурсів, то маємо такі обмеження: максимальна кількість шарів знаходиться у layers\_numbers; функціями активації можуть бути лише ті, що наведені в activations; кількість епох знаходиться у epochs\_numbers; кількість вхідних нейронів у прихованому шарі знаходиться у neurons\_numbers.

Результати моделей будемо зберігати у список models, що зберігатиме параметри, точності на тренувальних та тестових даних. Запишемо отриманий список у файл.

  
  
Рисунок 3.11 - Функція утворення списку можливих комбінацій параметрів

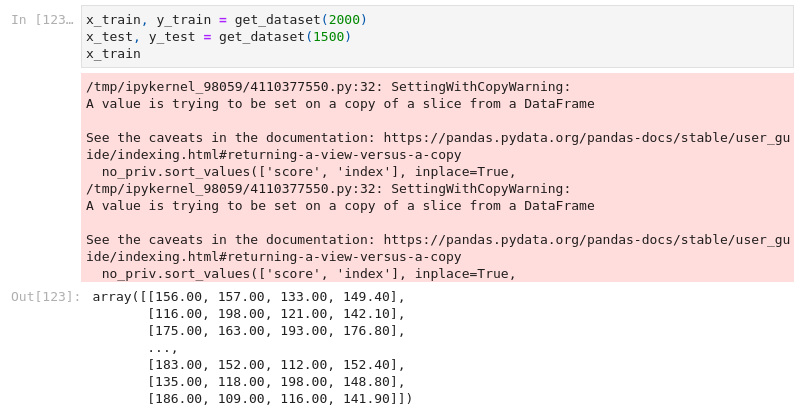
Утворимо список.

  
  
Рисунок 3.12 - Утворення списку можливих комбінацій параметрів

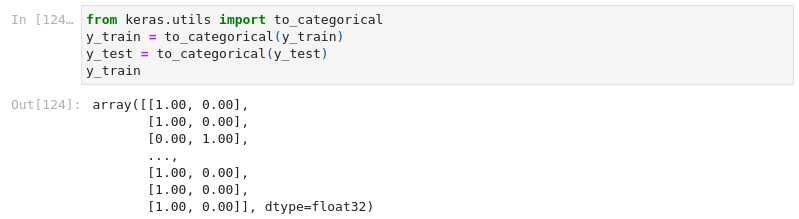
Розібравши, як створювати набір даних, запишемо генерування у функцію та утворимо з неї два набори даних тренувальний та тестовий.

  
  
Рисунок 3.13 - Функція створення набору даних

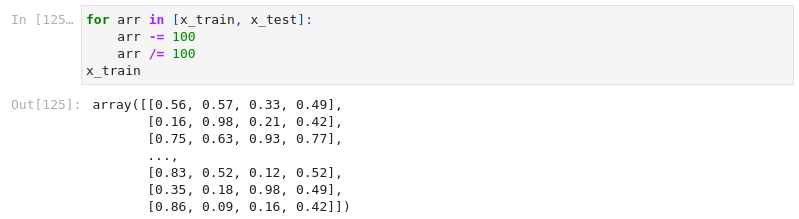
Утворення тренувального та тестового наборів даних.

  
  
Рисунок 3.14 - Навчальні та тестові набори даних

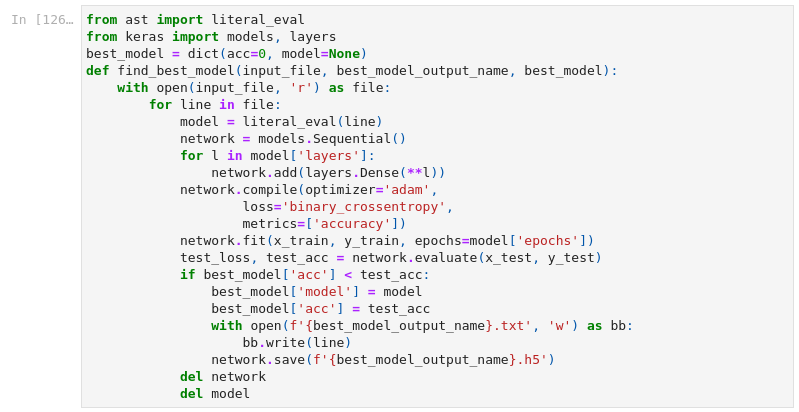
Підготуємо мітки за допомогою to\_categorical. Тобто 0 перетвориться на [1, 0], а 1 - [0, 1].

  
  
Рисунок 3.15 - Підготування міток

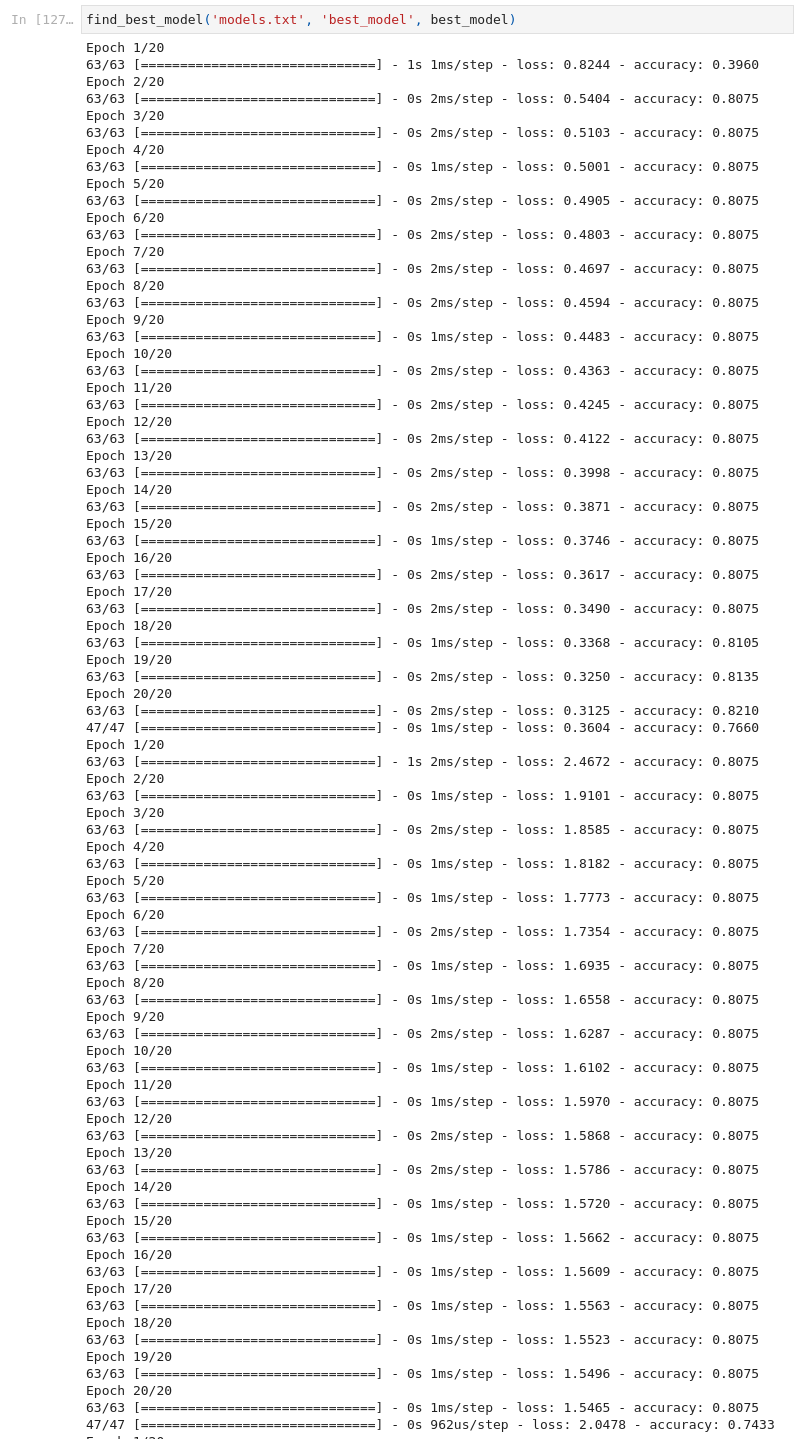
Нормалізуємо набори даних аргументів. Для початку віднімемо 100 та поділимо на 100.

  
  
Рисунок 3.16 - Нормалізація даних

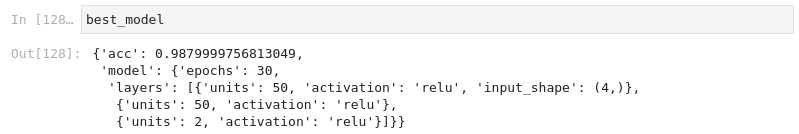
Запишемо функцію знаходження найкращої моделі.

  
  
Рисунок 3.17 - Функція знаходження найкращої моделі

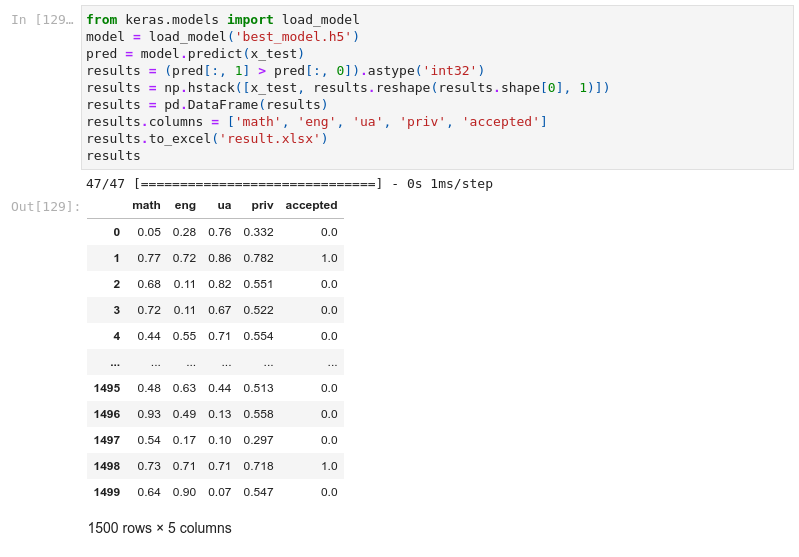
Визначимо найкращу модель. Імпортуємо модуль ast для перетворення текстового рядка.

  
  
Рисунок 3.18 - Визначення найкращої моделі

Виведемо найкращу модель.

  
  
Рисунок 3.19 - Найкраща модель

Перенесемо прогнози в excel.

  
  
Рисунок 3.20 - Результати моделі

# Висновок

З огляду на умови задачі та виконання лабораторної роботи можна зробити наступні висновки. Була успішно створена, навчена та апробована багатошарова нейронна мережа з прямою передачею сигналу для ухвалення рішення про зарахування до університету абітурієнтів, які здали вступні іспити з математики, англійської та української мови. Для цього були складені функції генерування тренувальних наборів даних та виведені параметри нейронної мережі шляхом перебору параметрів кількості нейронів в шарах, функцій активації, кількості шарів, кількості поколінь.

Отримані результати були виведені в Excel-таблицю, що дозволить детально проаналізувати їх та виконати подальші маніпуляції з даними. Застосування нейронної мережі може бути корисним у відборі кандидатів на навчання в університеті, оскільки вона дозволяє провести аналіз даних і прийняти рішення на основі об'єктивних показників.

Отже, можна стверджувати, що лабораторна робота була успішно виконана та дозволила отримати корисні результати, які можуть бути використані в подальших дослідженнях та практичних застосуваннях.