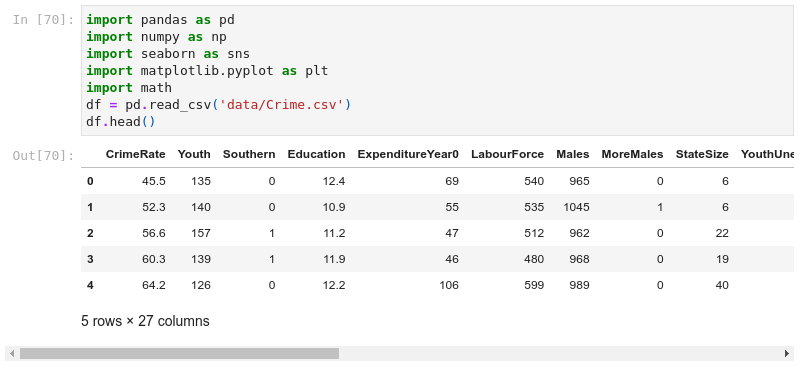
# Виконання

## Перетворення даних

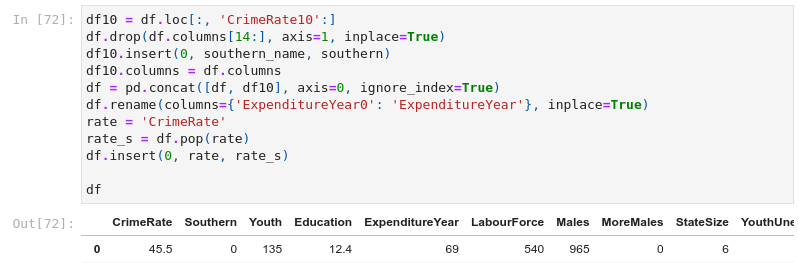
Для початку імпортуємо модулі pandas, numpy, seaborn, matplotlib. Завантажимо датафрейм.

  
  
Рисунок 3.1 - Завантаження датафрейму

Переставимо колонку у "Southern" на перше місце, оскільки розташування штату не змінювався.

  
  
Рисунок 3.2 - Переставлення колонки

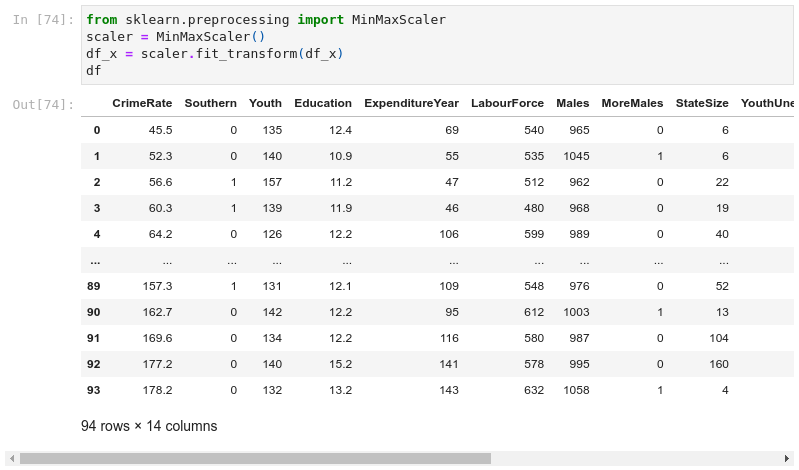
Переставимо показники через 10 років у звичайні колонки, розділивши датафрейм надвоє та з'єднавши частини вертикально. Таким чином ми збільшуємо розмір вибірки у два рази, тому модель матиме більше даних, аніж у тому випадку якби ми знаходили, наприклад, середнє арифметичне між колонками за 10 років.

  
  
Рисунок 3.3 - Перетворення даних

Розділимо дані на аргументи та значення.

  
  
Рисунок 3.4 - Розділення даних на аргументи та значення

Зробимо масштабування даних за допомогою MinMaxScaler.

  
  
Рисунок 3.5 - Масштабування даних

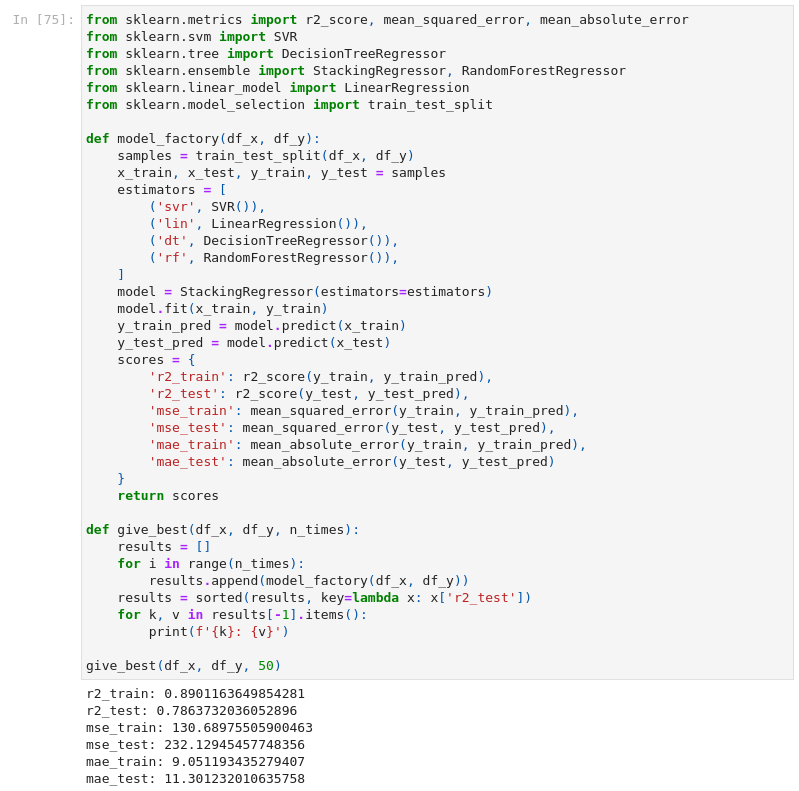
## Навчання та тестування моделі

Зараз побудуємо модель, де будемо використовувати одночасно декілька алгоритмів для того, щоб мати кращий результат. До прикладу, оберемо SVR, LinearRegression, DecisionTree та RandomForest. Для цього імпортуємо з пакету sklearn.ensemble StackingClassifier та RandomForestRegressor, з sklearn.svr - SVR, з sklearn.linear\_model - LinearRegression, з sklearn.tree - DecisionTreeRegressor.

Напишемо функцію model\_factory та проведемо тренування n-кількість разів, зберігатимемо найкращу з моделей. Також будемо всередині використовувати функцію train\_test\_split, яка ділить датасет на 75% тренувальних та 25% тестових даних за замовчуванням.

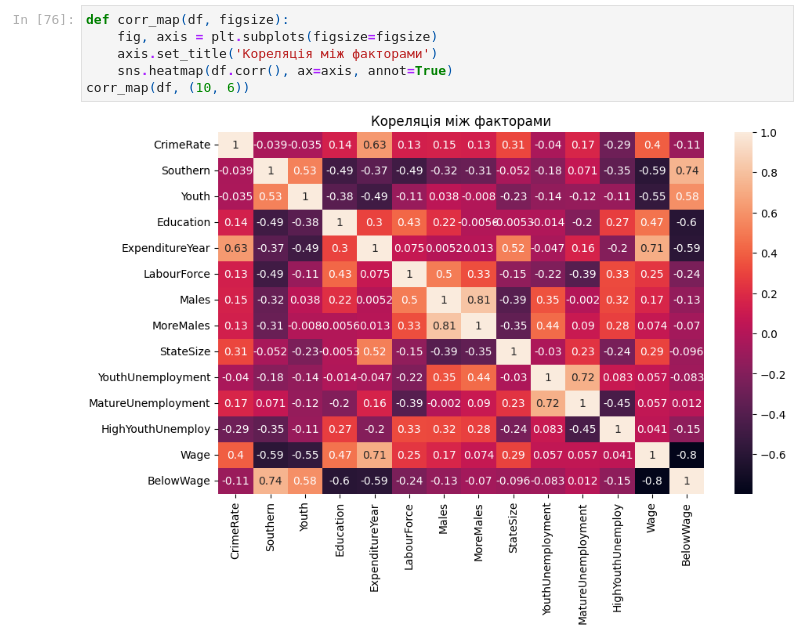
Оцінювання буде проводитися параметр R2, або коефіцієнт детермінації, який кількісно визначає частку дисперсії залежної змінної, яку можна передбачити на основі незалежних змінних. Також використаємо середню квадратичну похибку та середню абсолютну.

Проведемо певну кількість тренувань моделі та оберемо найкращу.

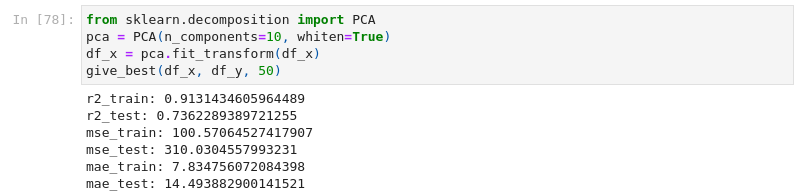
  
  
Рисунок 3.6 - Тренування моделей

## Зменшення вимірів (Dimensionality reduction)

Однак дана модель має один мінус. У ній забагато даних обробляється. І, можливо, результати можна покращити, зробивши dimesionality reduction. Для початку побудуємо матрицю кореляцій.

  
  
Рисунок 3.7 - Матриця кореляцій

Як бачимо, матриця має доволі багато взаємозалежних величин, тому зменшимо їхню кількість. Для цього використаємо PCA - Principal Component Analysis, або метод головних компонент. В Імпортуємо відповідний пакет PCA з sklearn.decomposition.

  
  
Рисунок 3.8 - Зменшення залежних величин

Як бачимо, точність для тестових даних погіршилася, а тому сенсу зменшувати далі нема, оскільки величини сильно взаємозалежні.