Report for Salary Dataset Simple Linear Regression

1. Dataset Introduction

Salary Dataset in CSV for Simple linear regression.

Columns:

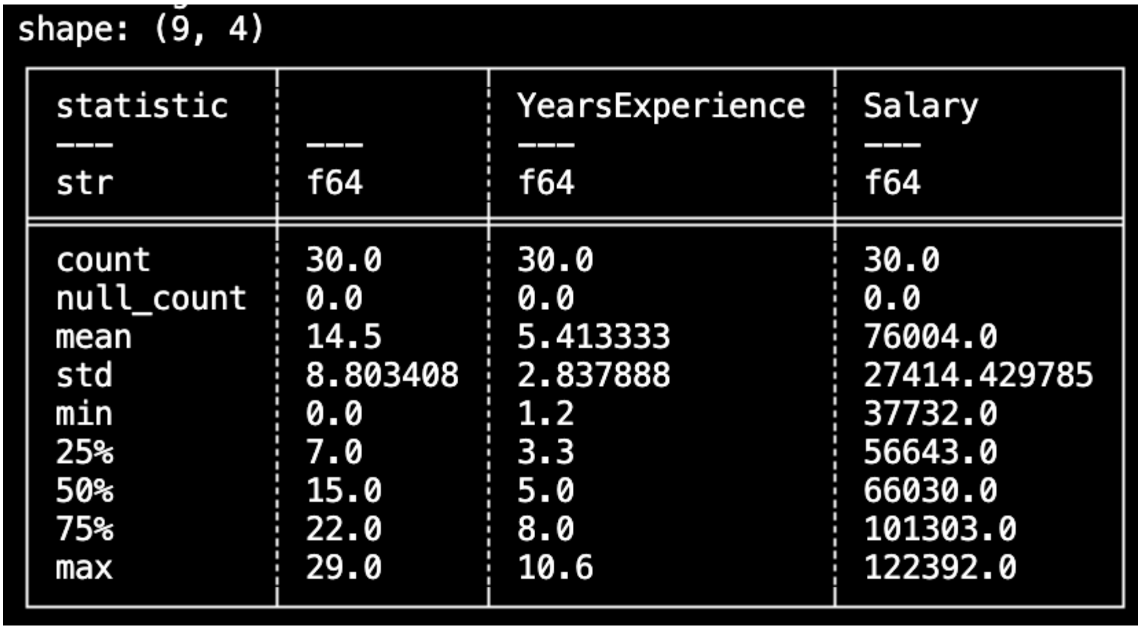
#

YearsExperience

Salary

Data source: *https://www.kaggle.com/datasets/abhishek14398/salary-dataset-simple-linear-regression/data*

2. Descriptive Statistics



```
shape: (9, 4)
┌────────────┬──────────┬─────────────────┬──────────────┐
│ statistic  ┆          ┆ YearsExperience ┆ Salary       │
│ ---        ┆ ---      ┆ ---             ┆ ---          │
│ str        ┆ f64      ┆ f64             ┆ f64          │
╞════════════╪══════════╪═════════════════╪══════════════╡
│ count      ┆ 30.0     ┆ 30.0            ┆ 30.0         │
│ null_count ┆ 0.0      ┆ 0.0             ┆ 0.0          │
│ mean       ┆ 14.5     ┆ 5.413333        ┆ 76004.0      │
│ std        ┆ 8.803408 ┆ 2.837888        ┆ 27414.429785 │
│ min        ┆ 0.0      ┆ 1.2             ┆ 37732.0      │
│ 25%        ┆ 7.0      ┆ 3.3             ┆ 56643.0      │
│ 50%        ┆ 15.0     ┆ 5.0             ┆ 66030.0      │
│ 75%        ┆ 22.0     ┆ 8.0             ┆ 101303.0     │
│ max        ┆ 29.0     ┆ 10.6            ┆ 122392.0     │
└────────────┴──────────┴─────────────────┴──────────────┘
```

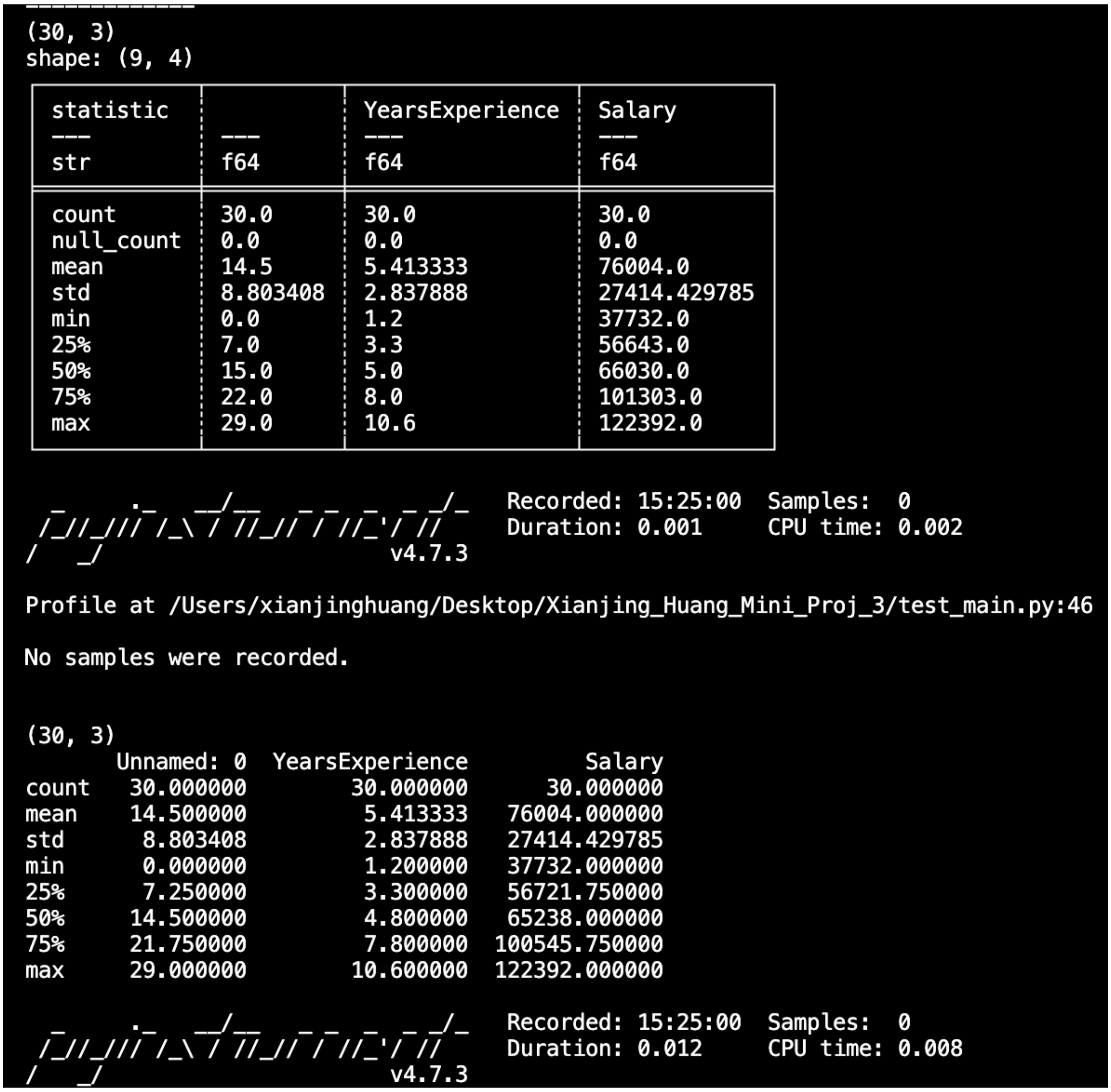
3. Profiler benchmark for Polars vs Pandas

I use Profiler from pyinstrument to compare Polars & Pandas.

For my dataset, Polars took 0.001s while Pandas took 0.012s.

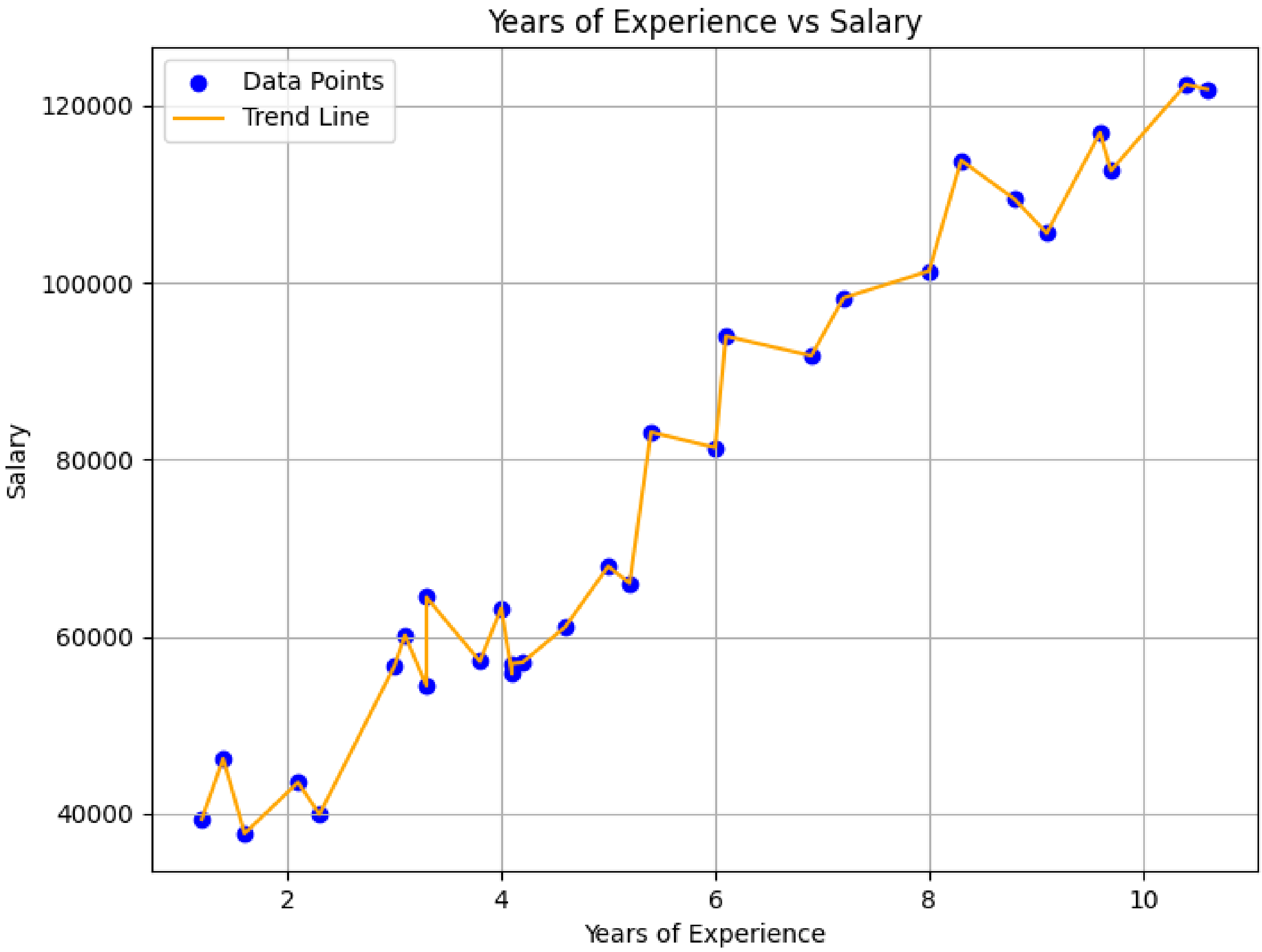
Polars generally outperforms Pandas in terms of speed, especially for large datasets. This is because Polars is designed for parallelized execution and is optimized for in-memory performance.

Pandas may still be more familiar or convenient for certain smaller tasks or when using legacy systems that require it.



```
-------------
(30, 3)
shape: (9, 4)
┌────────────┬──────────┬─────────────────┬──────────────┐
│ statistic  ┆          ┆ YearsExperience ┆ Salary       │
│ ---        ┆ ---      ┆ ---             ┆ ---          │
│ str        ┆ f64      ┆ f64             ┆ f64          │
╞════════════╪══════════╪═════════════════╪══════════════╡
│ count      ┆ 30.0     ┆ 30.0            ┆ 30.0         │
│ null_count ┆ 0.0      ┆ 0.0             ┆ 0.0          │
│ mean       ┆ 14.5     ┆ 5.413333        ┆ 76004.0      │
│ std        ┆ 8.803408 ┆ 2.837888        ┆ 27414.429785 │
│ min        ┆ 0.0      ┆ 1.2             ┆ 37732.0      │
│ 25%        ┆ 7.0      ┆ 3.3             ┆ 56643.0      │
│ 50%        ┆ 15.0     ┆ 5.0             ┆ 66030.0      │
│ 75%        ┆ 22.0     ┆ 8.0             ┆ 101303.0     │
│ max        ┆ 29.0     ┆ 10.6            ┆ 122392.0     │
└────────────┴──────────┴─────────────────┴──────────────┘

  _     ._   __/__   _ _  _  _ _/_   Recorded: 15:25:00  Samples:  0
 /_//_/// /_\ / //_// / //_'/ //     Duration: 0.001     CPU time: 0.002
/   _/                      v4.7.3

Profile at /Users/xianjinghuang/Desktop/Xianjing_Huang_Mini_Proj_3/test_main.py:46

No samples were recorded.


(30, 3)
       Unnamed: 0  YearsExperience         Salary
count   30.000000        30.000000      30.000000
mean    14.500000         5.413333   76004.000000
std      8.803408         2.837888   27414.429785
min      0.000000         1.200000   37732.000000
25%      7.250000         3.300000   56721.750000
50%     14.500000         4.800000   65238.000000
75%     21.750000         7.800000  100545.750000
max     29.000000        10.600000  122392.000000

  _     ._   __/__   _ _  _  _ _/_   Recorded: 15:25:00  Samples:  0
 /_//_/// /_\ / //_// / //_'/ //     Duration: 0.012     CPU time: 0.008
/   _/                      v4.7.3
```

4. Data Visualization


Years of Experience vs Salary
● Data Points
Trend Line
Salary
Years of Experience