|  |
| --- |
| **به نام خدا** |
| **دانشگاه تهران دانشکده مهندسی برق و کامپیوتر هوش مصنوعی گزارش تمرین کامپیوتری صفر** |

|  |
| --- |
| **نام و نام خانوادگی: فاطمه شاه حسینی** |
| **شماره دانشجویی: 810199440** |

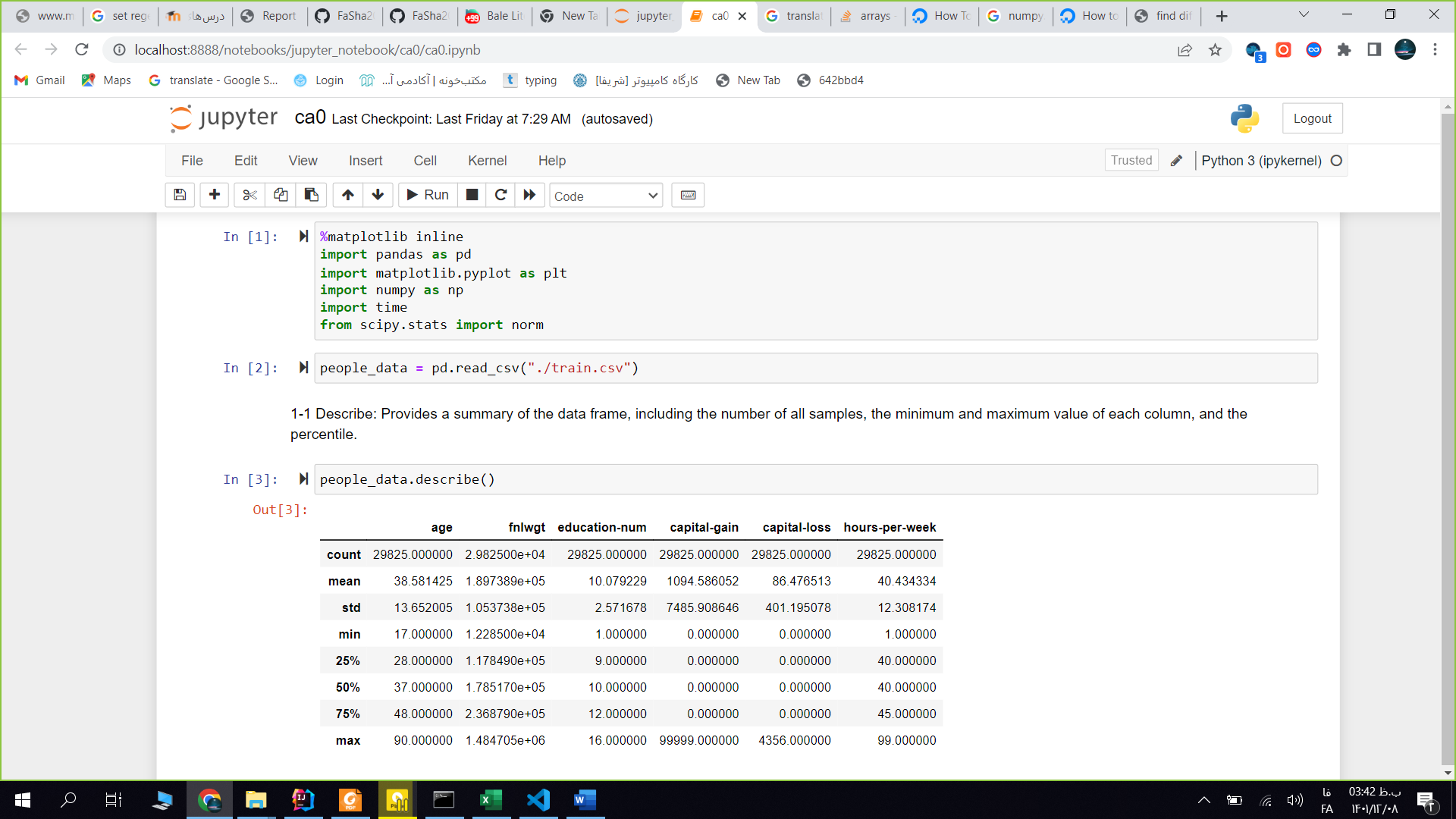
هدف پروژه : پیش بینی نواقص یک دیتا با کمک تحلیل آماری

صورت پروژه : یک فایل داده train داریم که شامل اطلاعات مربوط به افراد متقاضی کار است و با توجه به شرایط سنی، تحصیلاتی، جنسیتی، نژادی، مالی، خانوادگی و سابقه کاری میزان دستمزد این افراد را مشخص کرده است.

در فایلی دیگر به نام test همین اطلاعات متعلق به افراد دیگری وجود دارد که ما باید با کمک تحلیل اماری فایل train و پیدا کردن ویژگی هایی که بیشترین تاثیر در میزان دستمزد را دارند، دستمزد این افراد را پیش بینی کنیم.

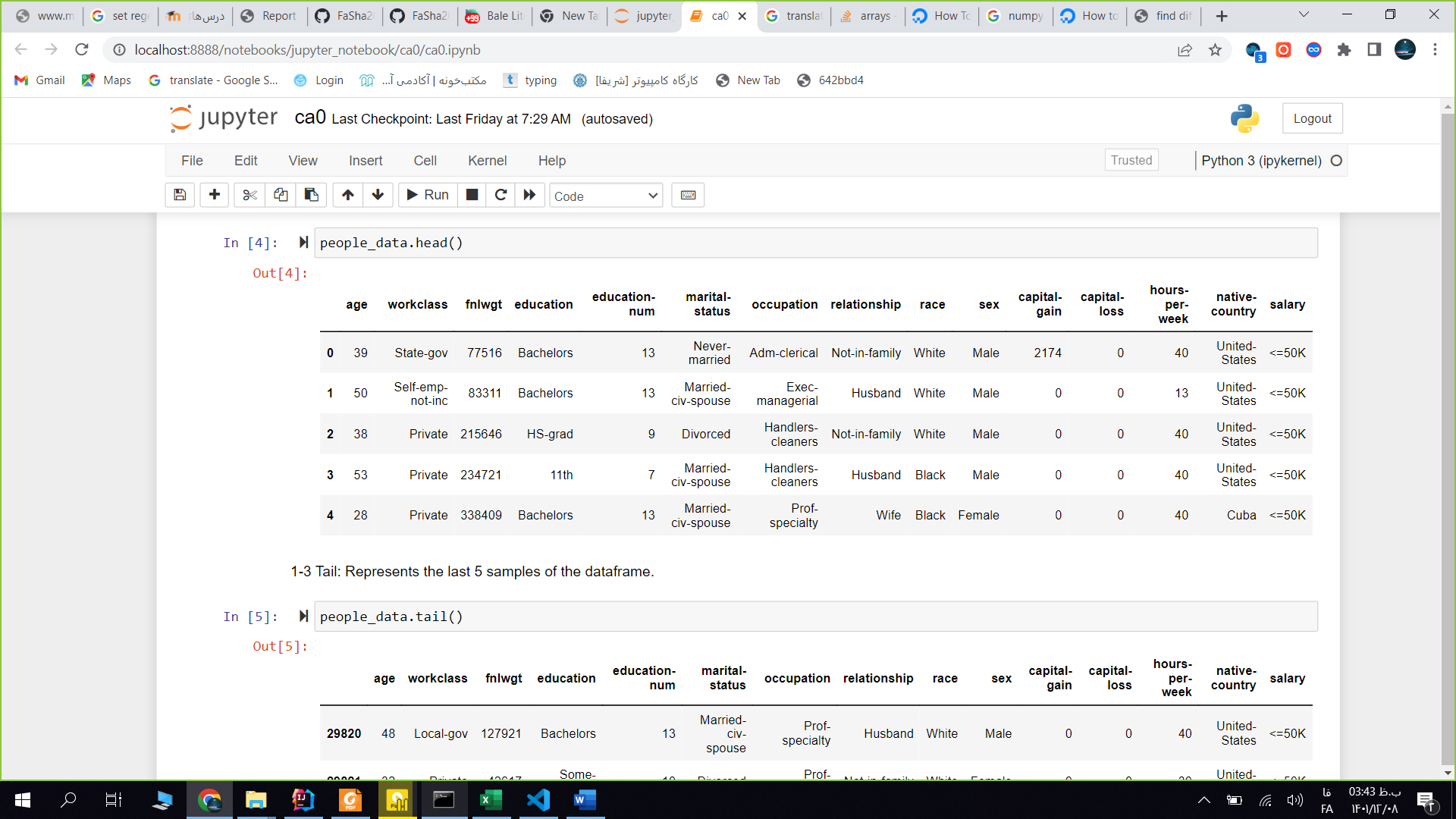
1-

خواندن از روی فایل csv و ایجاد یک دیتافریم به هدف انجام تحلیل اماری و پیدا کردن بهترین و موثرترین ویژگی در میزان دستمزد.

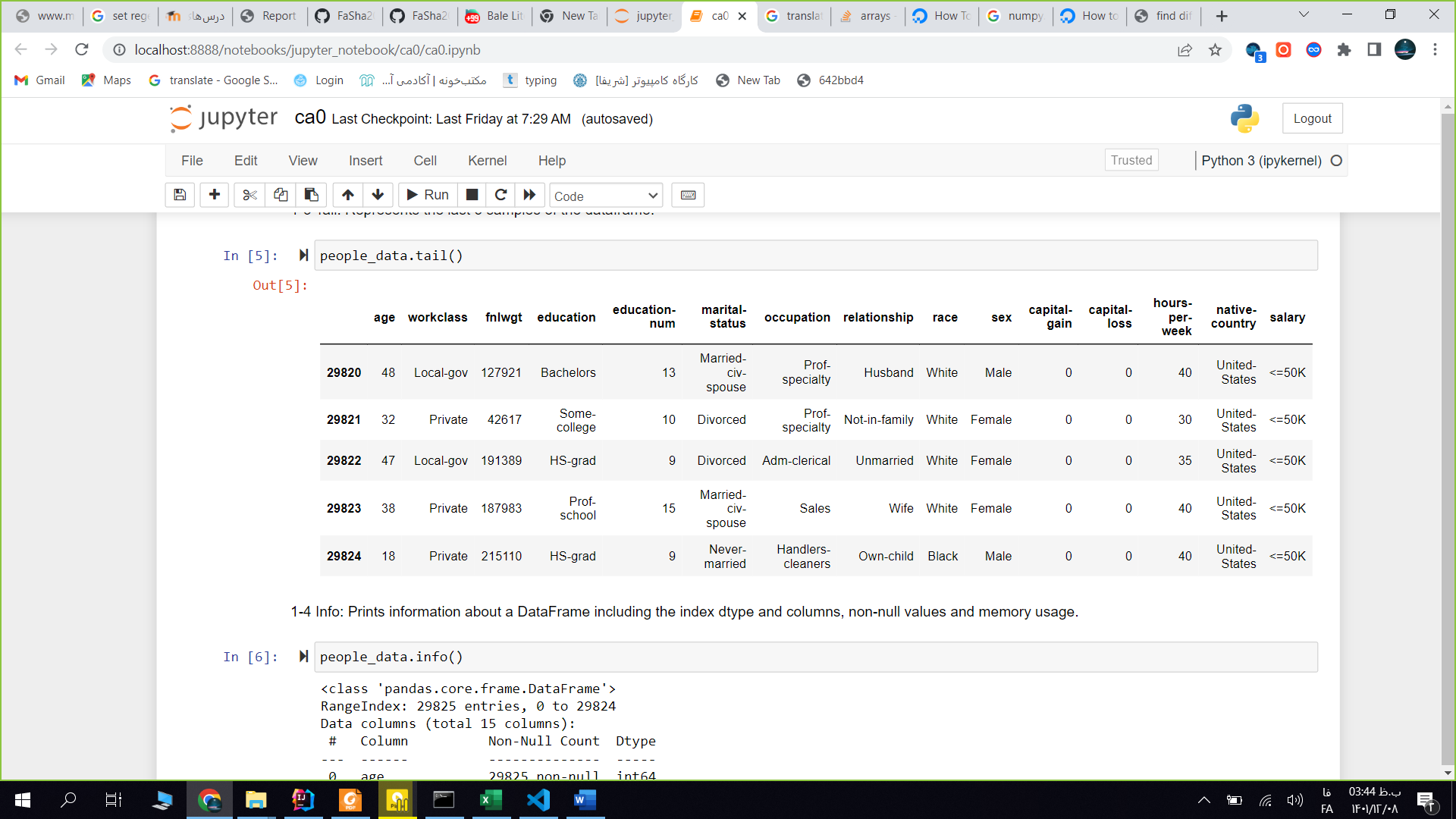


بررسی متد های:

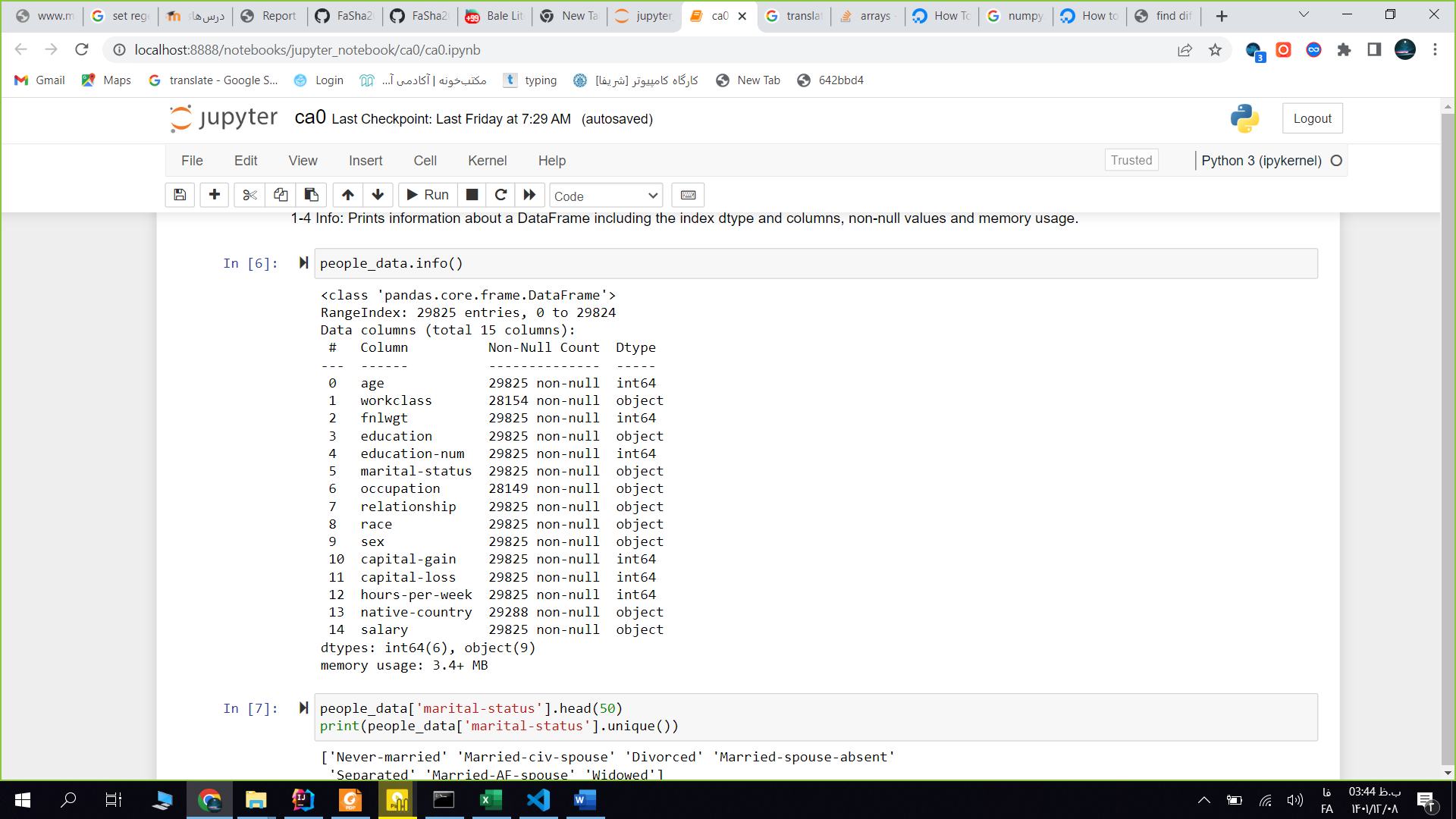
head(), : نمایش 5 عنصر اول دیتا فریم



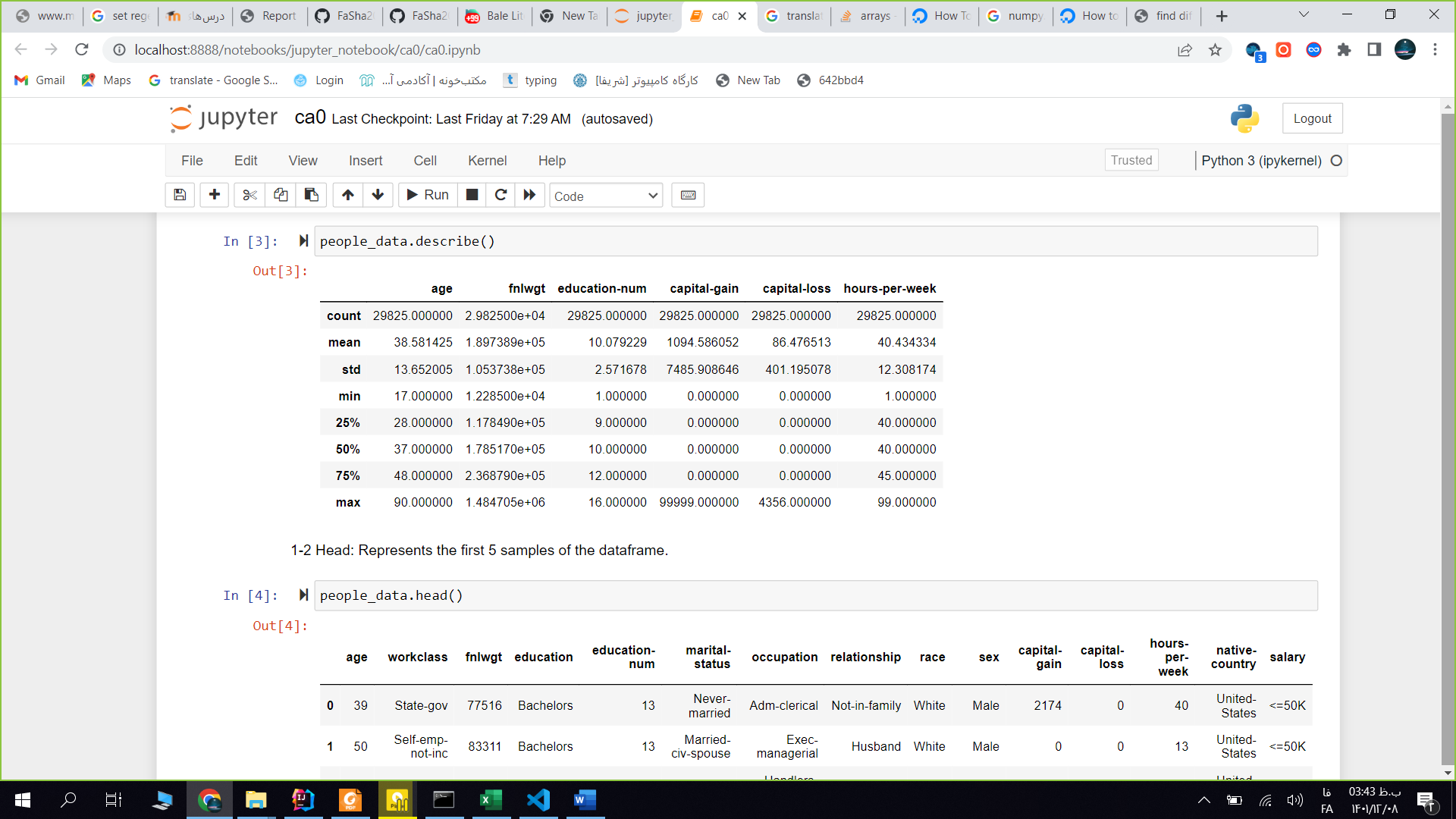
tail(): نمایش 5 عنصر اخر دیتا فریم



info(),: نمایش تایپ هر ستون و تعداد عناصر غیر تهی هر ستون به همراه حافظه مصرفی



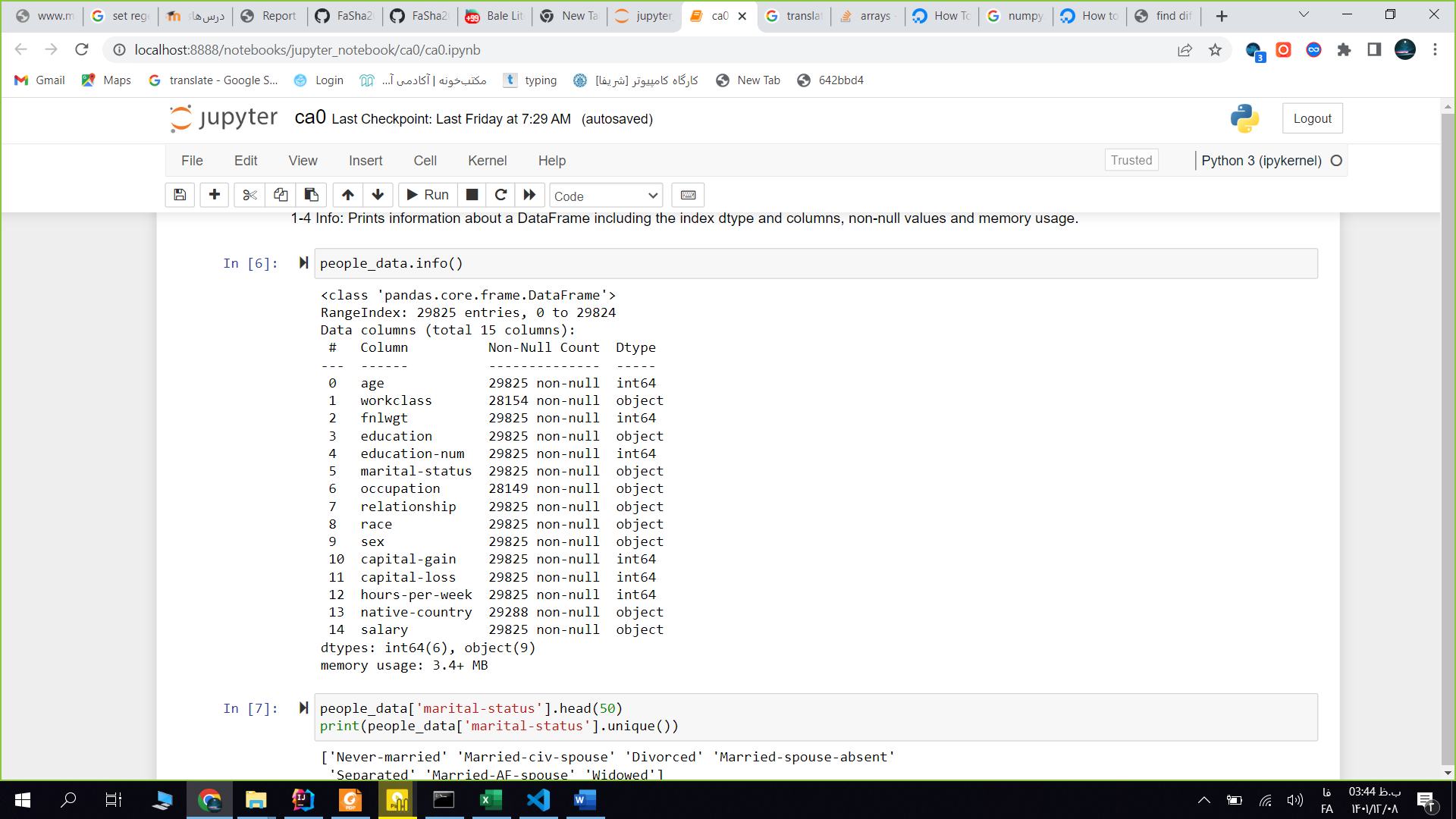
describe(): نمایش اطلاعات کلی درباره دیتا فریم شامل صدک های اول و دوم و سوم و چهارم، تعداد کل نمونه ها، کمترین و بیشترین مقدار، میانگین و انحراف معیار و ...



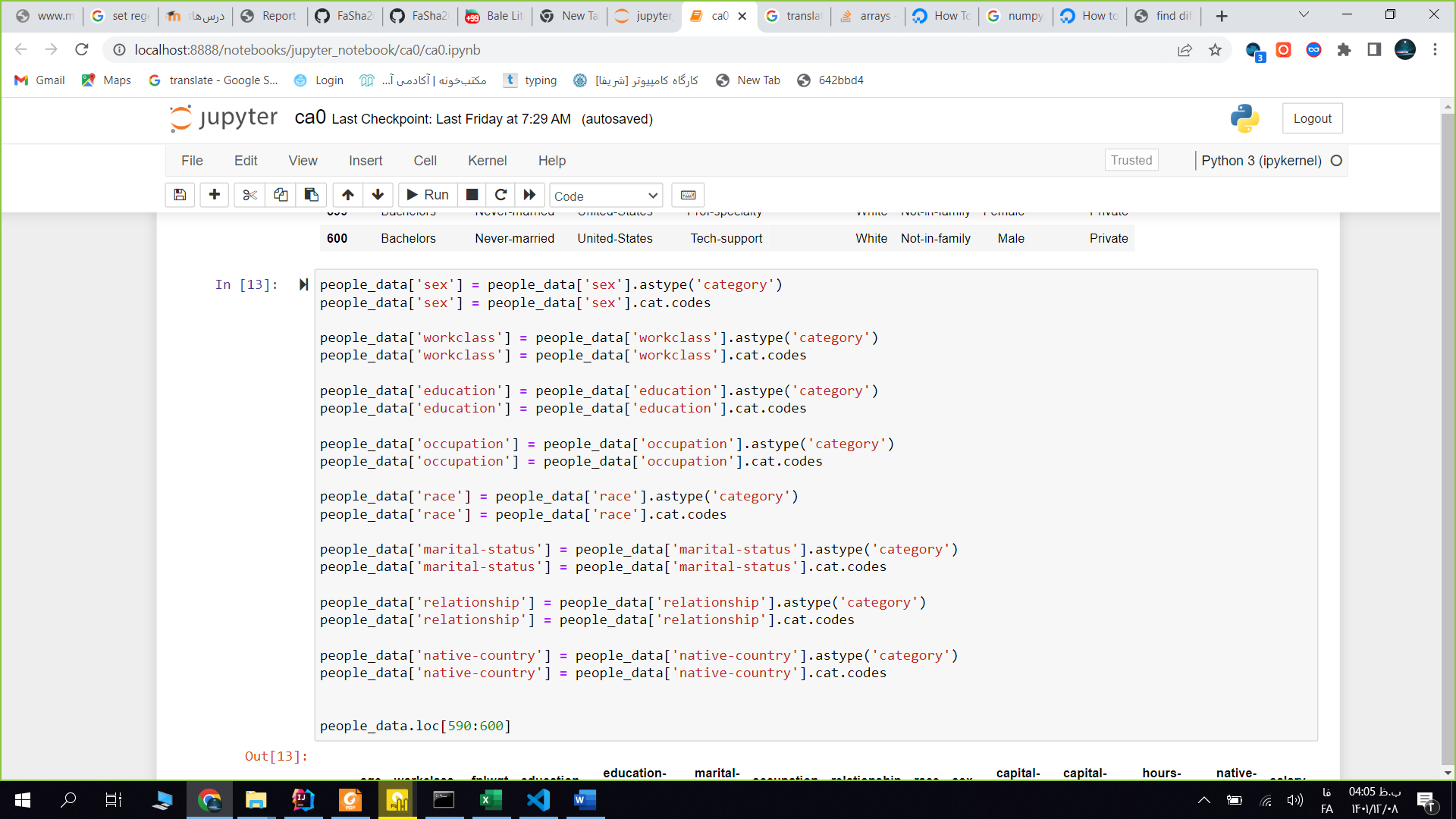
2-

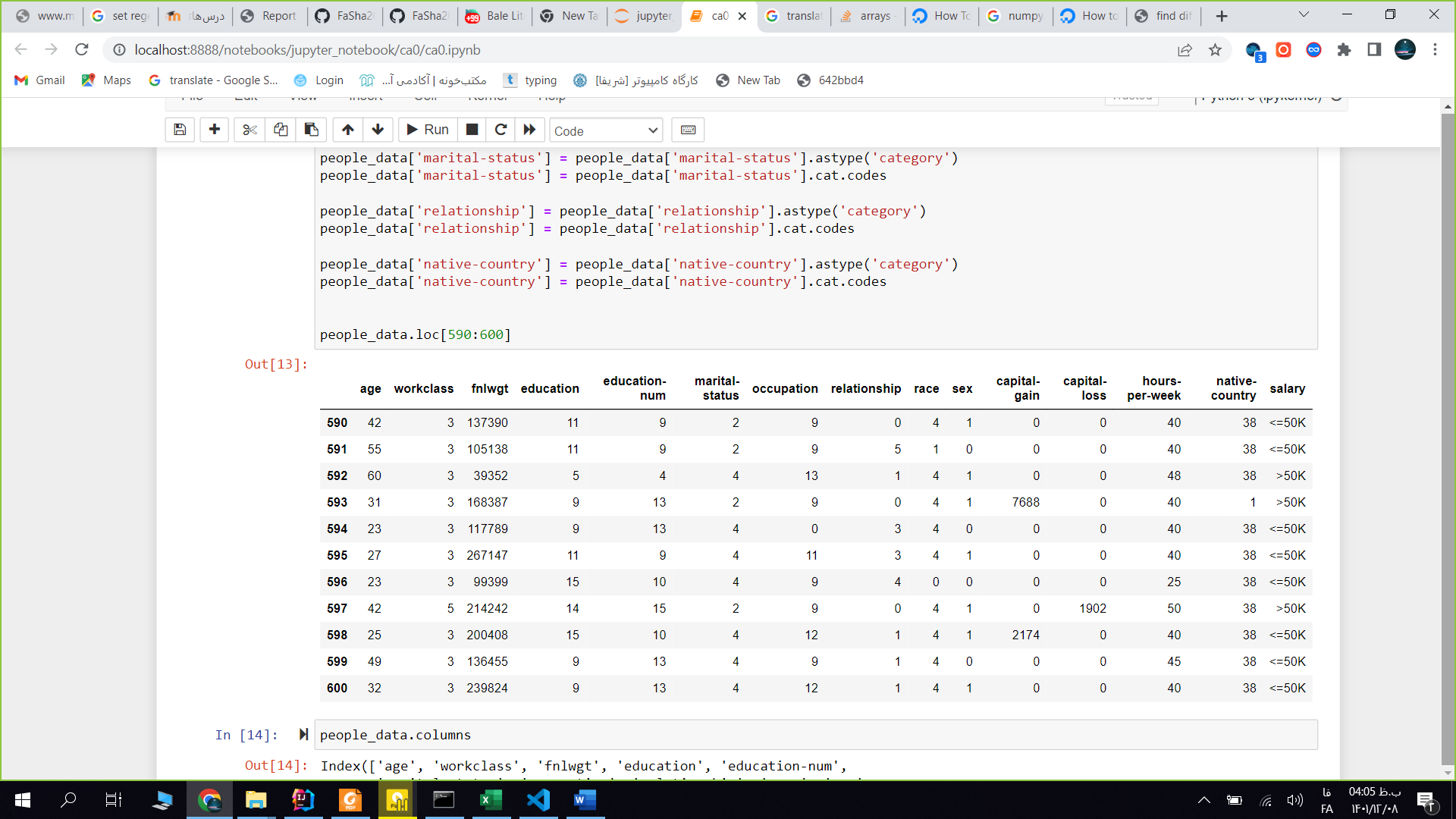
بررسی ستون های عددی و دسته ای:

با کمک ستون Dtype خروجی تابع info می توان فهمید ستون های age, fnlwgt, education-num, capital-gain, capital-loss, hours-per-week عددی و workclass, education, occupation, relationship, race, sex, native-country, salary دسته ای هستند.



Label encoding ستون های دسته ای با کمک توابع زیر که مقادیر مختلف هر ویژگی را پیدا کرده و به جای هر کدام یک کد اختصاص می دهند.

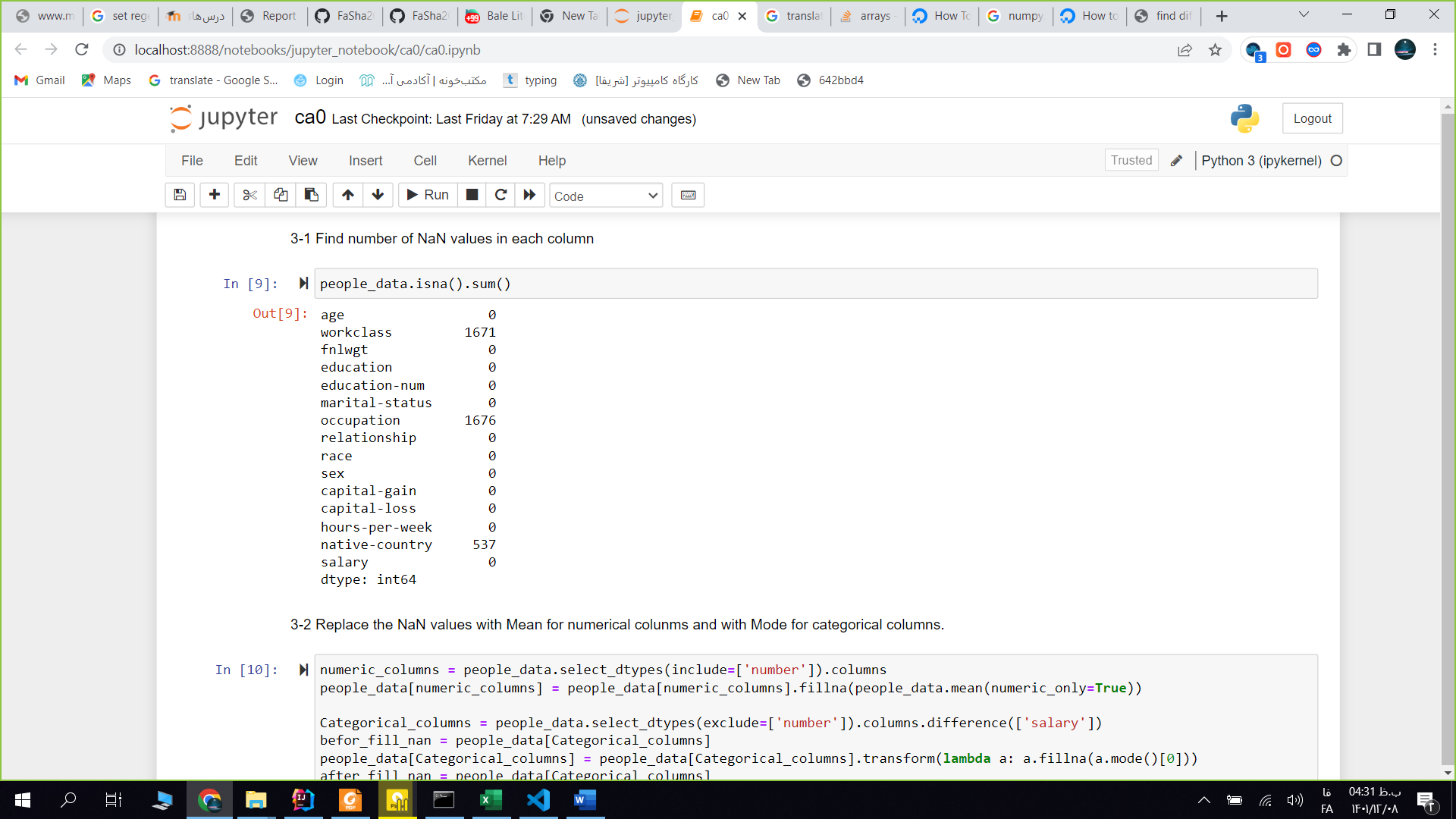




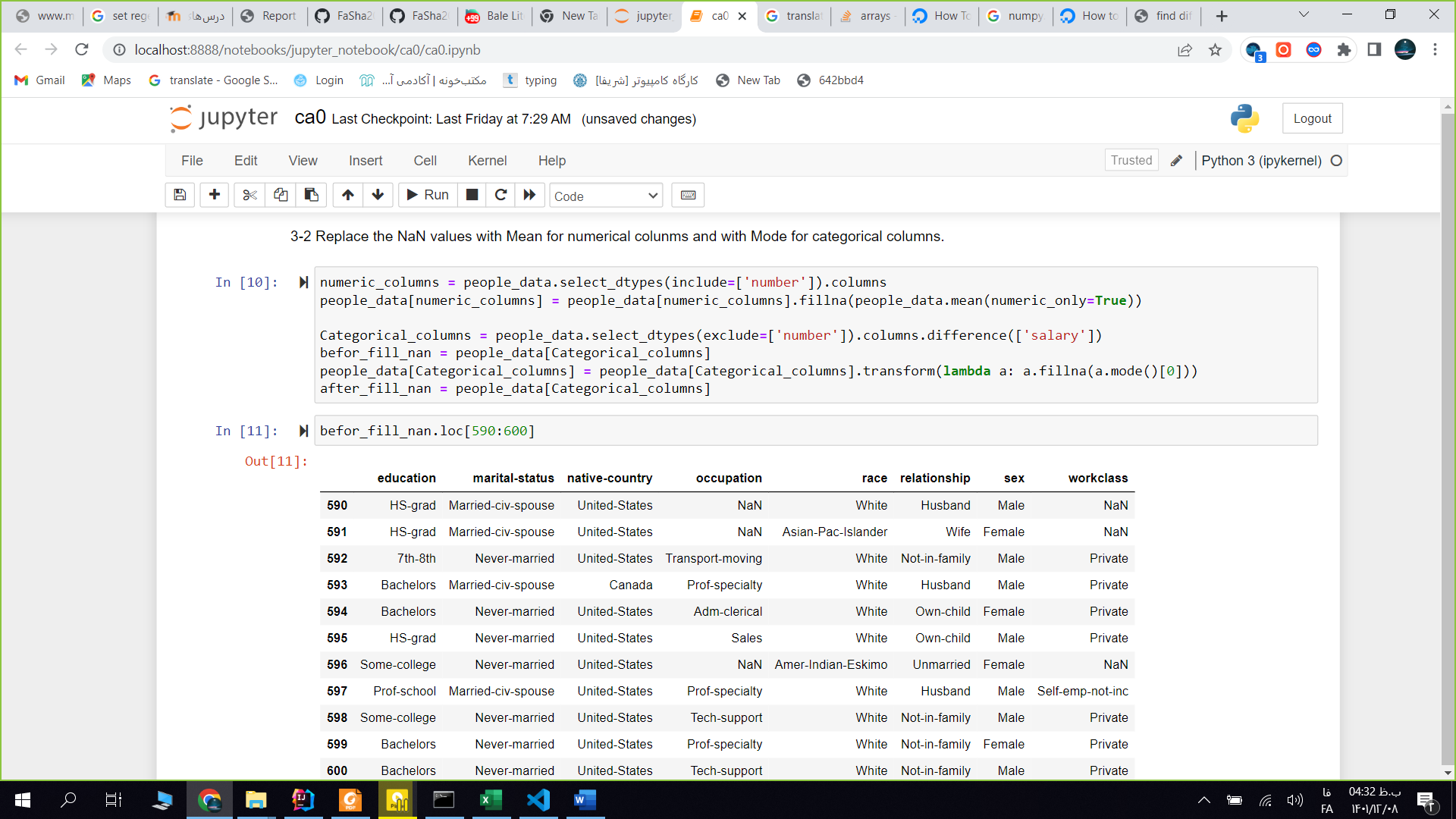
بخشی از دیتا فریم بعد از label encode شدن

3-

پیدا کردن ستون هایی که داده های تهی دارند و پر کردن آنها.

تعداد nan های هر ستون

این داده های تهی را می توان در ستون های عددی با مقدار میانگین ستون و در ستون های دسته ای با mod ستون پر کرد.



فواید پر کردن خانه های خالی با میانگین: این کار برای تعداد داده کم جواب می دهد و باعث می شود مقدار میانگین کل ثابت بماند.

معایب پر کردن خانه های خالی با میانگین: در تعداد داده زیاد باعث می شود انحراف معیار زیاد شود و به تخمین خوبی نرسیم.

4-

حذف ستون هایی که مقادیر تکراری کمی دارند و برای یادگیری به ما کمک نمی کنند،

rate\_unique\_fnlwgt = 20337 / 29825 = 0.68

rate\_unique\_hours-per-week = 94 / 29825 = 0.003

rate\_marital-statust = 7 / 29825 = 0.0002

rate\_age = 72 / 29825 = 0.002

rate\_unique\_workclass = 9 / 29825 = 0.0002

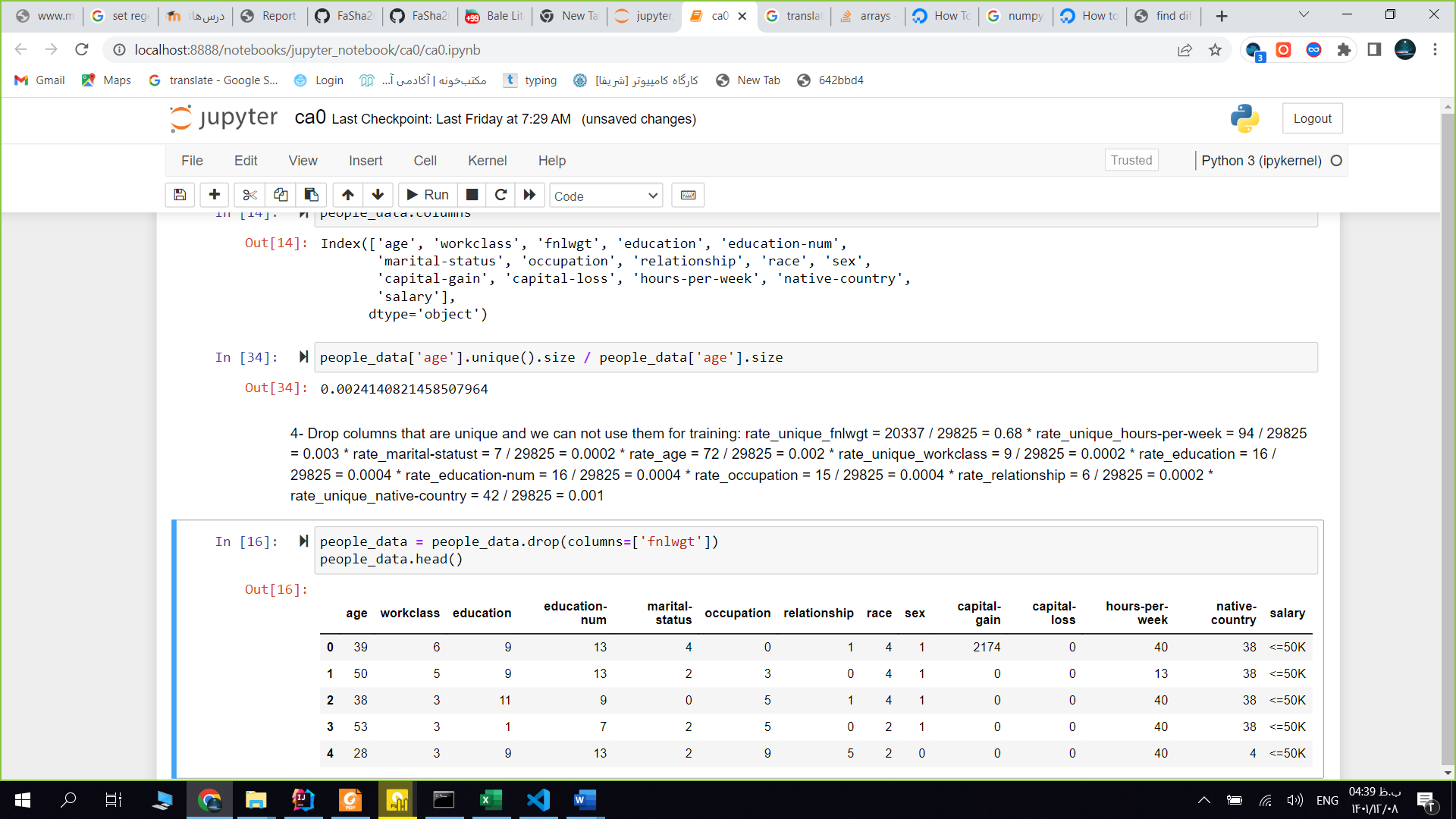
rate\_education = 16 / 29825 = 0.0004

rate\_education-num = 16 / 29825 = 0.0004

rate\_occupation = 15 / 29825 = 0.0004

rate\_relationship = 6 / 29825 = 0.0002

rate\_unique\_native-country = 42 / 29825 = 0.001



که درصد تنوع fnlwgt نسبت به بقیه خیلی زیاد است و کمک زیادی به تحلیل ما نمی کند.

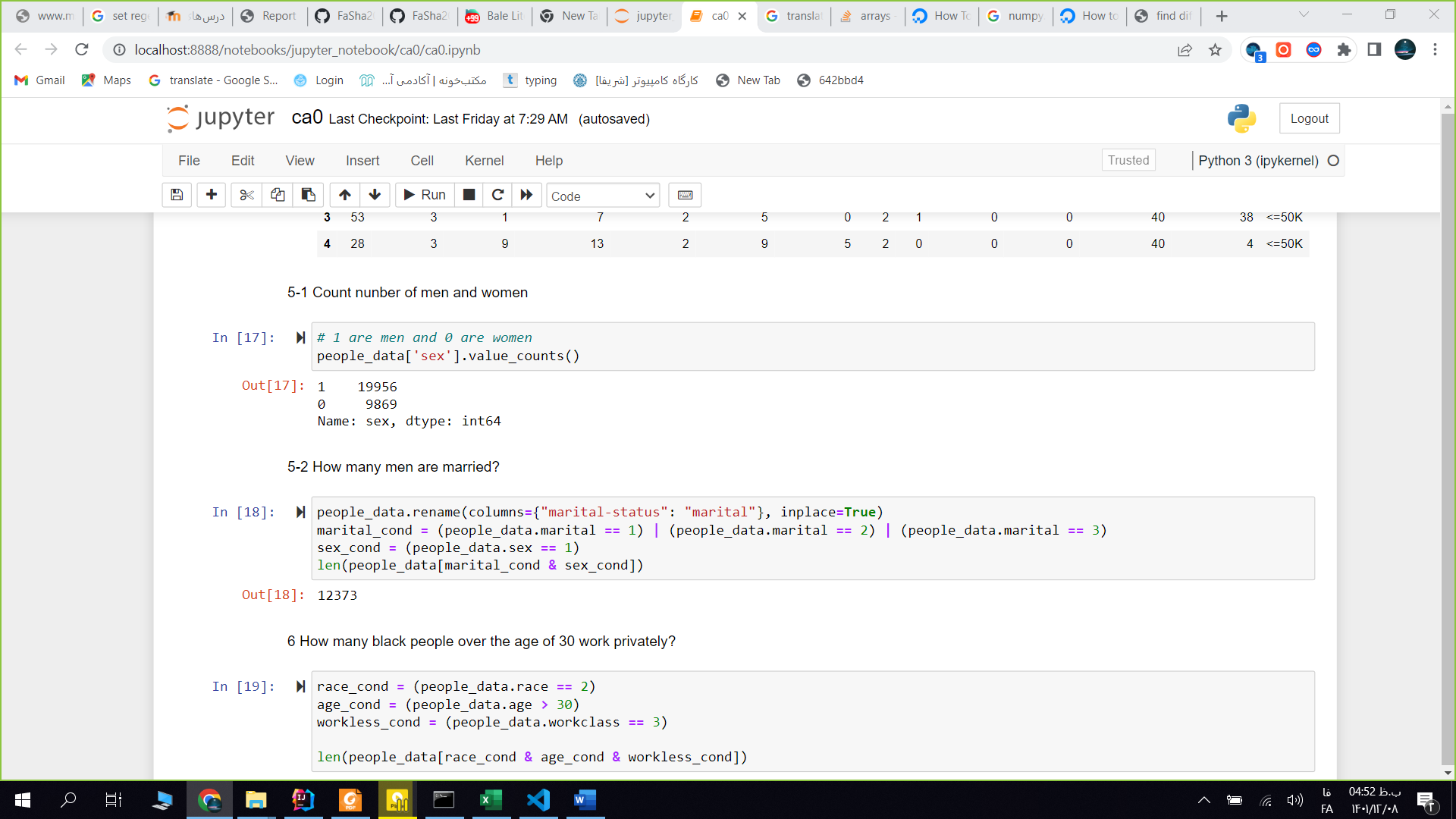
5-

در این بخش به کمک Boolean mask ها شرایط هر کدام را مشخص کرده و با عبور کل دیتا از این mask ها سایز مجموعه دیتای خواست مسله را حساب می کنیم.

چند نفر مرد و زن داریم؟ 19956 تا مرد و 9869 زن ( با استفاده از تابع value\_counts که تعداد تکرار مقادیر مختلف هر ستون sex را می دهد)

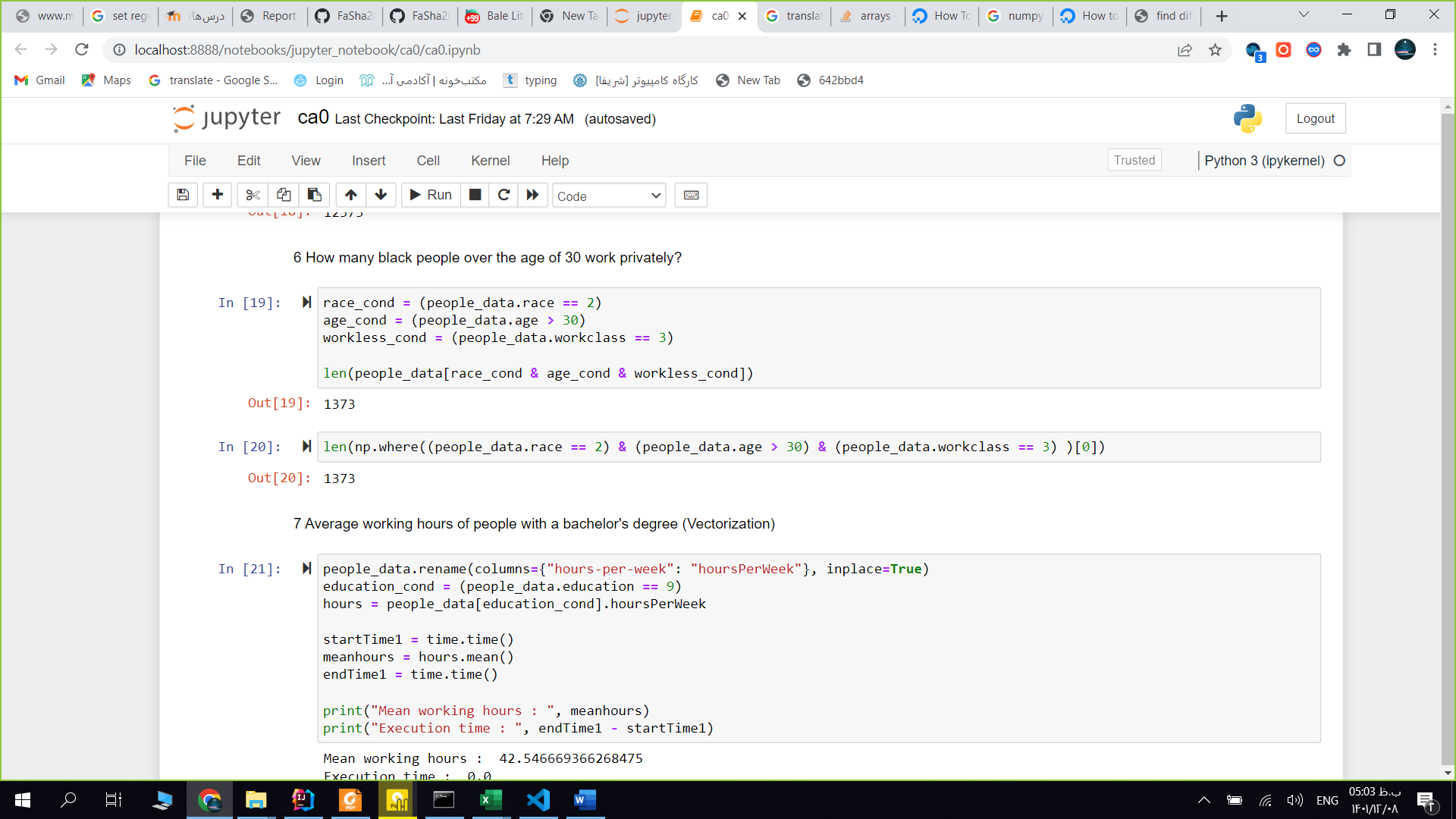
چند نفر از مردان متاهلند؟ 12373

همه این موارد را متاهل حساب کردیم.)Married-civ-spouse, Married-spouse-absent, Married-AF-spouse(



6-

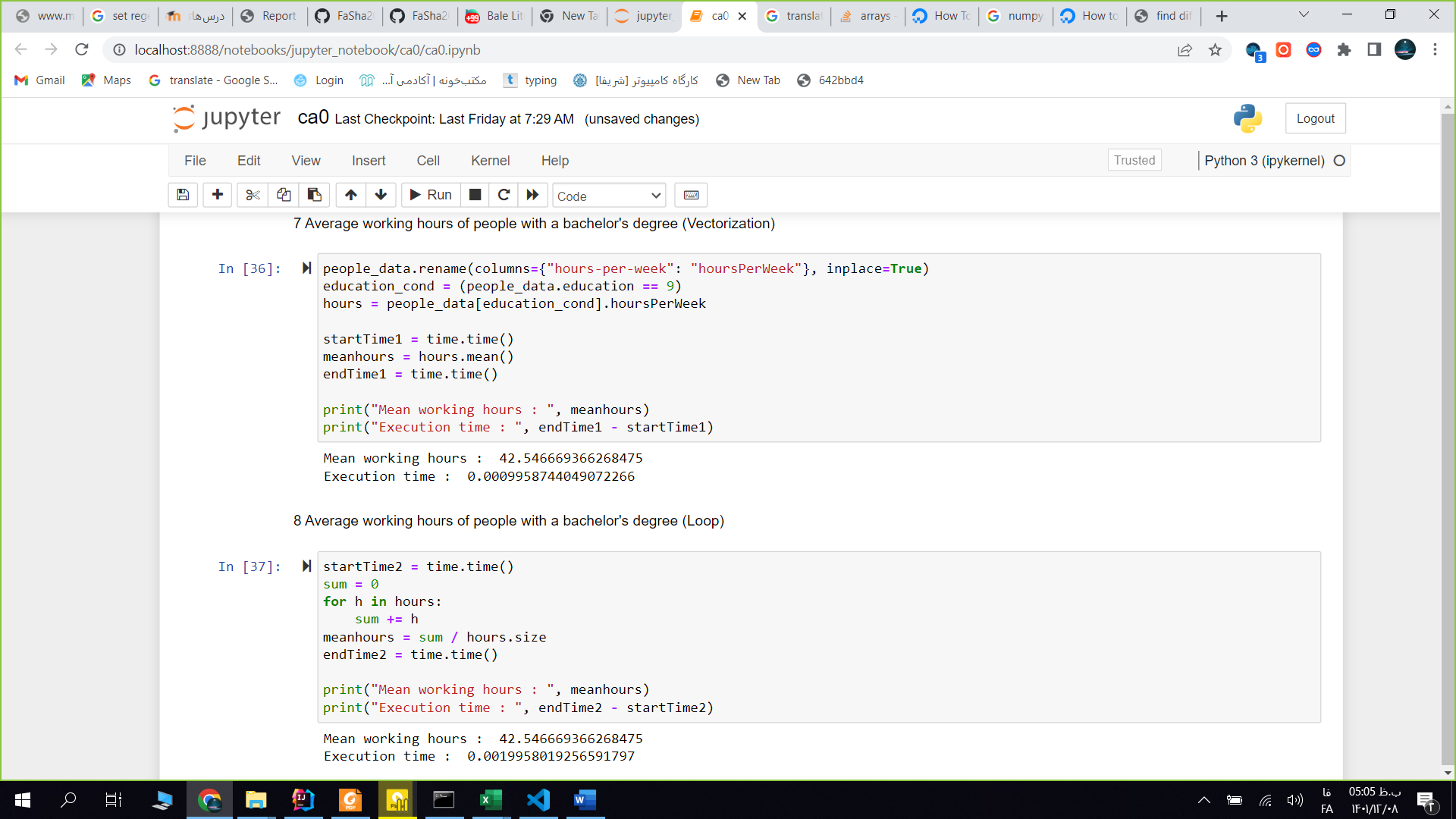
چند نفر سیاه پوست ، بالای سی سال، کار کردن انفرادی؟ 1373



اینجا به دو روش فیلتر کردن دیتا را انجام داده ایم. روش اول همان Boolean masking که بهتر و گویا تر است و در روش دوم از کتابخانه numpy و متد where استفاده کرده ایم که دیتا با شرایط داده شده را جدا می کند.

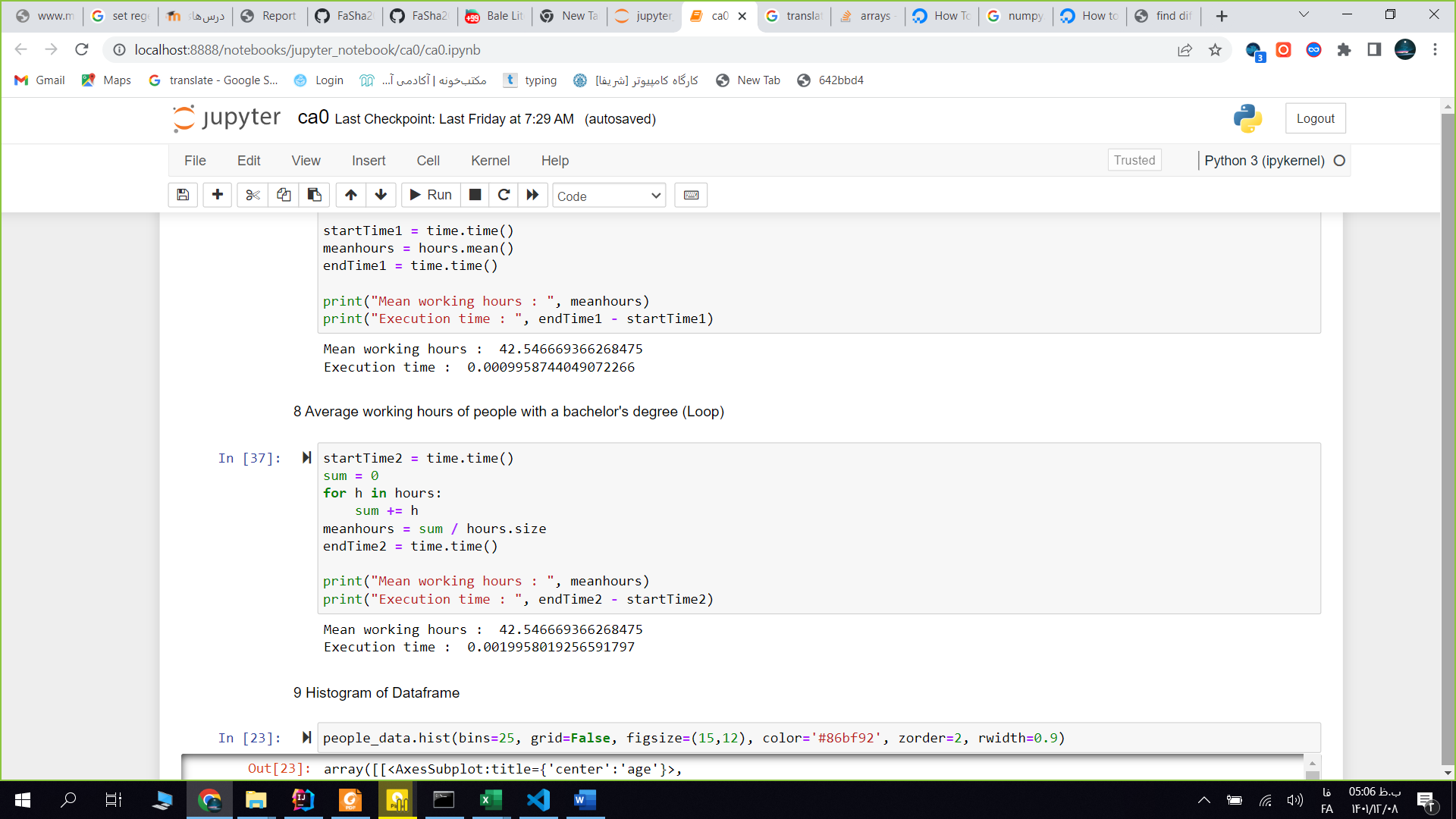
7-

میانگین ساعت کاری افرادی که لیسانسه هستند؟(با وکتوریزیشن)



8-

میانگین ساعت کاری افرادی که لیسانسه هستند؟(با حلقه)



زمان در وکتوریزیشن :

0.0009982585906982422

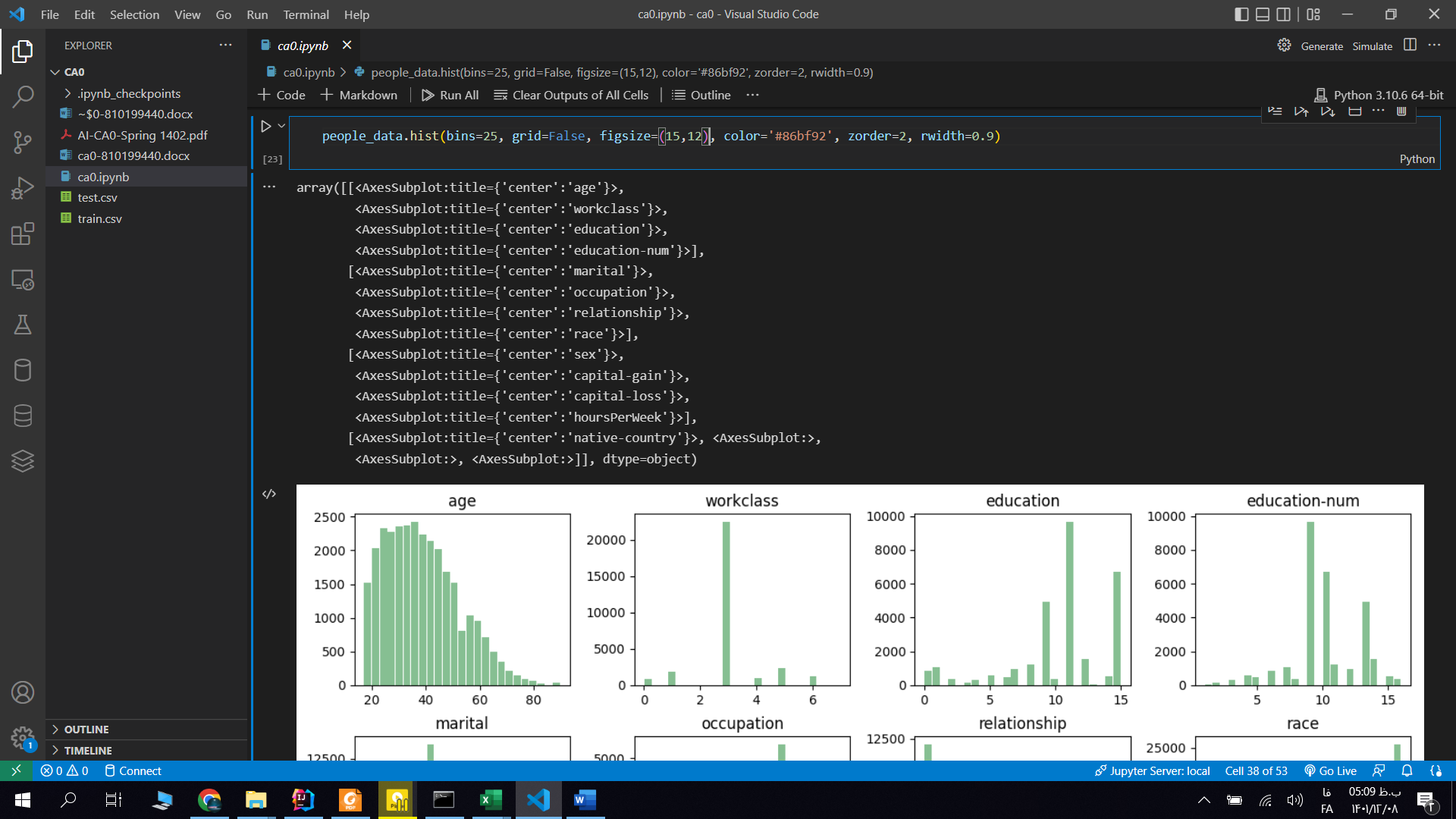
زمان در حلقه :

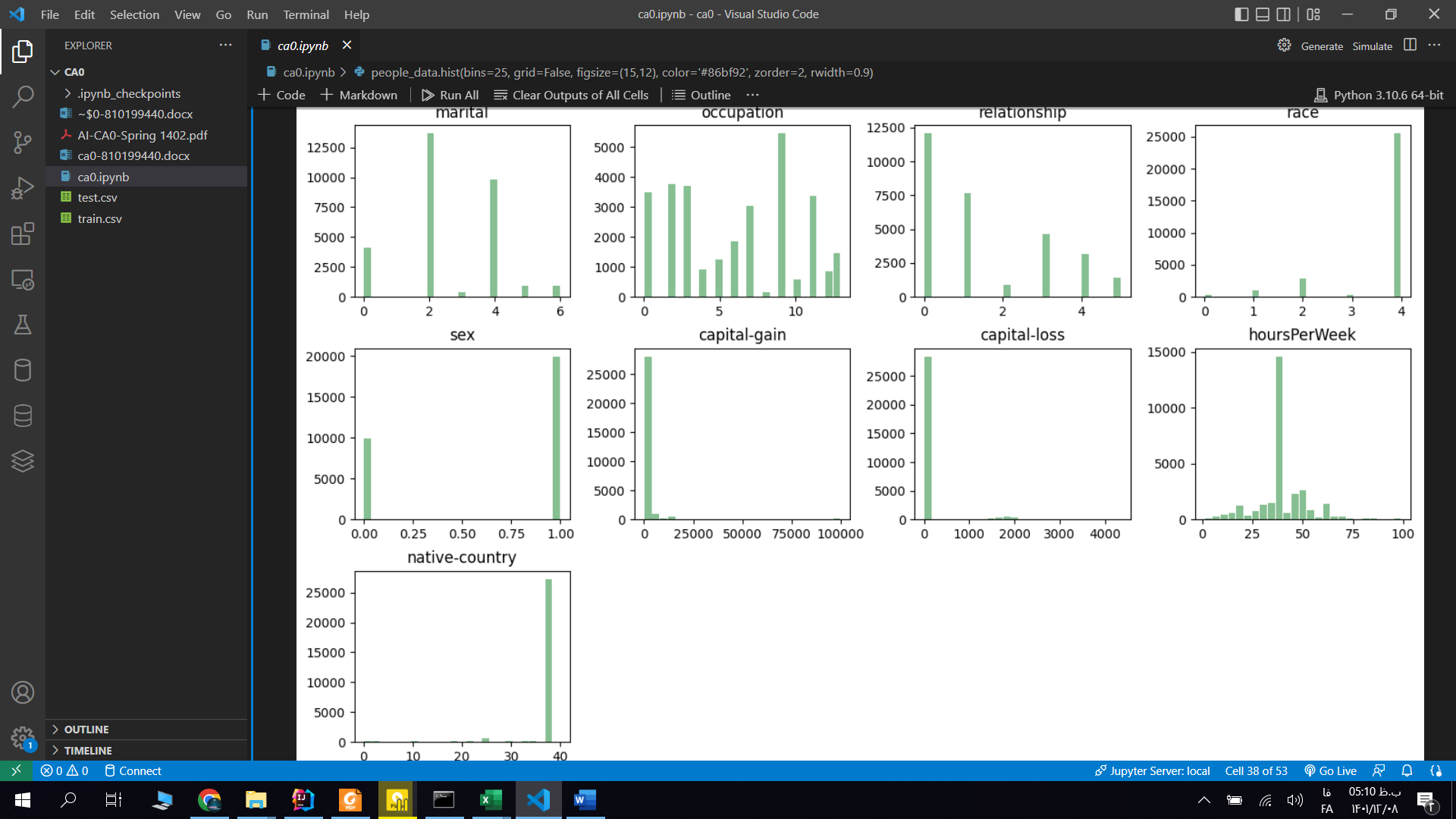
0.001994609832763672

تقریبا در حلقه دوبرابر زمان لازم است چون در وکتوریزیشن دیتا ها به صورت موازی پراسس می شوند و زمان بسیار کمتری صرف می کنند.

9-

نمایش توزیع هر ستون از داده روی نمودار هیستوگرام

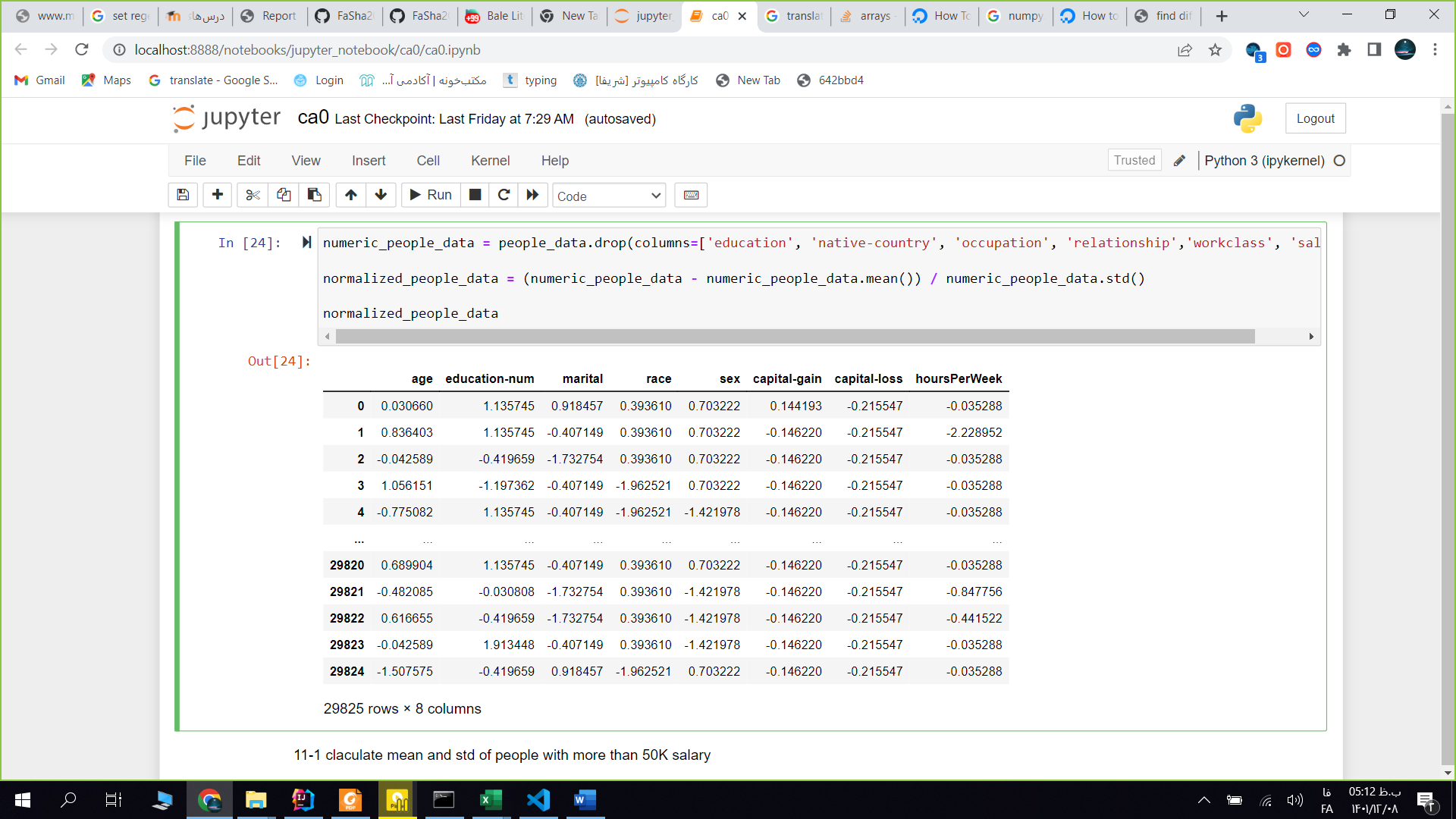




10-

نرمال سازی دیتا ها

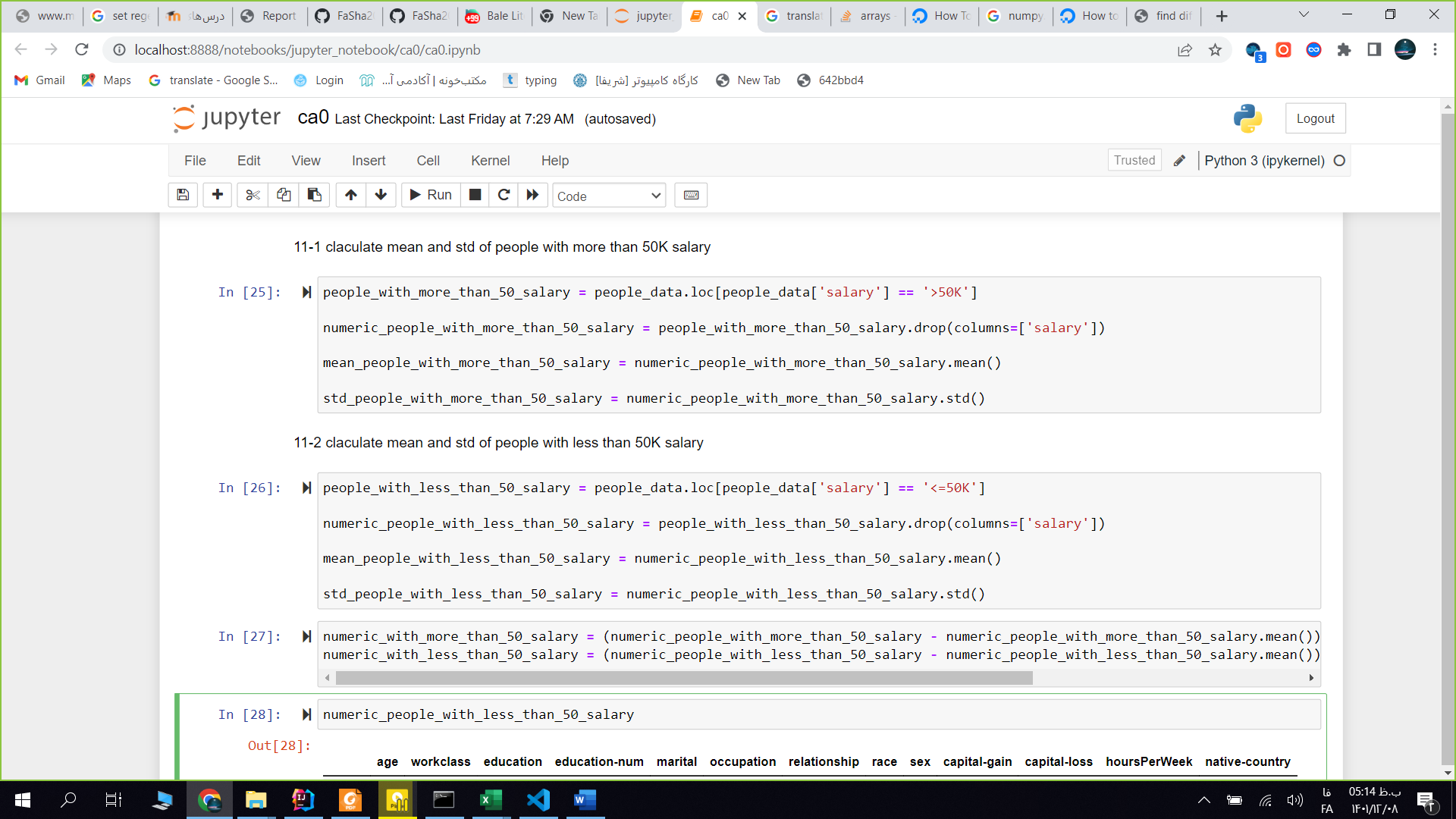
از انجایی که میانگین گیری از کد هایی که به داده های غیر عددی نسبت دادیم معنا ندارد، ابتدا ستون های غیر عددی را حذف می کنیم،

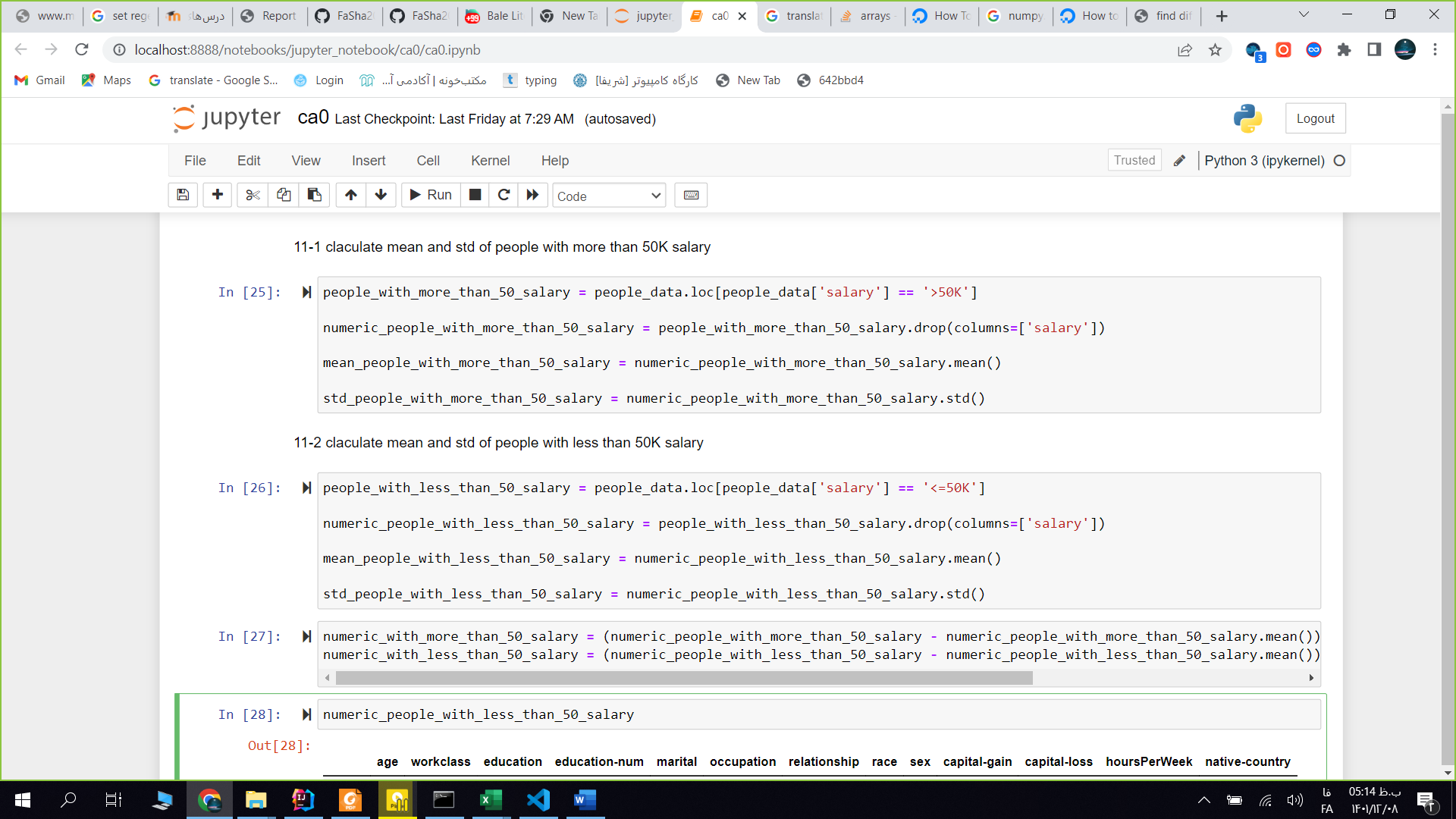


11-

محاسبه میانگین و انحراف معیار برای حالت دستمزد بیشتر از 50 و کمتر از 50.

به کمک تابع loc مواردی که دستمزد بیشتر و کمتر از 50 هزار دارند را از هم جدا می کنیم و به کمک توابع mean و std میانگین و انحراف معیار هر یک را به صورت vectorization حساب می کنیم.

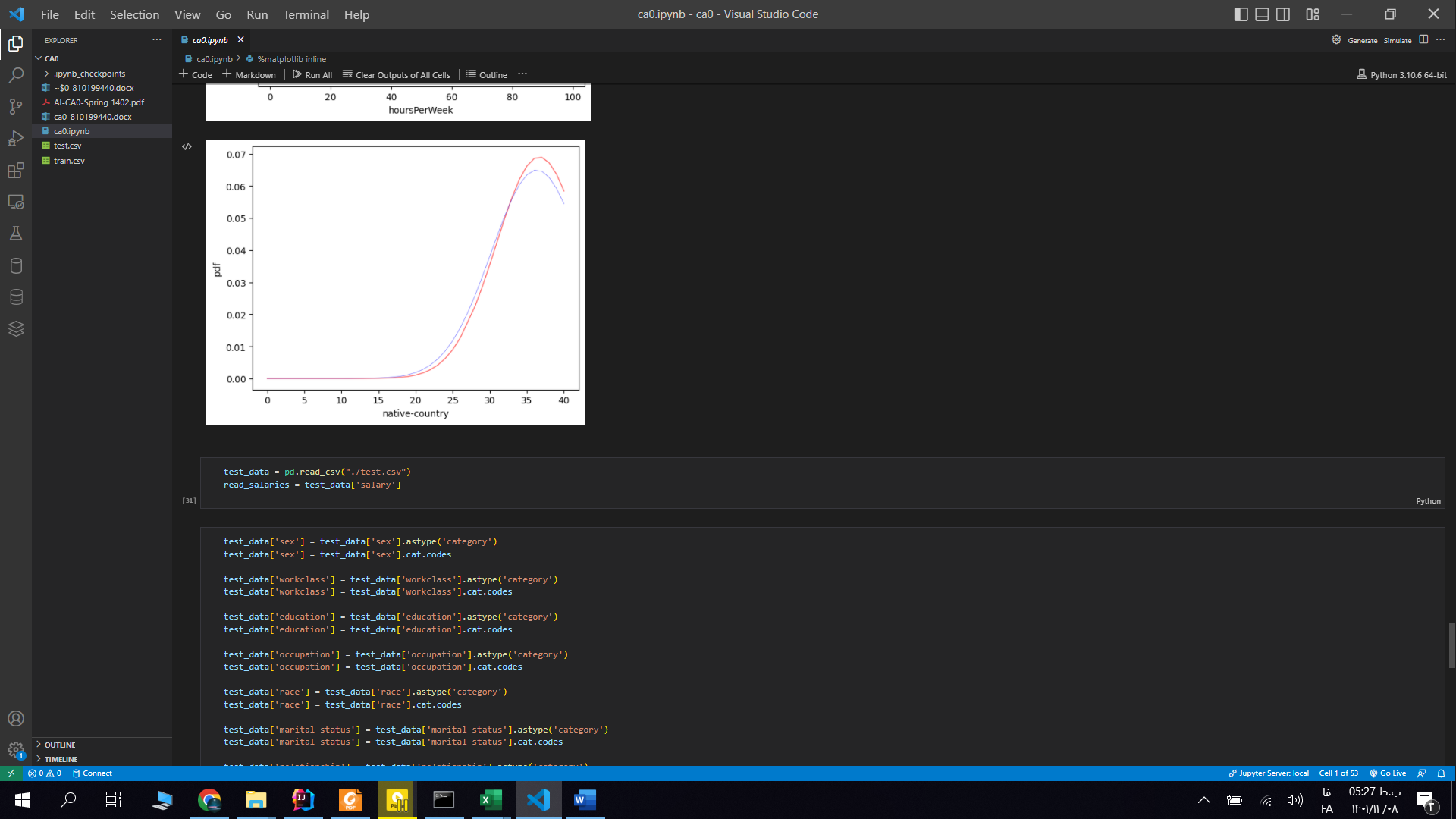


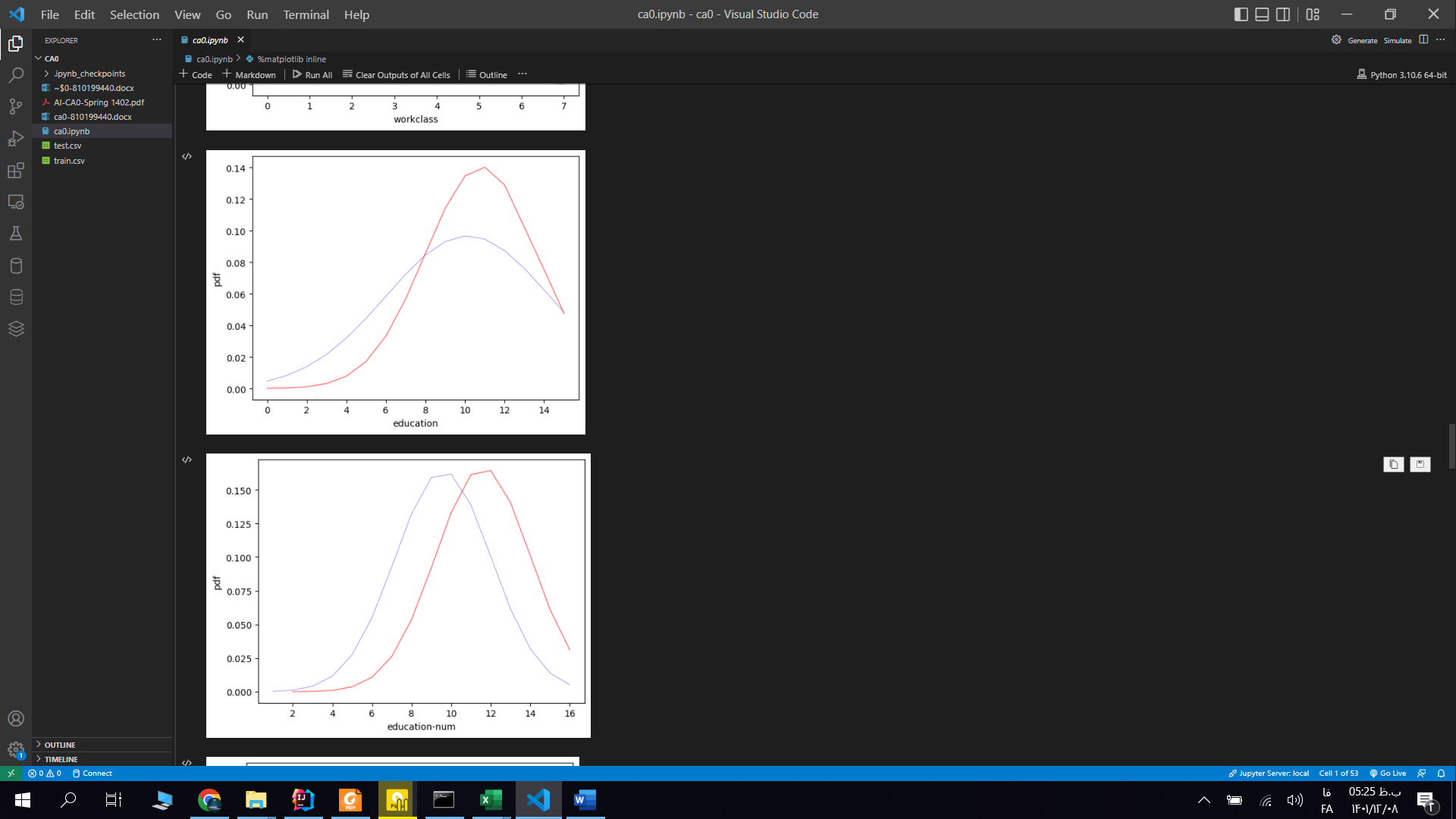
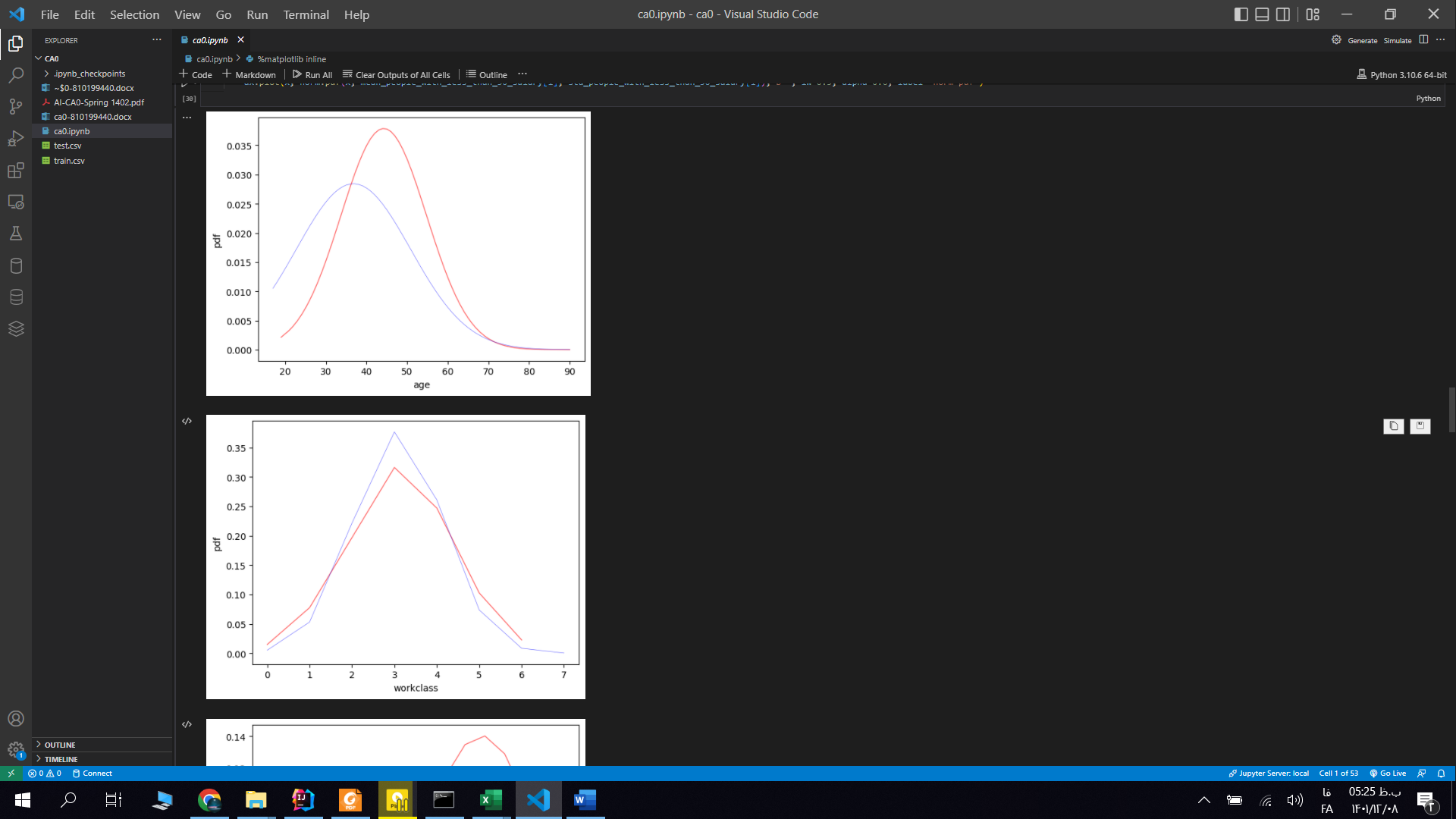


نمایش نمودار PDF مرتبط با هر ویژگی:

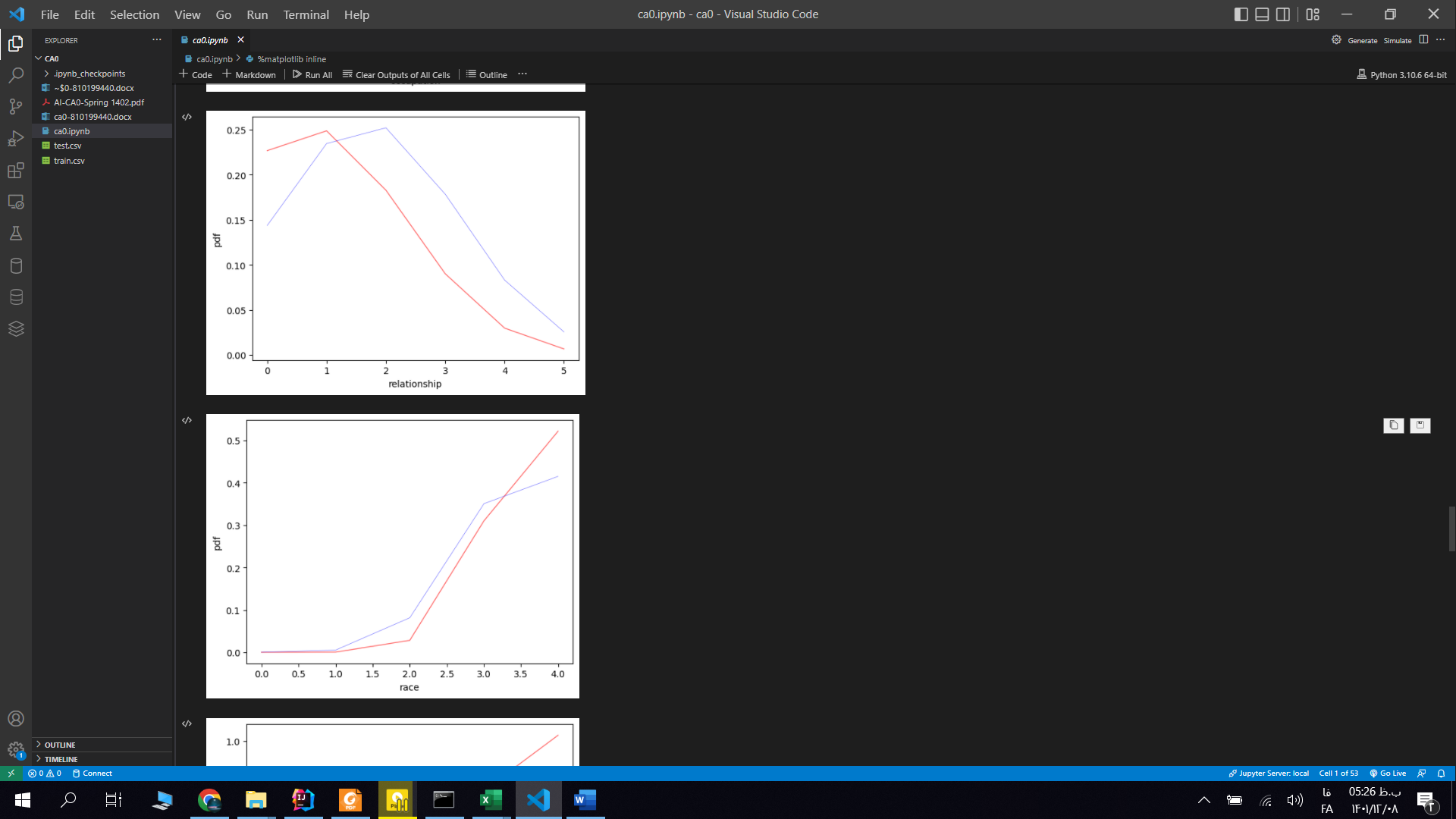
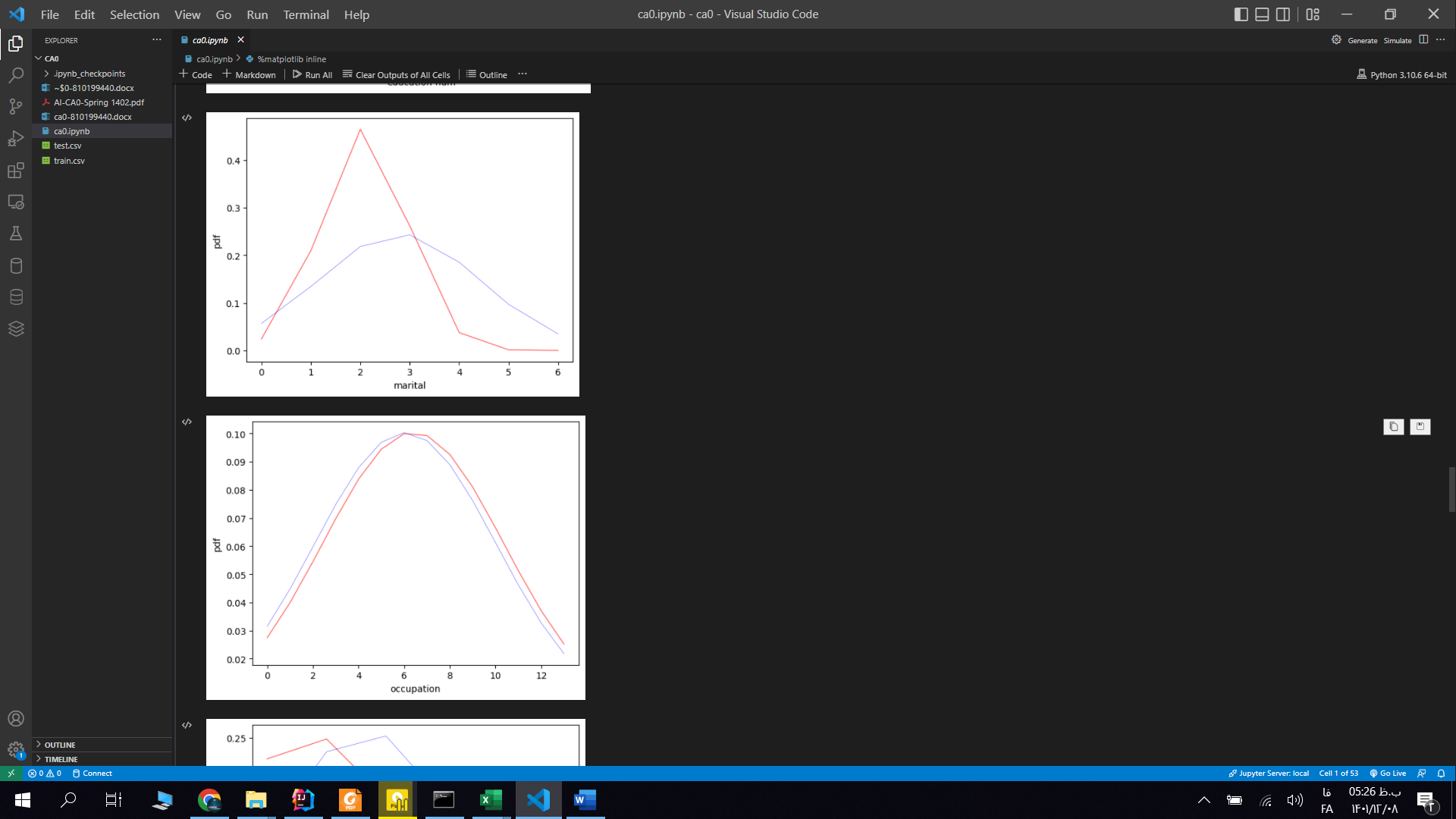


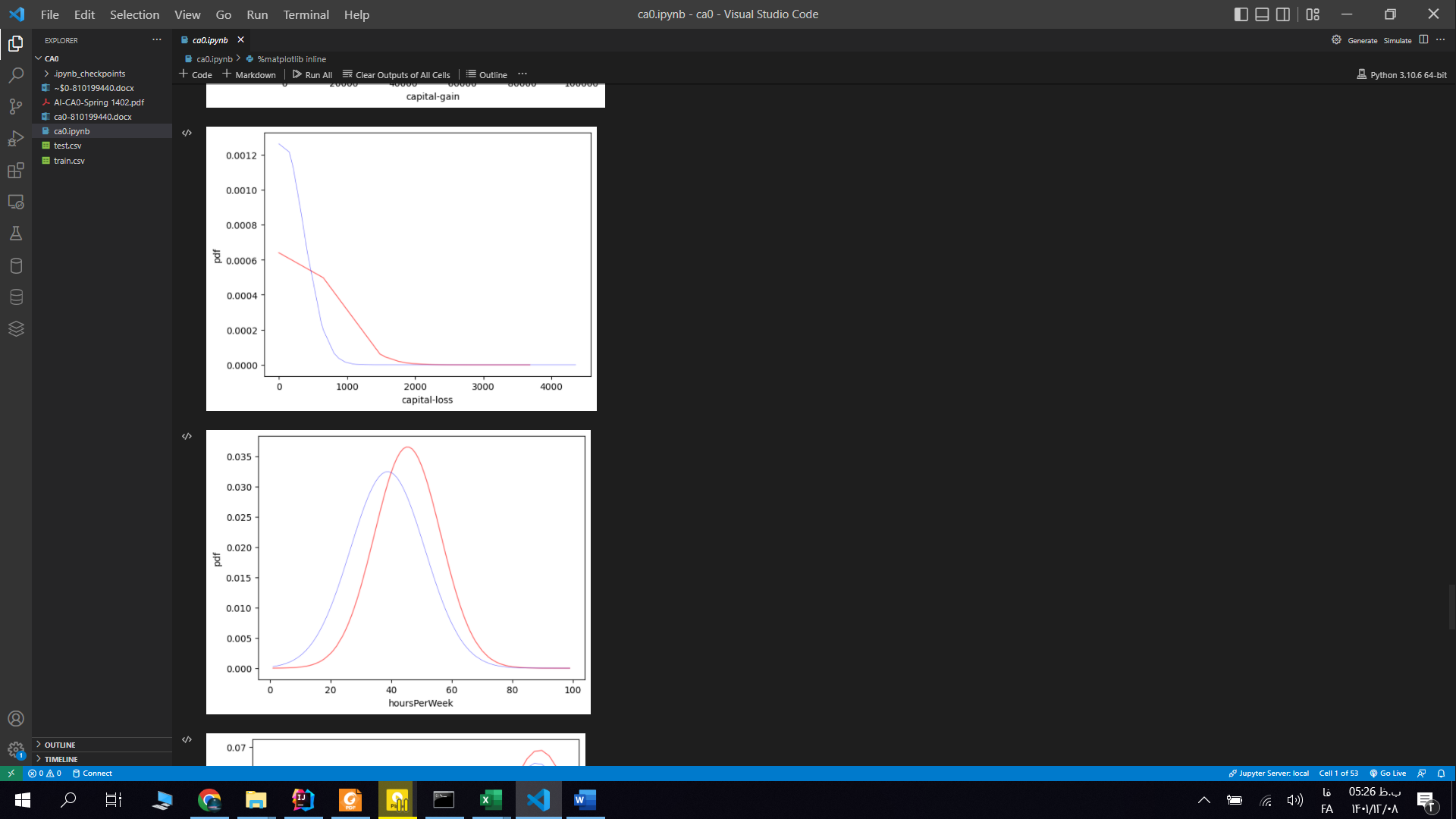
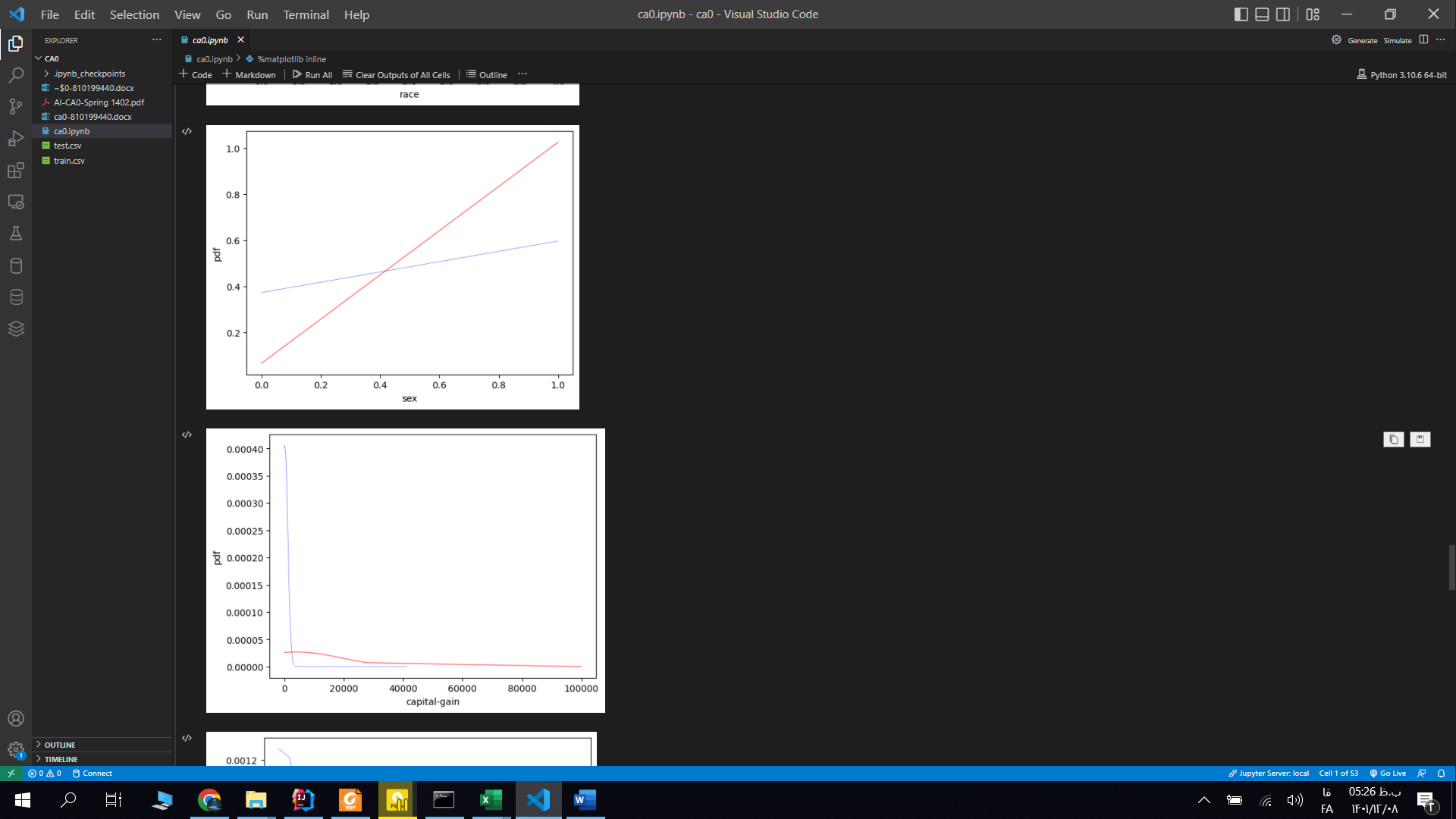
برای هر ستون که نشان دهنده یک ویژگی است نمودار pdf را رسم می کنیم. برای محاسبه تابع pdf از متد pdf در کتابخانه spicy.stat استفاده می کنیم که با گرفتن linespace, mean , std تابع پی دی اف را می سازد. داده ای که به عنوان ورودی به پی دی اف می دهیم یکبار داده های مرتبط با افراد با دستمزد بالاتر از 50 هزار و یکبار داده های مرتبط با افراد با دستمزد کمتر از 50 هزار است. و در نهایت هر دو را روی یک نمودار می کشیم.نمودار قرمز رنگ مرتبط با بالای 50 هزار و آبی زیر 50 هزار است.

 در این نمودار پیوستگی زیادی بین دو نمودار می بینیم که یعنی کشور مادری تاثیرچشمگیری در دستمزد ندارد و اگر مثلا کشور مکزیک به عدد 30، encode شده باشد، احتمال اینکه فردی که در مکزیک متولد شده بیشتر از 50 هزار یا کمتر از آن بگیرد برابر است.

در نمودار education تفاوت چشمگیر تری می بینیم، و به وضوح کسی که 10 کلاس درس خوانده است احتمال زیادی دارد که بیشتر از 50 هزار دستمزد بگیرد.

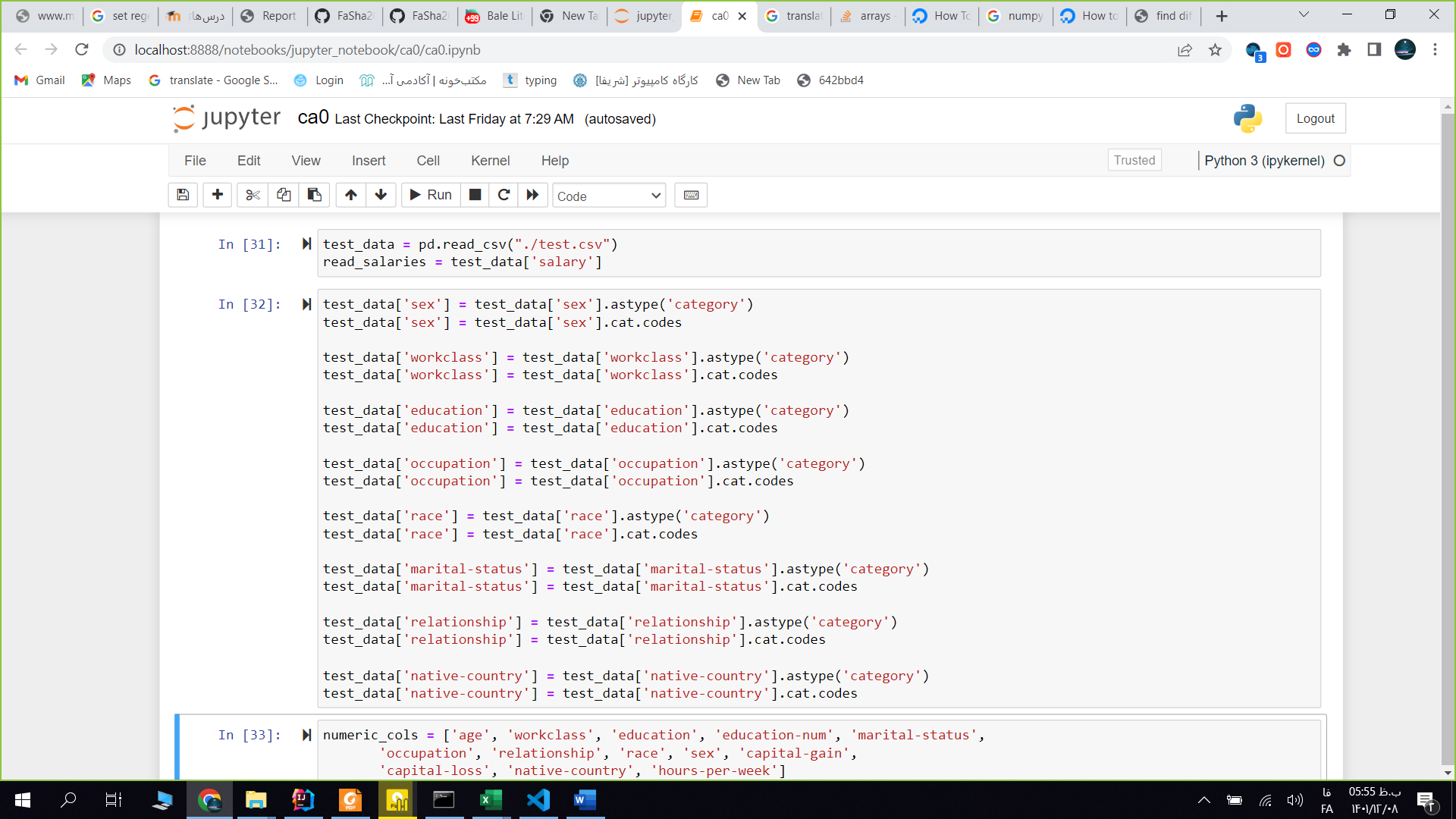


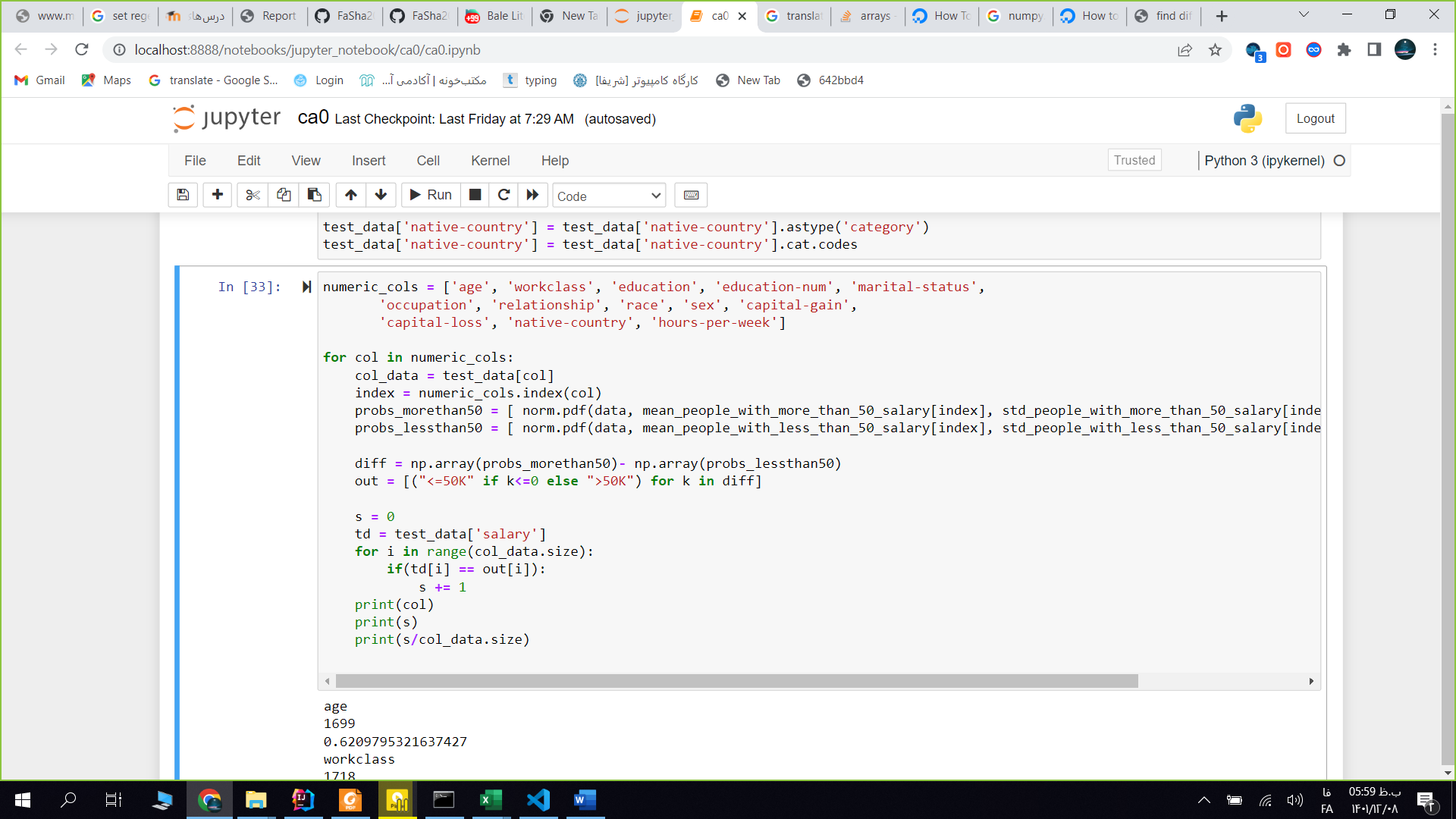


و در نمودار سود سرمایه ای capital gain می بینیم که اگر سود سرمایه ای کسی صفر است تقریبا صد در صد حقوق کمتر از 50 هزار دارد و در غیر این صورت حقوق بالاتری دارد. و به نظر می رسد که این ویژگی بیشتر از بقیه توانسته دو نمودار را از هم تفکیک کند که به معنای تاثیرگذاری بیشتر آن در مقدار دستمزد است.

حالا برای مقایسه دقیق تر هربار با توجه به یک ویژگی ، دیتای فایل test را پیش بینی و با واقعیت مقایسه می کنیم تا میزان صحت پیش بینی با هر ویژگی را بسنجیم.

ابتدا فایل تست را در یک دیتافریم جدید لود می کنیم و مثل قبل ستون های غیر عددی را به عددی encode می کنیم، ستون دستمزد واقعی را هم دریک لیست جدا برای مرحله محاسبه خطا ذخیره می کنیم.

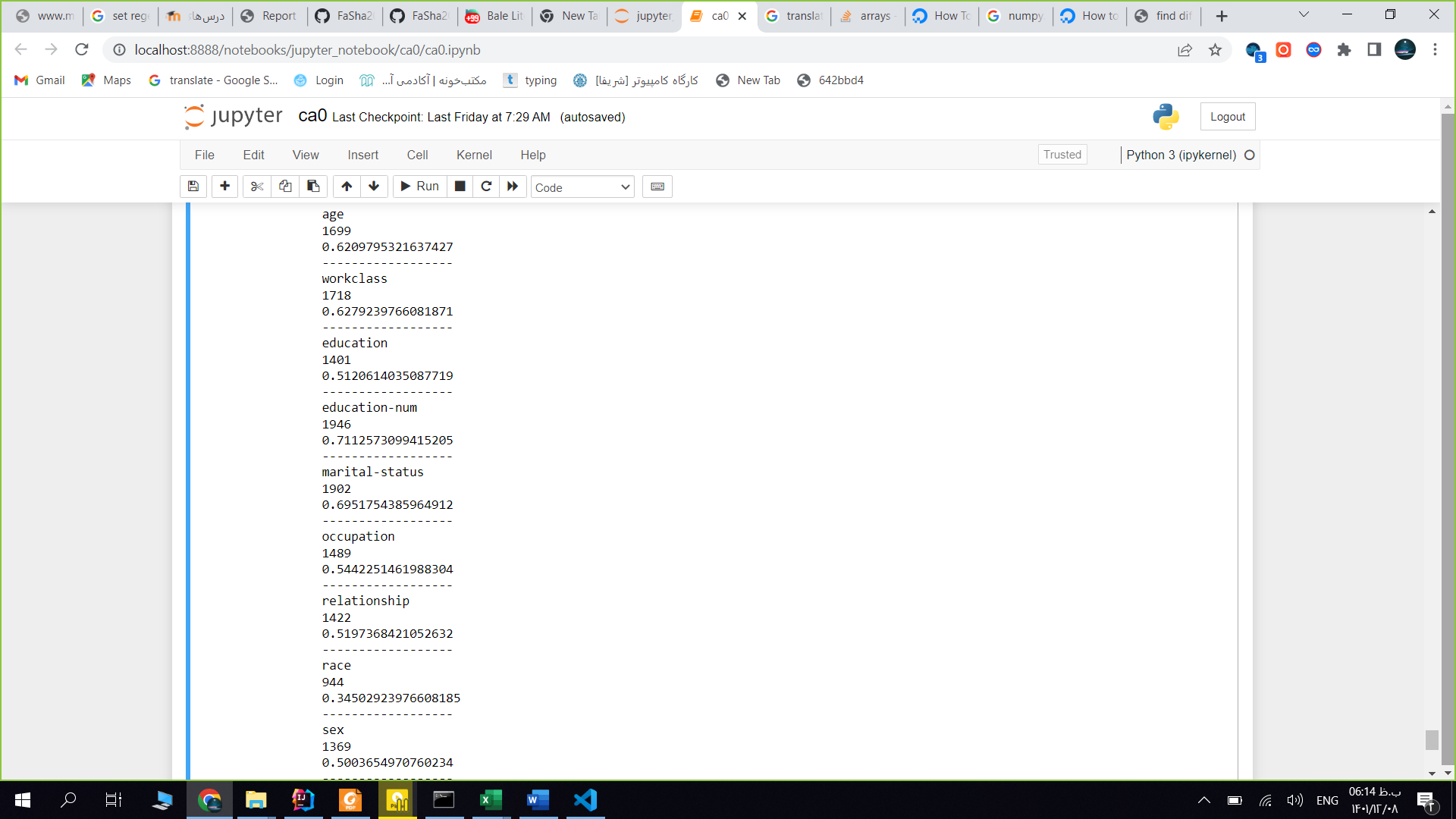
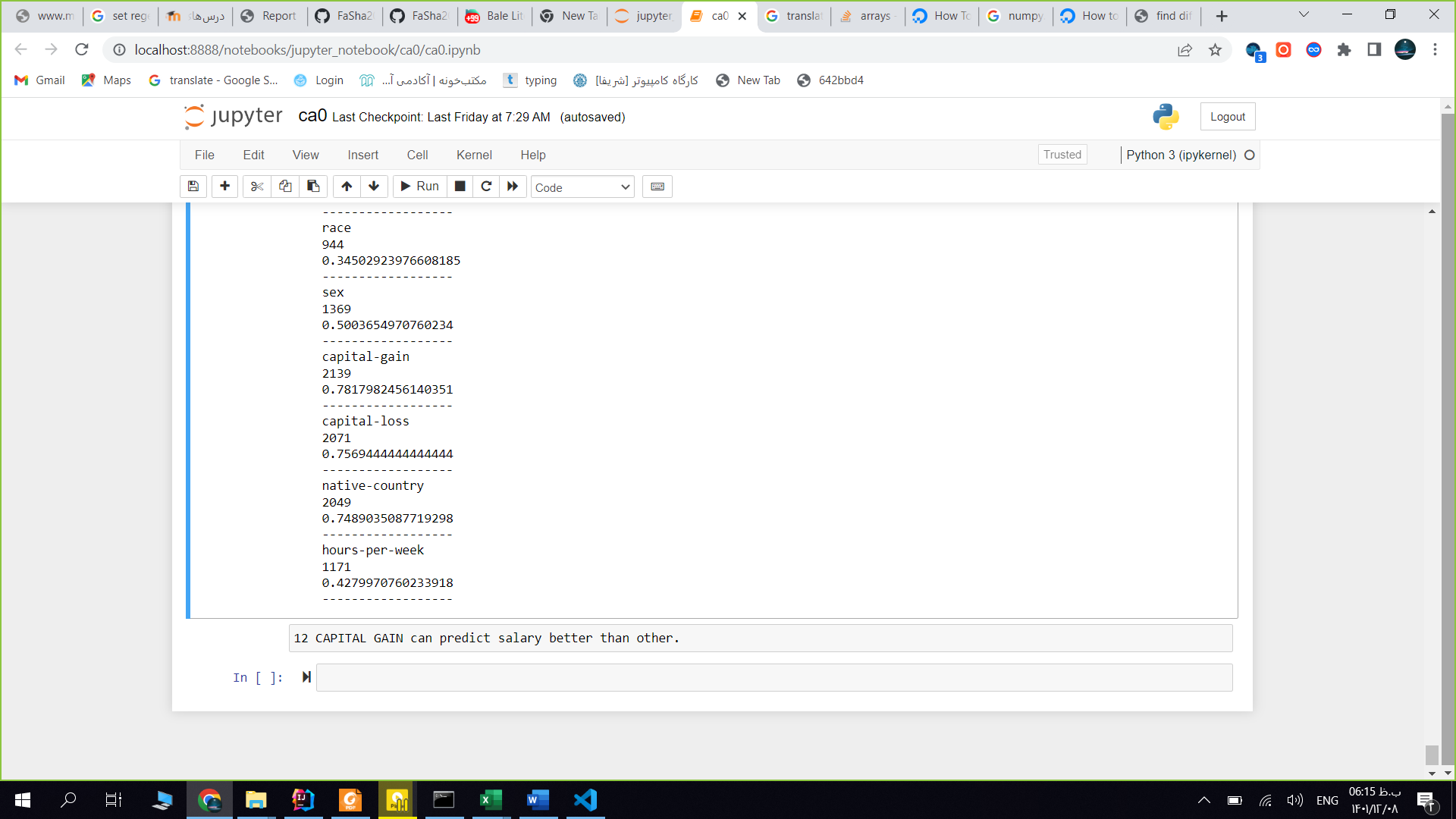




احتمال بالای 50 یا زیر 50 بودن دستمزد تحت هر شرایط یک ویژگی را بررسی و با هم مقایسه می کنیم، هر یک از احتمالات که بالاتر بود را به عنوان دستمزد آن فرد پیش بینی می کنیم و در out قرار می دهیم.

برای محاسبه خطا هم کافی است مقدار واقعی دستمزد را با مقدار پیش بینی شده مقایسه کنیم و تعداد پیش بینی ها غلط به کل را حساب کنیم.

تعداد موارد درست پیش بینی شده و درصد صحت پیش بینی هر یک از ویژگی ها را در خروجی زیر می بینیم:

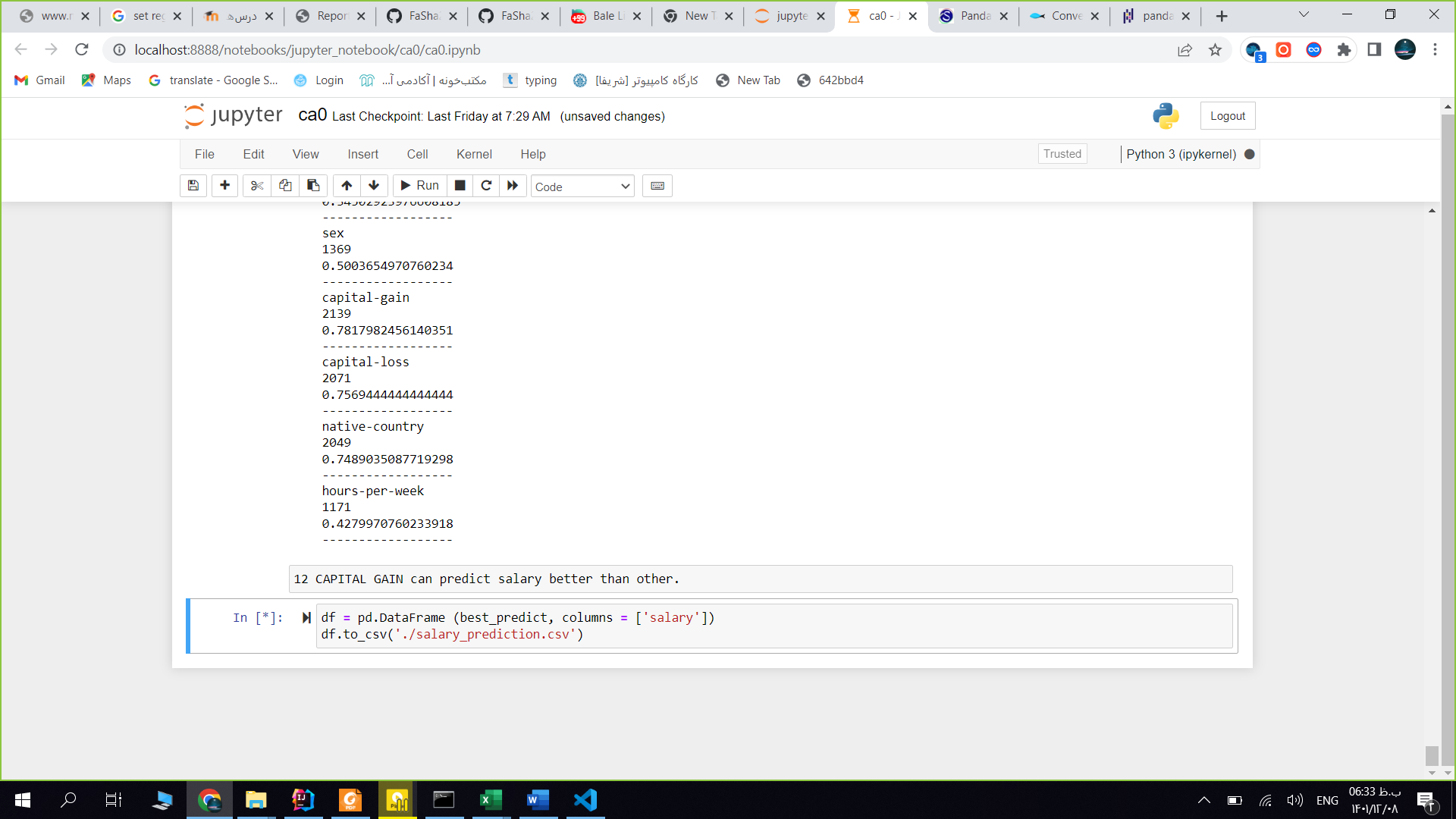


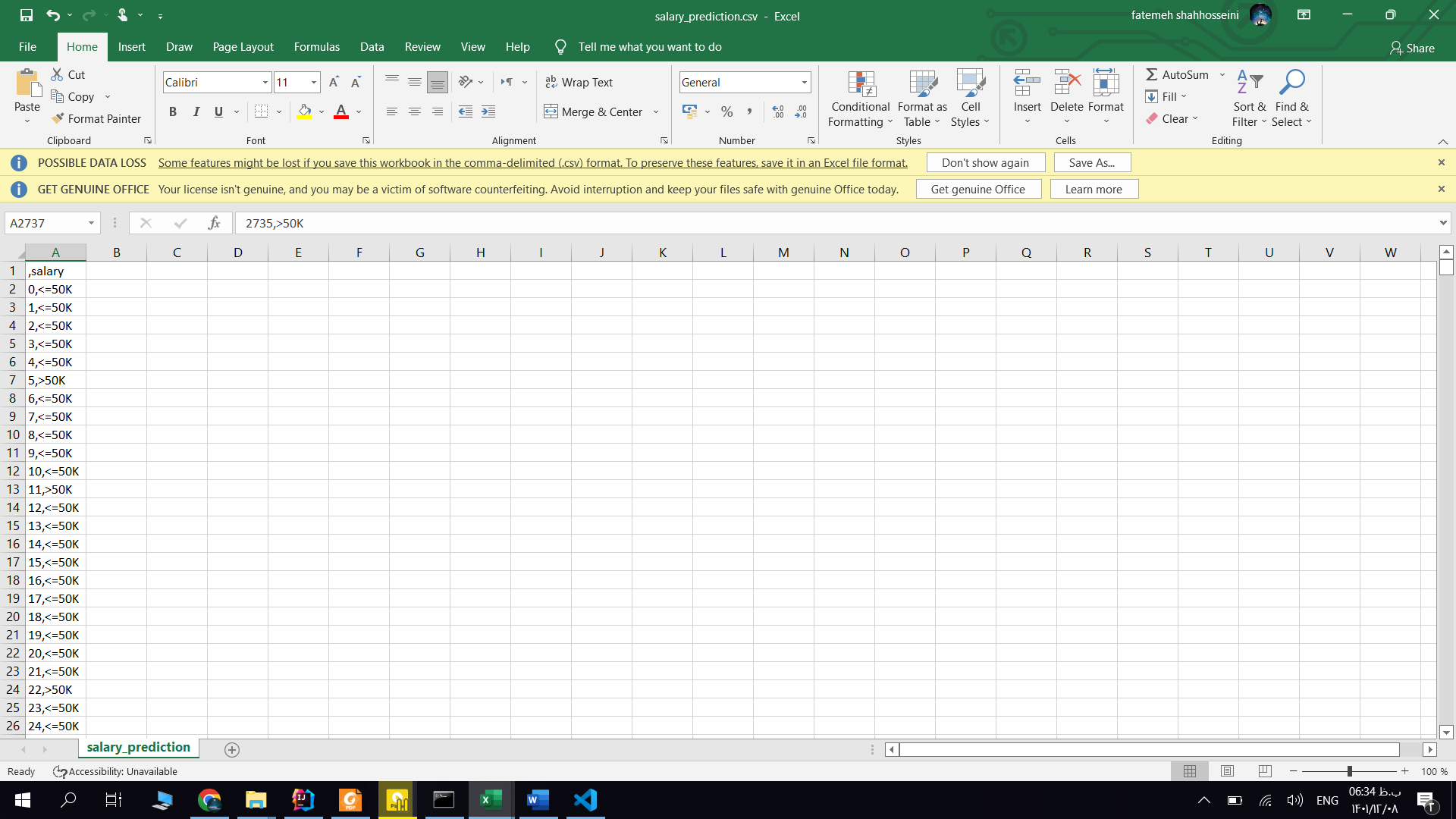
همان طور که پیش بینی کرده بودیم، ویژگی capital-gain ویژگی دقیق تری است و با صحت 78 درصد و خطای حدودا 22 درصدی بهتر از بقیه می تواند salary را تعیین کند.

علل وجود خطا:

وجود خطا دلایل متعددی دارد مثلا بخشی از آن به این خاطر است که دیتای tarin کاملا ساده است و ممکن است به خوبی همه احتمالات را پوشش ندهد و با توجه به تعداد ستون ها تعداد داده کم باشد. و بخشی از خطا هم می تواند به خاطر این باشد که دستمزد تنها با یک ویژگی ارتباط ندارد و ترکیبی از ویژگی های متفاوت است و مثلا اگر دو ویژگی داشته باشیم که اثر معکوسی روی دستمزد داشته باشند و هر دو بالا باشند، تنها با تکیه بر یک ویژگی نمی توانیم دستمزد را به درستی پیش بینی کنیم.

در نهایت مقدار پیش بینی توسط ویژگی capital gain به همراه اندیس فرد متناظرش را به عنوان دستمزد فرد در یک فایل csv به نام salary prediction ذخیره می کنیم.





برخی منابع :

<https://sparkbyexamples.com/pandas/pandas-write-dataframe-to-csv-file/#:~:text=By%20using%20pandas.,index%20as%20the%20first%20column>.

<https://datatofish.com/list-to-dataframe/>

<https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_csv.html>

[https://stackoverflow.com/questions/10712002/](https://stackoverflow.com/questions/10712002/create-an-empty-list-with-certain-size-in-python)

<https://towardsdatascience.com/boolean-masking-with-pandas-b21b7714d0b6>

<https://www.jetbrains.com/help/pycharm/running-jupyter-notebook-cells.html>

<https://stackoverflow.com/questions/29803093/check-which-columns-in-dataframe-are-categorical>