# به نام خدا

## پروژ‌ه‌ی دوم درس داده کاوی

## استاد :

### دکتر رضا رمضانی

## دانشجویان :

### امیررضا صدیقین 953611133050

### امیر سرتیپی 953611133039



# 1-پیش‌پردازش داده ها

## توضیح کد و روند کار

کد مربوط به بخش پیش‌پردازش در فایل preprocess.py آمده است که توابع و کلاس نوشته شده در آن به شرح زیر است:

* کلاس SqlManager : برای استفاده‌ی راحتتر از دیتابیس نوشته شده است.
* تابع drop\_numerical\_outliers : یک دیتافریم را در ورودی می‌گیرد و با محاسبه‌ی مقادیر مربوط به روش IQR ستون‌های عددی دیتافریم را بررسی می‌کند و عناصر با داده‌های پرت را ازدیتافریم خارج ‌می‌کند، خروجی این تابع شامل یک تاپل که عنصر اول آن یک دیتافریم مخصوص عناصر پرت است و عنصر دیگر یک دیتافریم اصلاح‌شده فاقد آن عناصر است.
* تابع missing\_data : در این تابع اطلاعات مربوط به missing value های دیتافریم را استخراج می‌کند و میزان missing value ها و درصد آن‌ها در هر ستون را در خروجی ‌می‌دهد.
* تابع pre\_processing : تابع اصلی این فایل است که در ابتدا دیتاهای پیش‌پردازش نشده را در دیتابیس ذخیره می‌کند.(برای تحلیل بهتر و تصمیم گیری در پیش‌پردازش این امر بسیار مفید است.)، سپس دیتا‌های شامل علامت سوال را به null تبدیل می‌کند و سپس خروجی تابع missing\_data را در دیتابیس ذخیره ‌می‌کند. بعد از آن سطر‌هایی که دارای null هستند را حذف می‌کند و همچنین دو ستون capital\_gain و capital\_loss را با هم ادغام ‌می‌کند و در ستونی به نام capital قرار می‌دهد.(دلیل ادغام در ادامه ذکر شده است.) همچنین ستون fnlwgt را حذف کردیم زیرا اطلاعات مفیدی در بر نداشت، سپس خروجی تابع drop\_numerical\_outliers را گرفته و در دیتابیس ذخیره می‌کند. همچنین تابع describe که توضیحی از جمله چارک ها و خواصی از این قبیل نیز فراخوانی شده است و اطلاعات در دیتابیس ذخیره شده است .

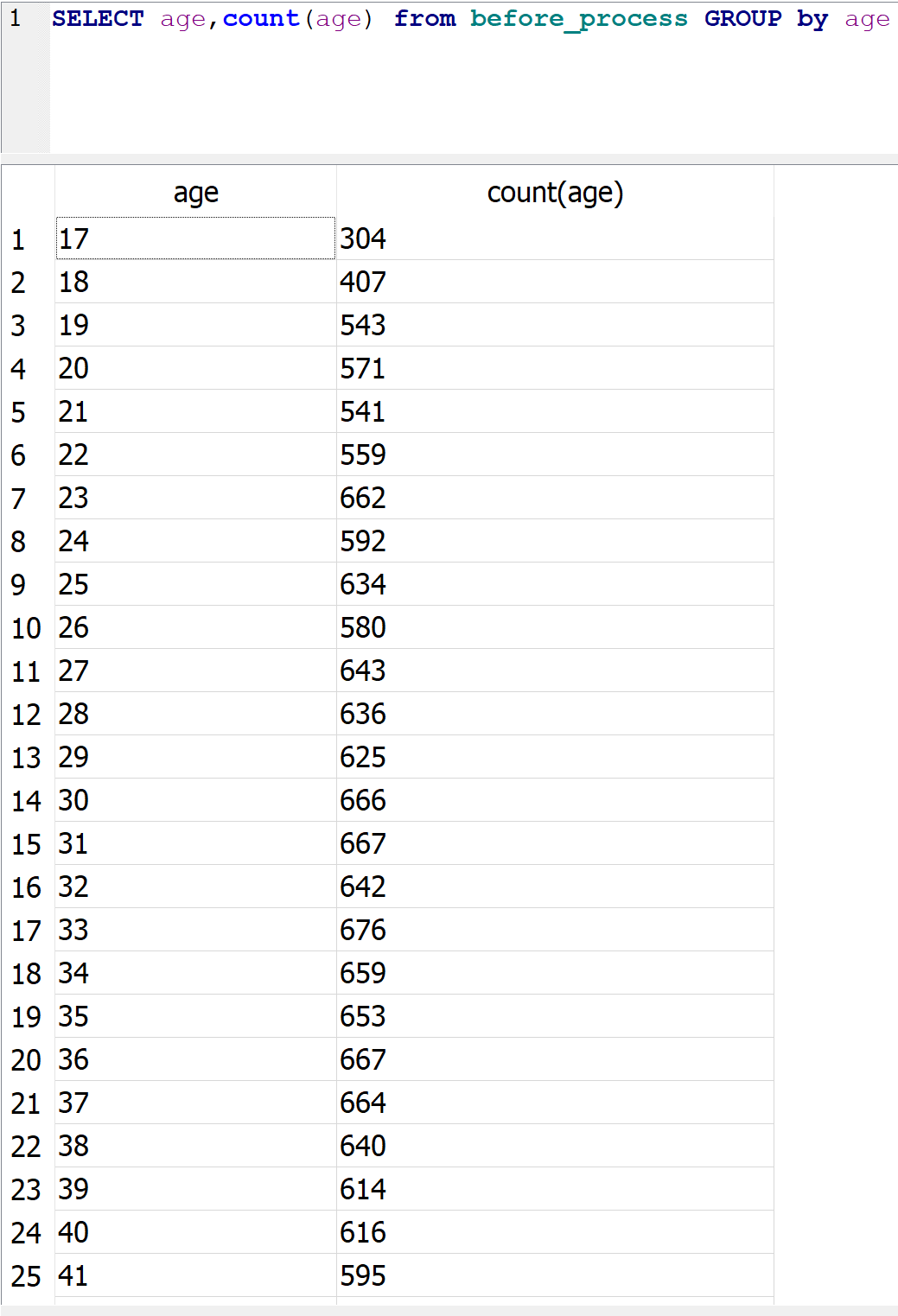
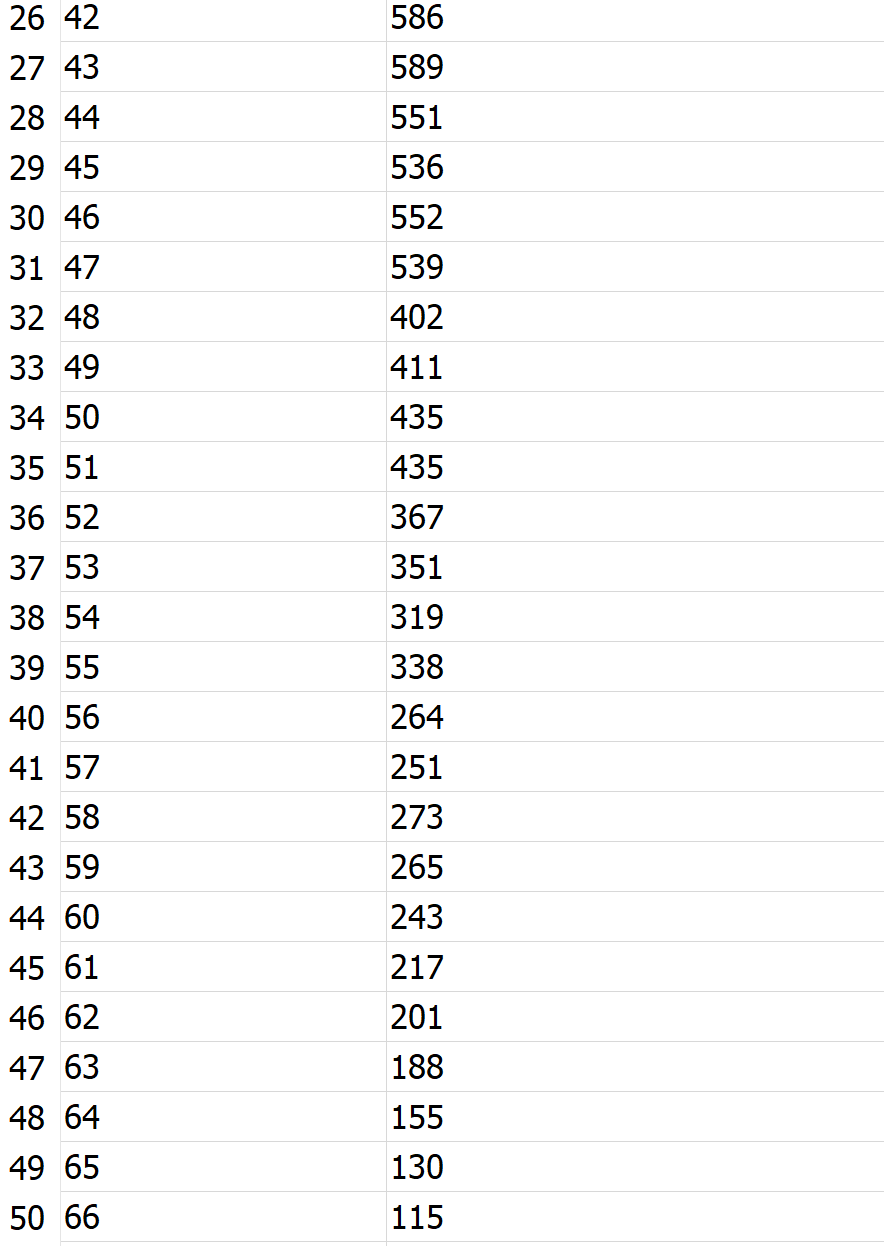
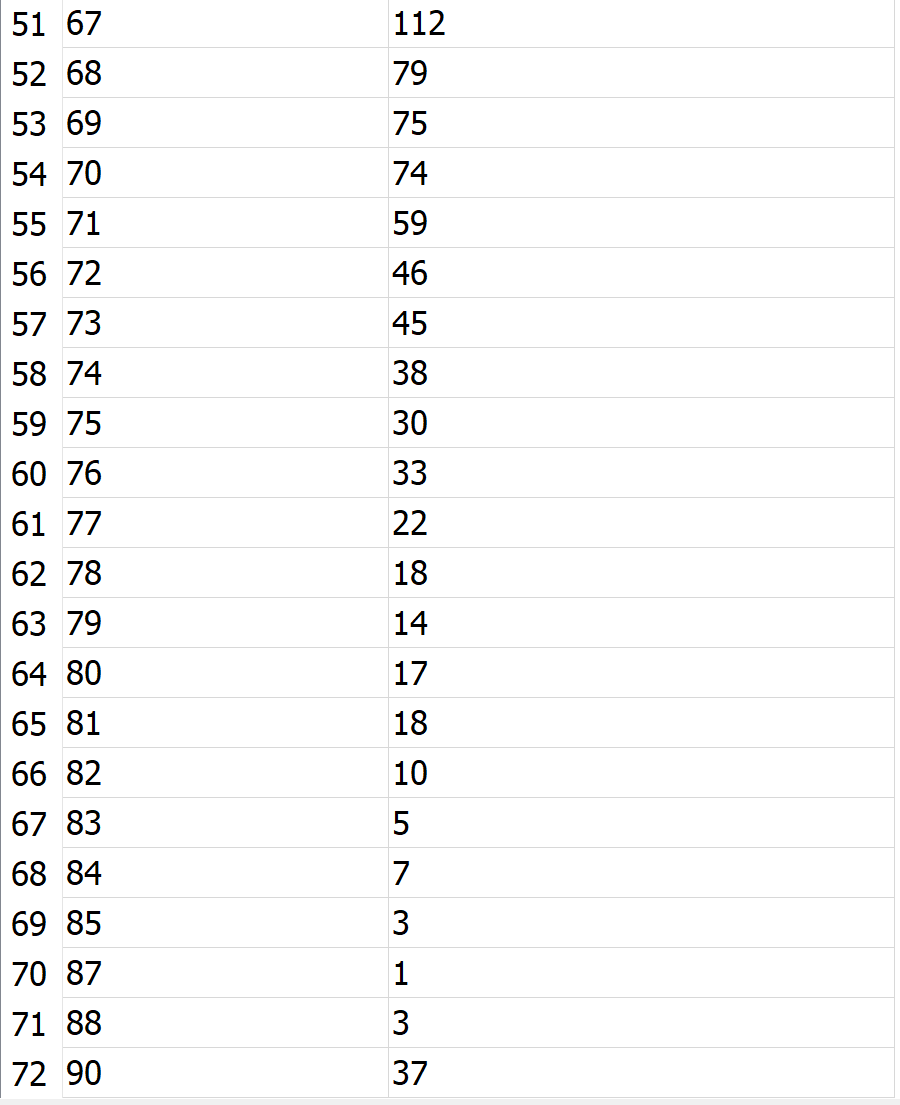
## خروجی

در این بخش خروجی شامل موارد زیر است :

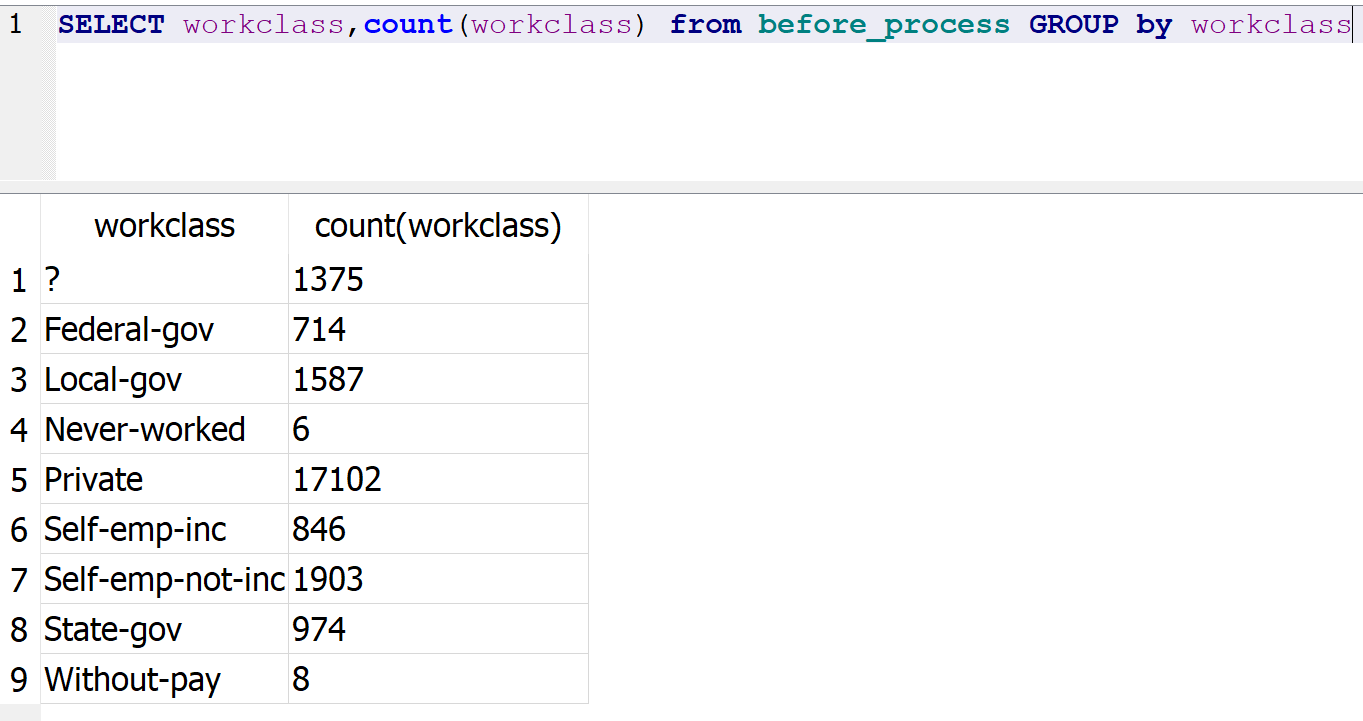
* دیتابیس information.sqlite : شامل اطلاعات ذخیره شده‌‌‌ی این بخش است .
  + جدول befor\_process : شامل دیتاها قبل پردازش
  + جدول describe : شامل اطلاعات ستون‌هایی از دیتاها مثل میانگین و تعداد و ... بعد از پیش‌پردازش
  + جدول information : شامل دیتاهای پیش‌پردازش شده.
  + جدول missing\_information : شامل اطلاعات مربوط به missing\_value هر ستون.
  + جدول outliers : عناصری که به عنوان عناصر outlier شناخته شده و از داده‌های اصلی پاک شده است.
* فایل dtypes.txt : که نوع داده‌های هر ستون از دیتافریم را ذخیره کرده است.

## بررسی داده‌های پیش‌پردازش نشده:

* اطلاعات مربوط به ستون age : که پراکندگی به شرح زیر است که البته در بخش حذف outlier ها داده های با سن خیلی بالا و داده‌های با سن پایین متناسب با معیار IQR حذف می‌شوند.



* اطلاعات مربوط به ستون : دراین ستون ما دارای مقادیر missing value است که حذف می‌شود.

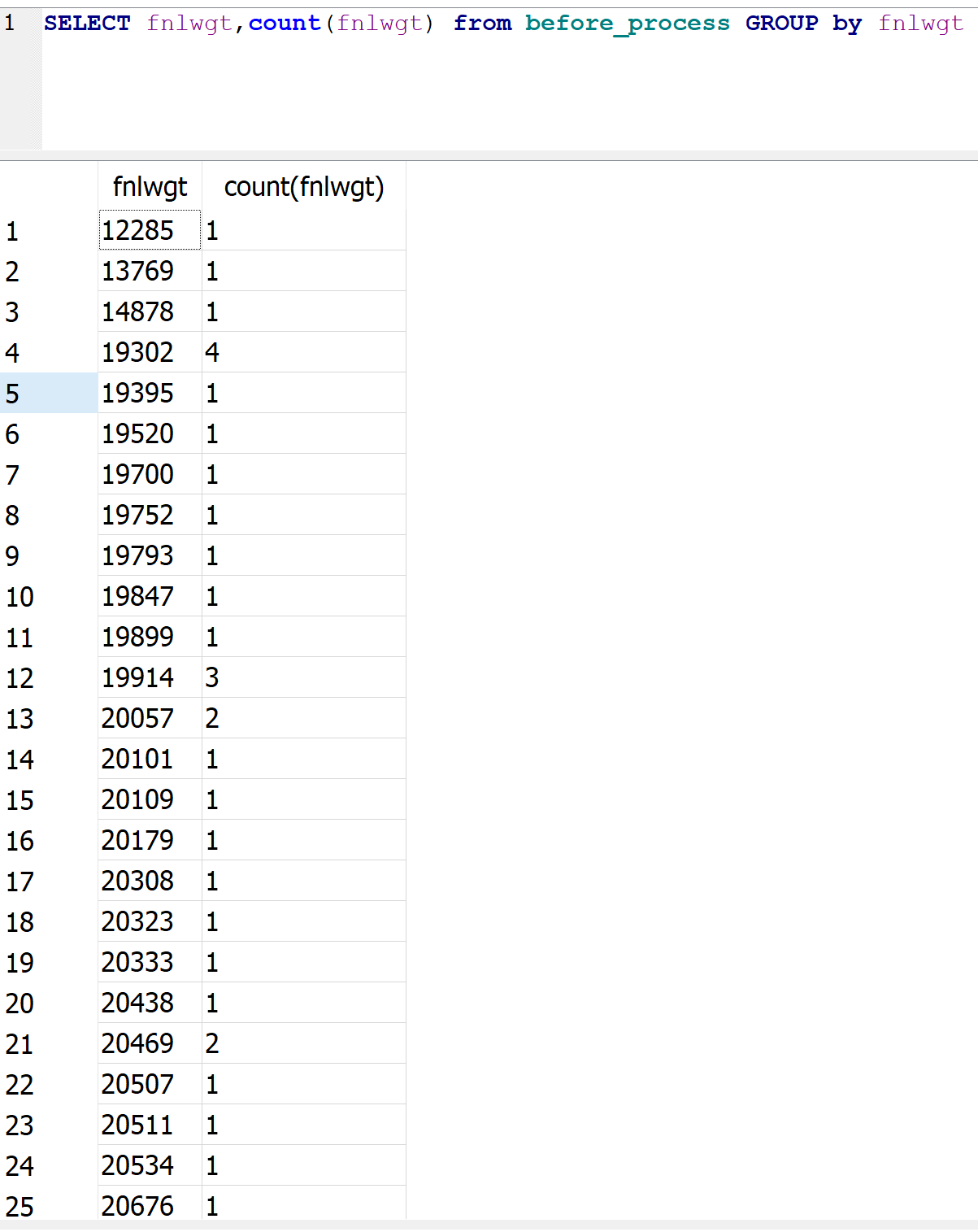


* اطلاعات مربوط به ستون fnlwgt:

این ستون دارای تنوع بسیار با میزان تکرار کم است.

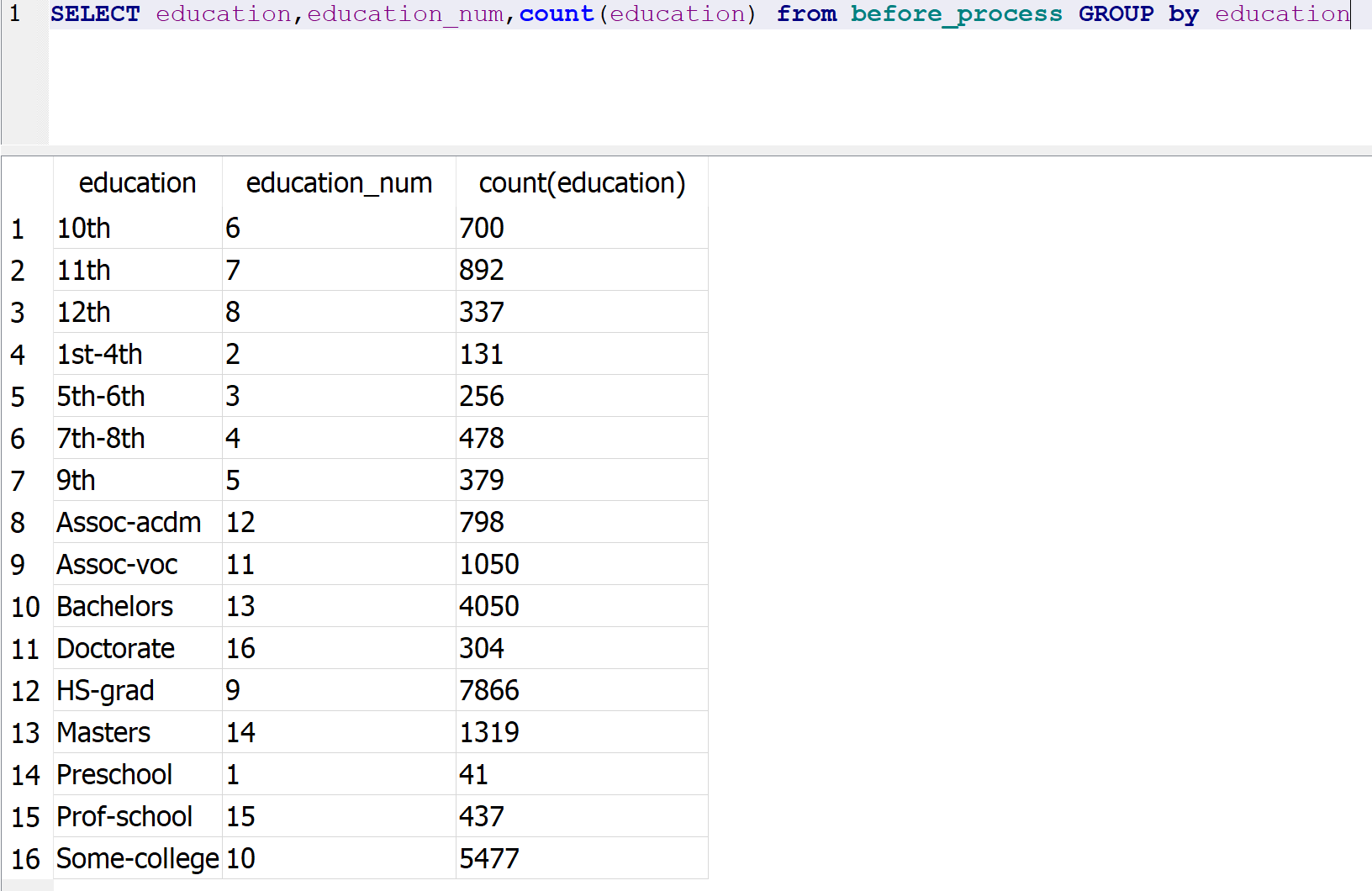
این ستون به دلیل نداشتن اطلاعات مفید و داشتن بار پردازشی زیاد حذف کردیم.

در زیر به دلیل زیاد بودن آن ها (17599 مقدار مختلف ) همه‌ی آن را نیاوردیم.



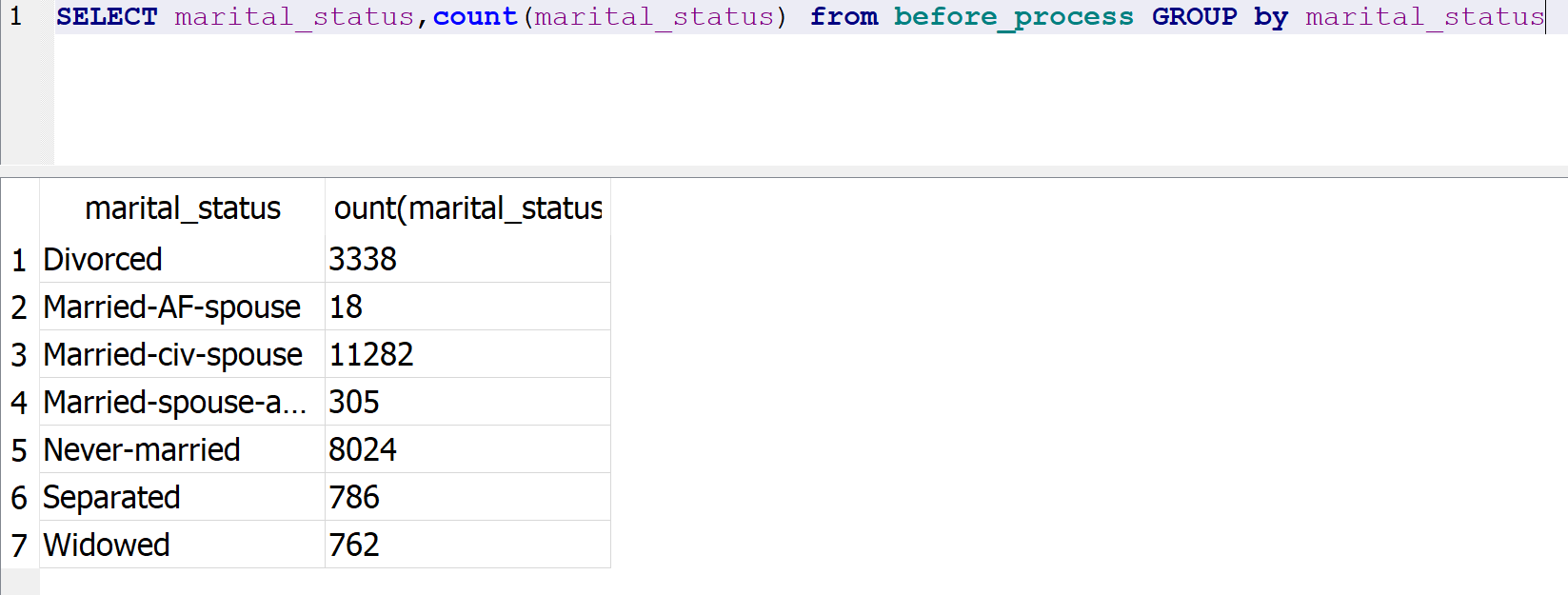
* اطلاعات مربوط به ستون سطح تحصیلات (education , education\_num)

به دلیل رابطه یک به یک education و education\_num که هر دو تقریبا یک مفهوم را می‌رساند education را حذف کردیم و education\_num را به دلیل عددی بودن و محاسبه‌ی بهتر نگه داشتیم.



* اطلاعات مربوط به ستون marital\_status

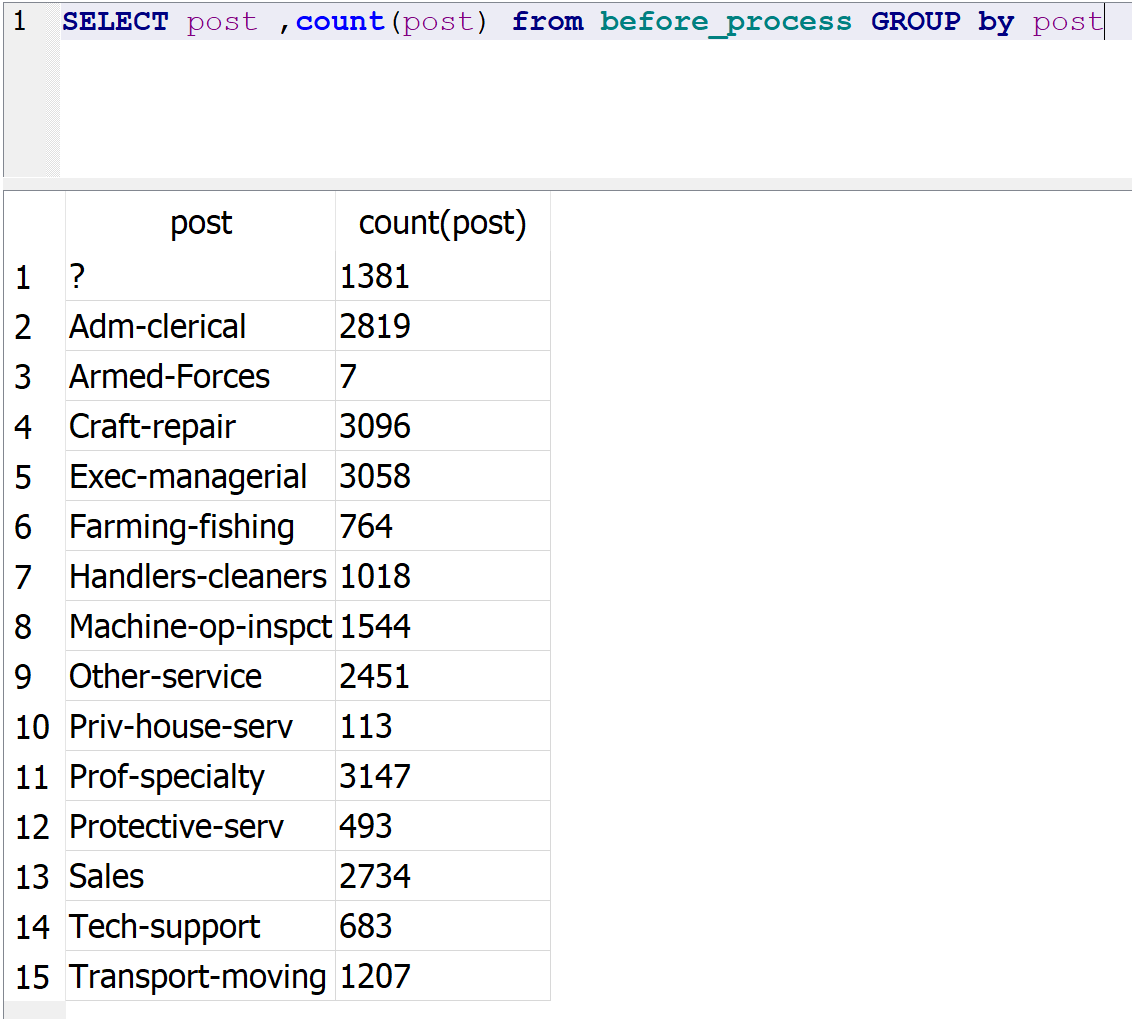
این ستون به طور عادلانه توزیع نشده است و برای مثال married\_af\_spouse به نسبت married\_civ\_spouseخیلی کم است.



* اطلاعات مربوط به ستون post

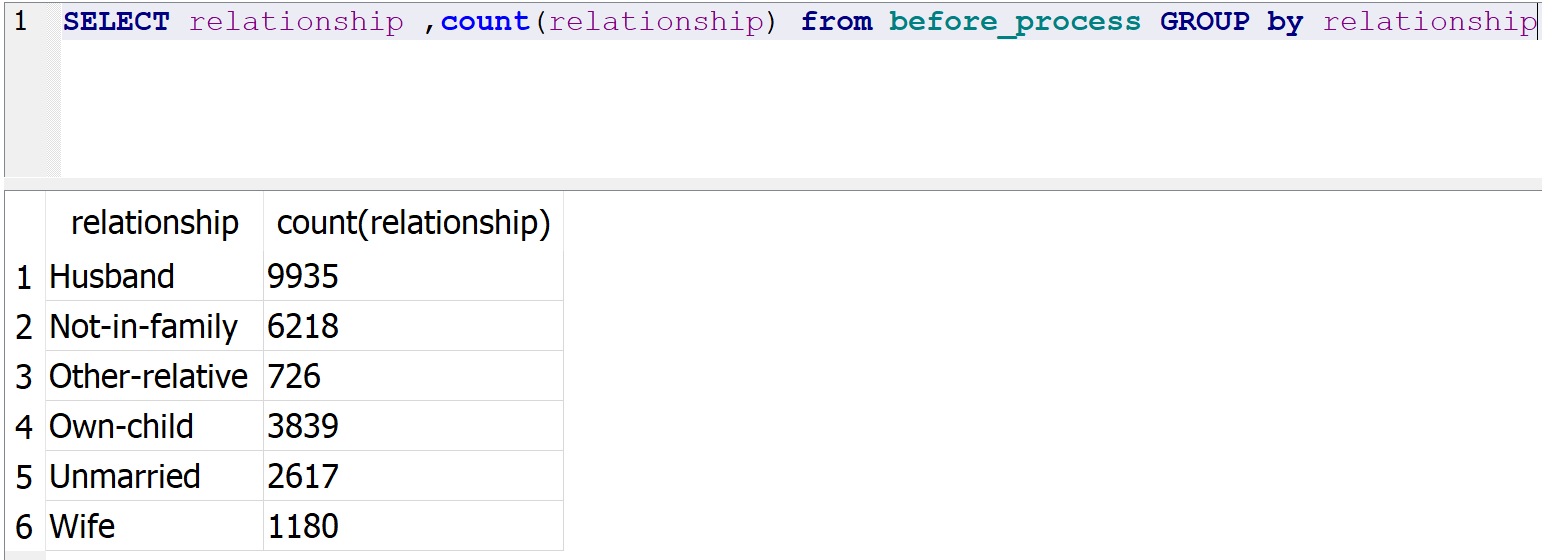
در این ستون نیز مقادیر missing value داریم.

توزیع خوب است ولی برای مثال Armed\_Forces خیلی کم است ولی در بقیه موارد به نسبت تعداد عادلانه بوده است.



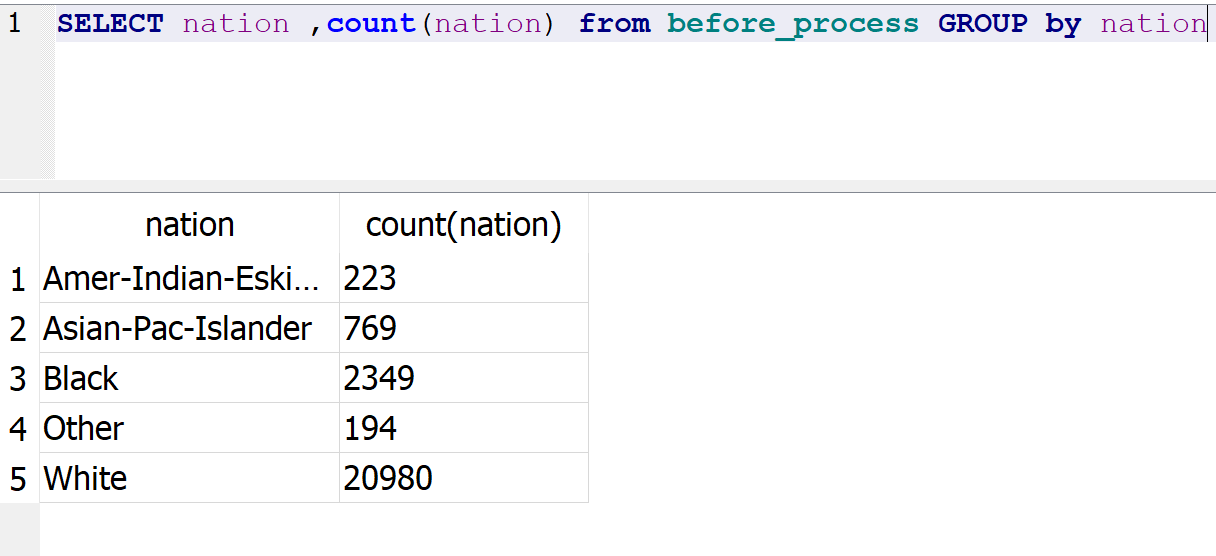
* اطلاعات مربوط به relationship

از نظر ما این ستون تا حدی همپوشانی با ستون‌های وضعیت تاهل و جنسیت دارد که البته پاکسازی آن هزینه بر است و به صرفه نیست.



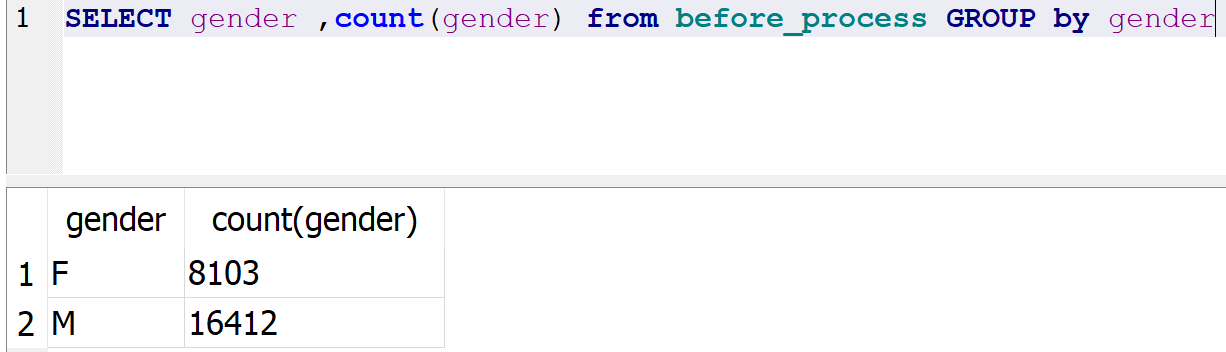
* اطلاعات مربوط به nation

این دیتاست بیشتر شامل افراد سفید پوست بوده است و برای پیش‌بینی افراد سفید پوست بهتر عمل می‌کند.



* اطلاعات مربوط به gender

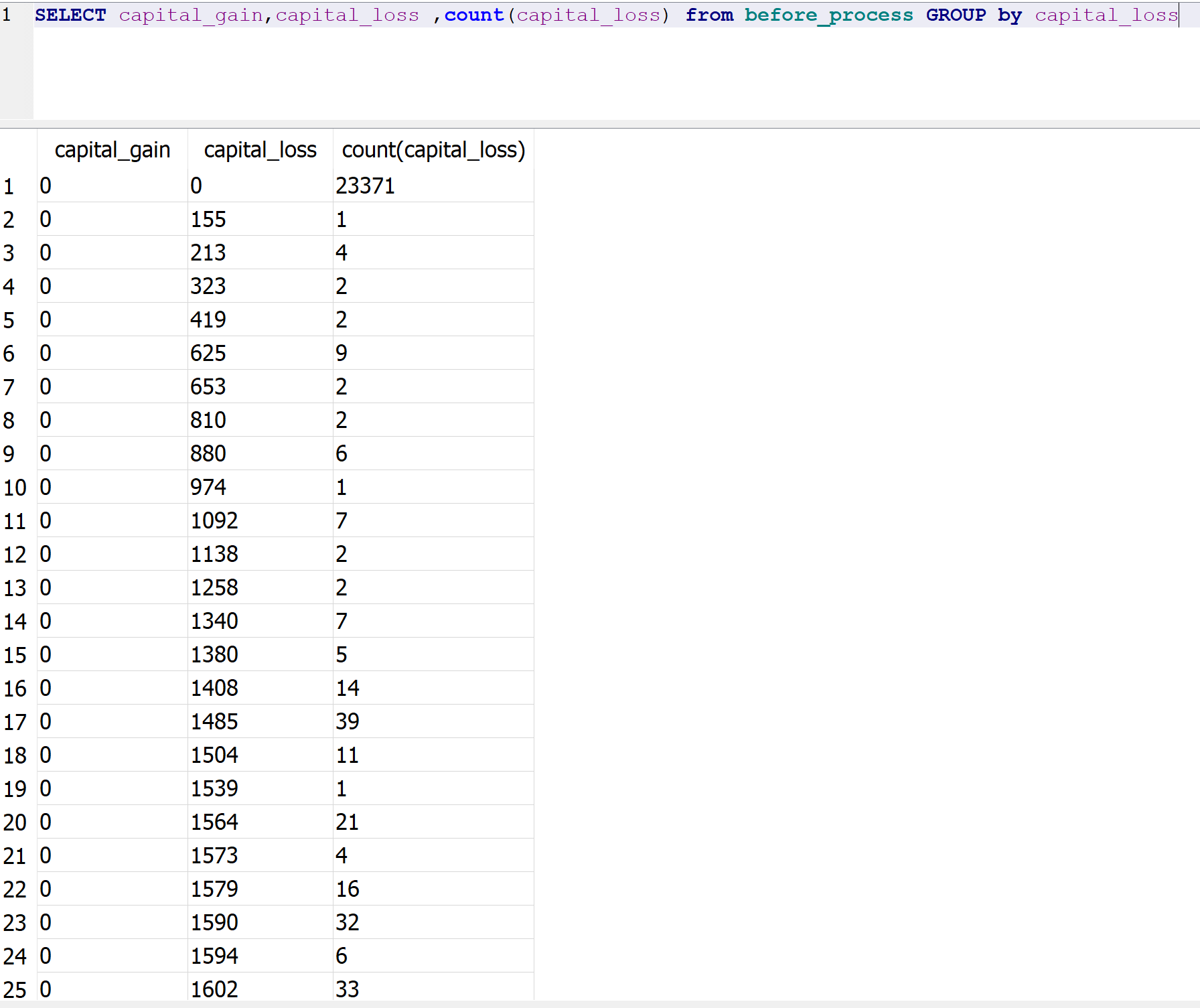
تعداد نمونه‌هایی که گرقته شده است مرد‌ها دوبرابر زن ها هستند و بهتر بود تعادل بیشتری وجود داشت.

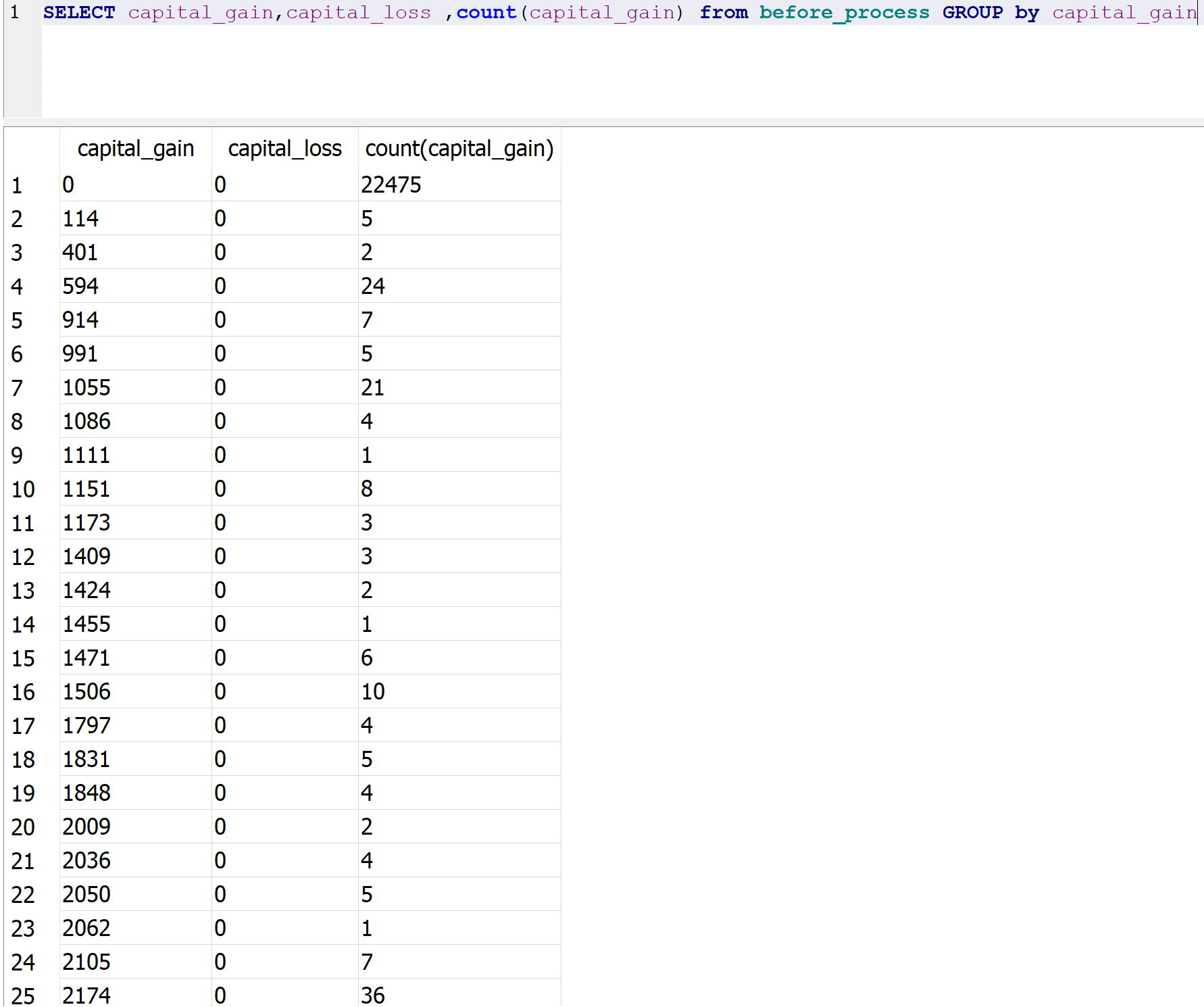


* اطلاعات مربوط به capital gain , capital loss

به دلیل زیاد بودن فقط تعدادی از آن‌ها را در این جا نمایش دادیم ولی می‌توانین برای مشاهده به دیتابیس مراجعه کنید.

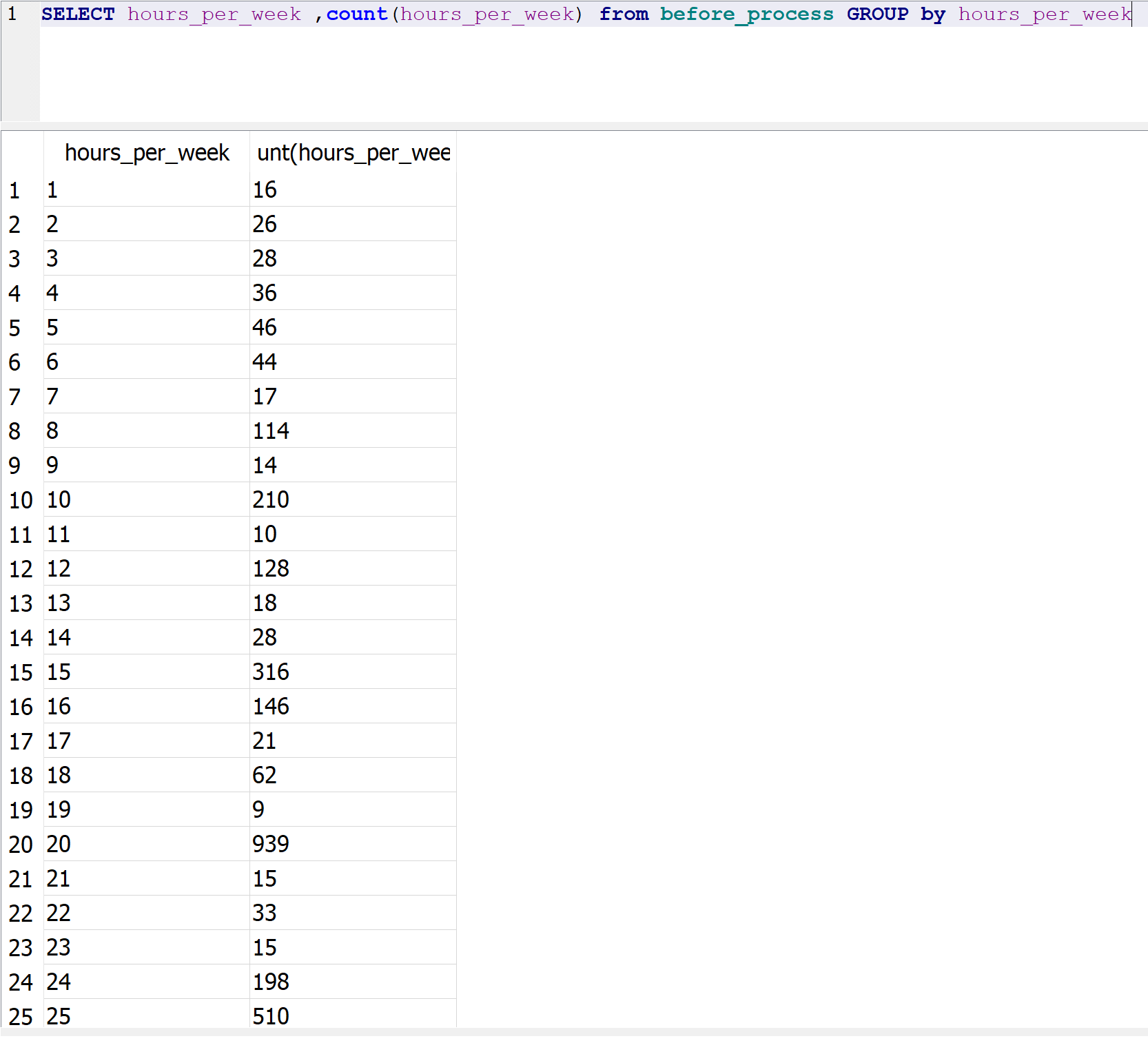
همانطور که نمایش داده شده است هر جا که capital\_gain عددی غیر از صفر است capital\_loss صفر است و برعکس برای همین و برای بهتر شدن نتیجه این تو ستون را ادغام کردیم به ازای هر کدام مقداری به نام capital که حاصل تفریق loss از gain است محاسبه کرده و در ستون capital قرار دادیم.

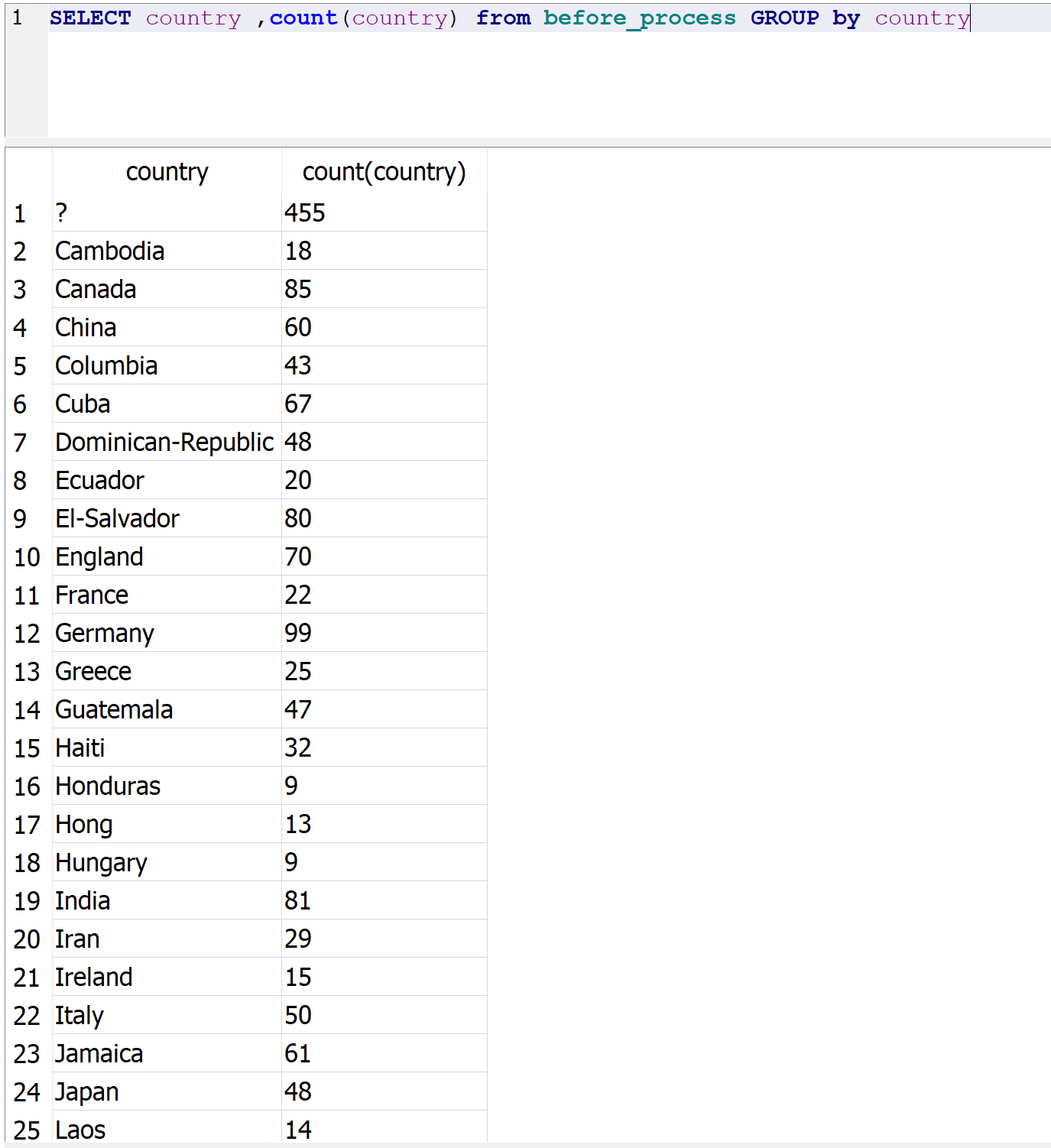
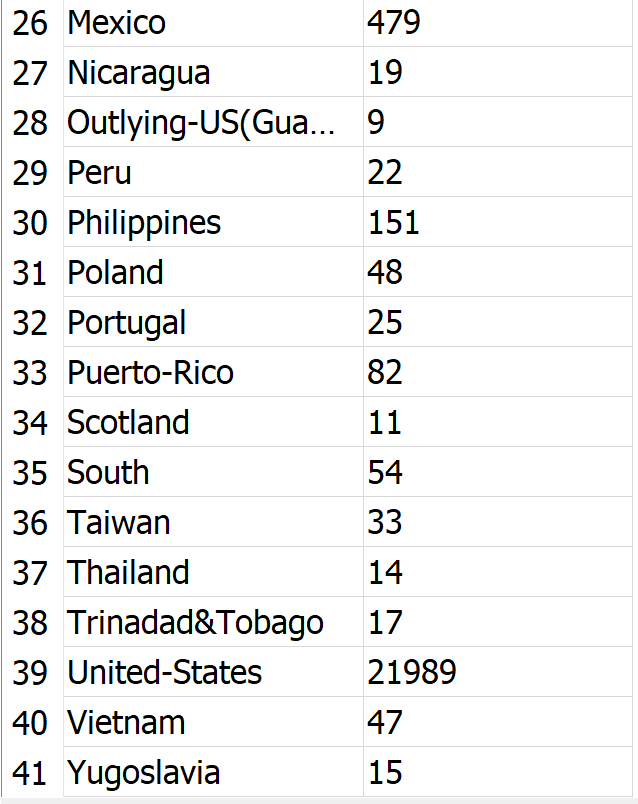




* اطلاعات مربوط به hours per week

دارای 91 مقدار مختلف است که مقادیری از آن در زیر نمایش داده شده است. پراکندگی به نسبت خوبی دارد .



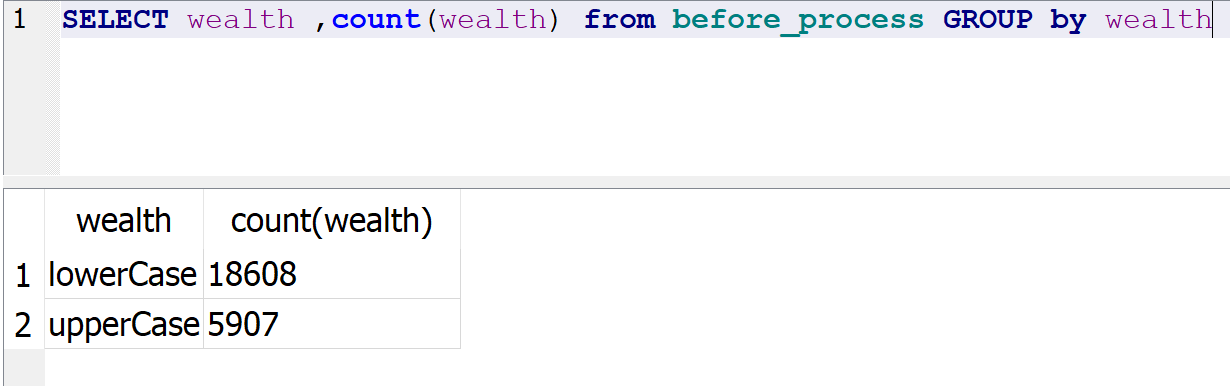
* اطلاعات مربوط به hours per week
* 

داده ها از 40 کشور مختلف جمع آوری شده است که البته این ستون دارای بسیاری missing value است و حذف می‌شود.

* اطلاعات مربوط به wealth

این ستون همون ستون برچسب‌های ما هست که بر اساس آن آموزش داده ‌می‌شود.

مشکلی که در دیتاها است داده ها به سمت lowercase میل دارند و تعادل در فراوانی آن ها نیست .



## موارد تکمیلی:

ما از دیتابیس استفاده کردیم برای این‌که داده‌ها به صورت منظم می‌شود با چشم مورد بررسی قرار داد و بهترین تصمیمات را برای پیش‌پردازش و همچنین دسته بندی، انتخاب کنیم و با زدن کوئری‌های مختلف اطلاعاتی همچون روابط بین ویژگی‌ها، پراکندگی‌ها، کارهای محاسباتی، میزان تاثیر هر ویژگی بر یرچسبی که خورده است و ... را با نگاه نیز تشخیص داد و به آسانی و راحت موارد مختلف و اطلاعات مهم دیگر را نیز ذخیره کرد. همچنین ذخیره سازی این موارد در آینده برای بازیابی اطلاعات بسیار مفید خواهد بود و صرفه جویی در وقت است.

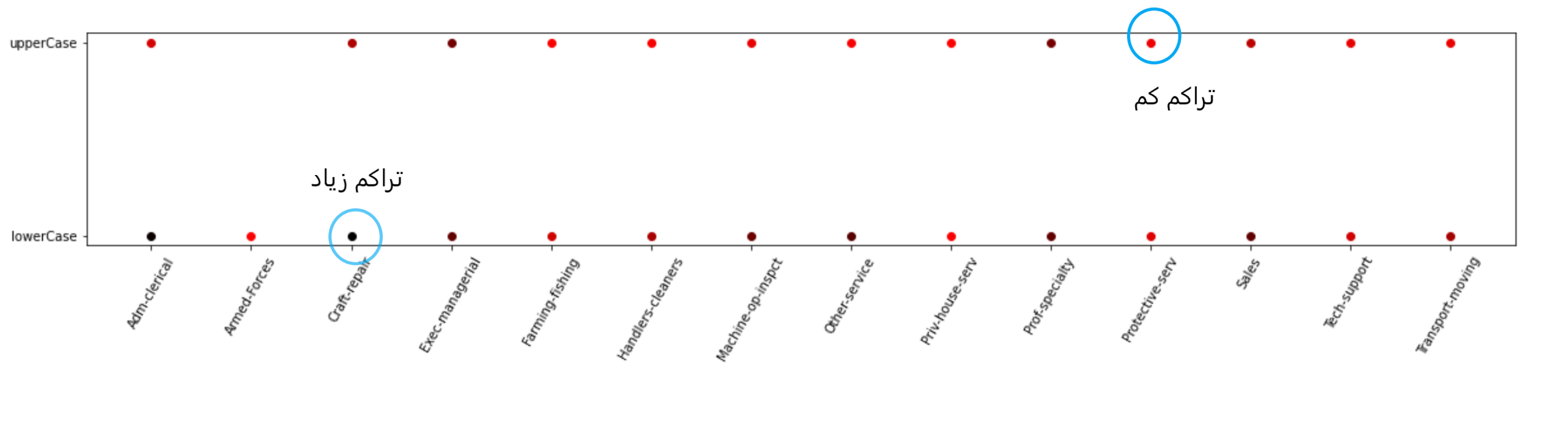
# 2-تاثیر هر ویژگی بر روی wealth

## نمودارهای پراکندگی :

کد مربوط به این بخش در فایل distribution.py آمده است که شامل تابع make\_plot است که پراکندگی نقاط در ویژگی‌های مختلف را با شدن رنگ نشان داده شده است (هر چه تیره تر frequency بیشتر).

خروجی این بخش در پوشه‌ی distribution\_plots وجود دارد و برای هر ستون یک عکس با نام خودش در نظر گرقته شده است.

برای مثال، تصویر زیر که تاثیر ویژگی post به wealth را نمایش می‌دهد.



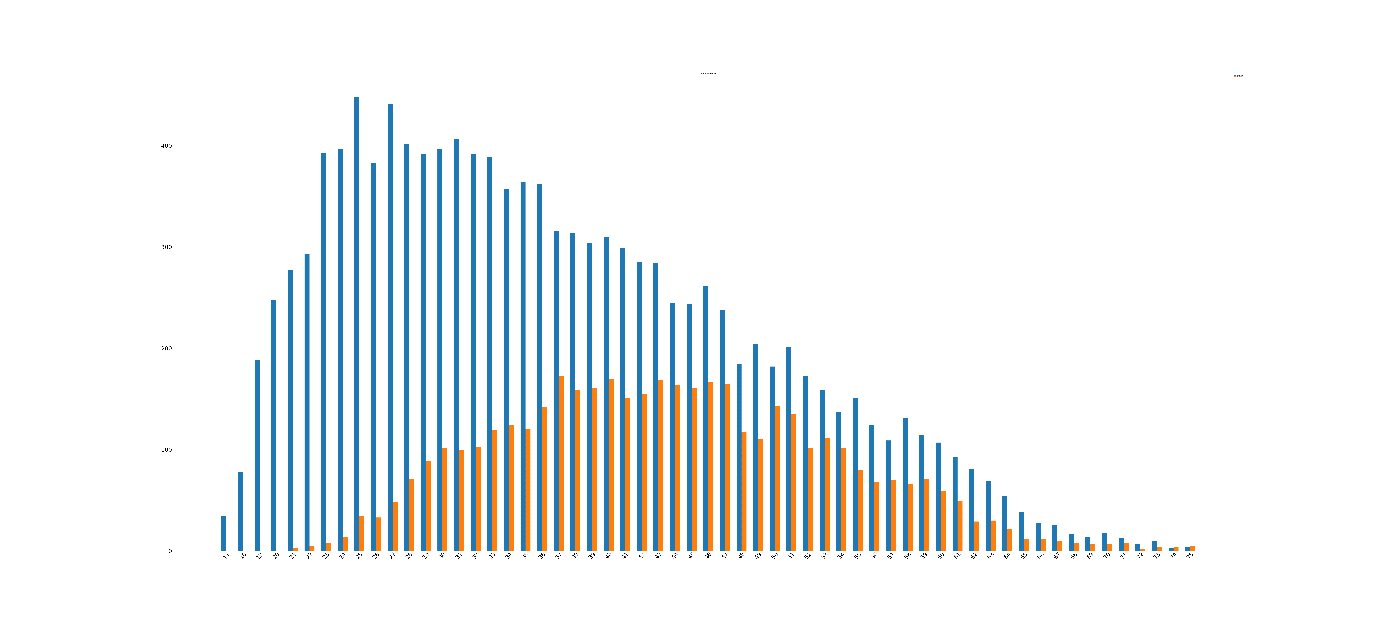
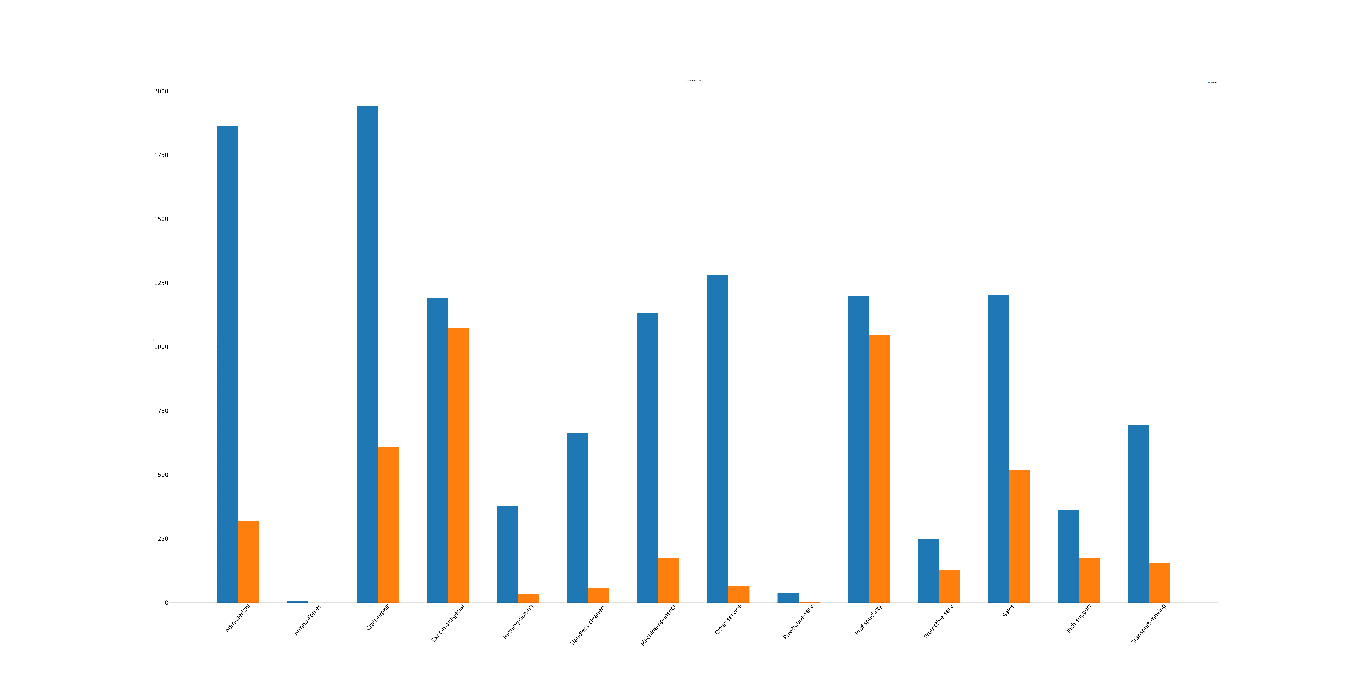
## نمودارهای هیستوگرام

در فایل histogram\_plots دومدل plot برای ویژگی‌ها ترسیم می‌شود که رنگ آبی به معنای در آمد پایین و رنگ نارنجی به معنای درآمد بالا می‌باشد.

که اگر زوم صورت گیرد در گوشه راست بالا مشخص است.

در دو حالت plot ها را به دست آوردیم.

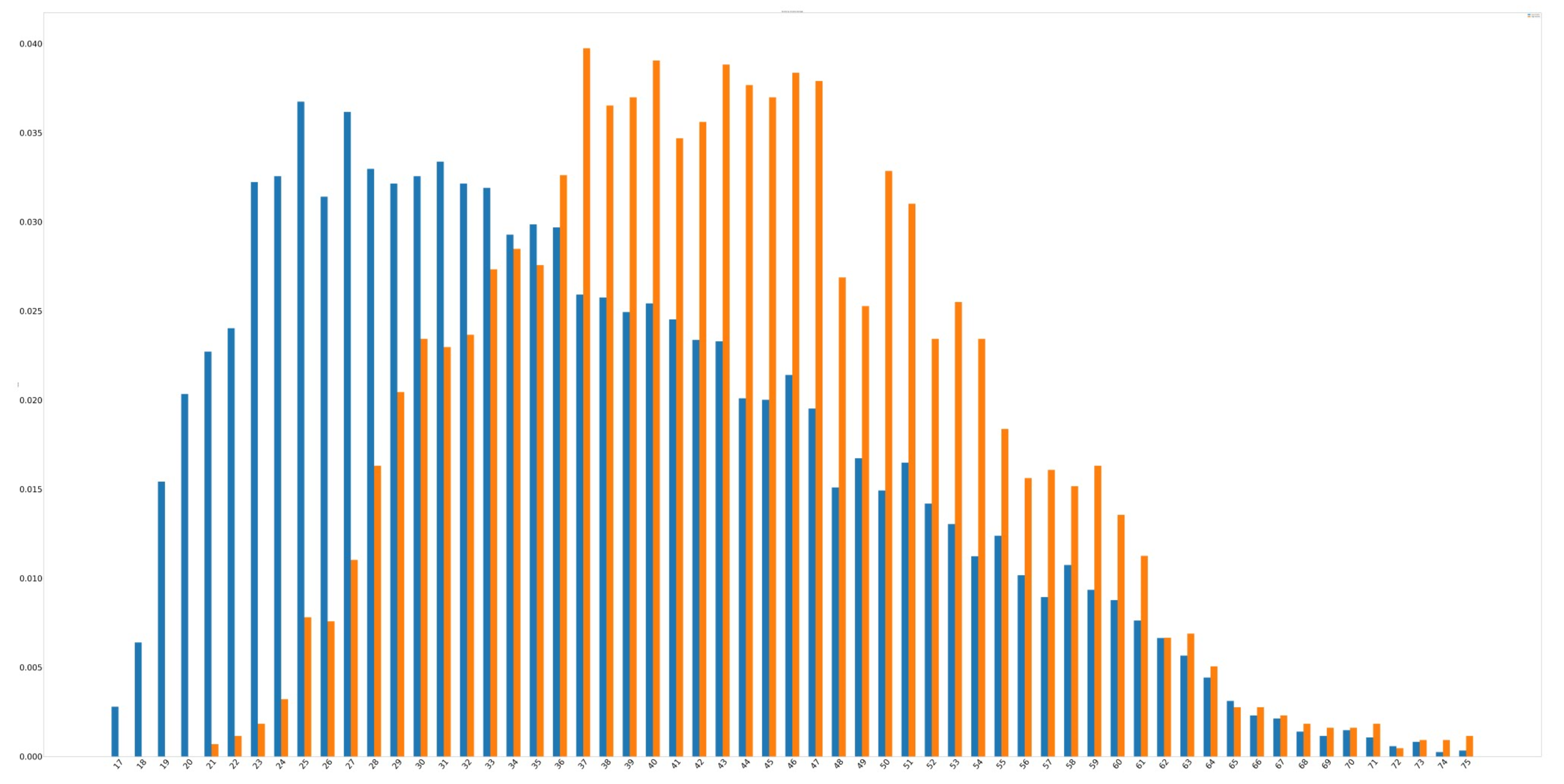
1. Plot براساس تعداد : طبق خواسته سوال پس از انجام مرحله پیش پردازش داده‌ها برای هرویژگی نمودار آن کشیده و در پوشه outs/plots قابل مشاهده است که با کوئری بر روی دیتابیس و بدست آوردن تعداد بر حسب مقدار هر ویژگی نمودار آن ترسیم شده است و برای ویژگی‌هایی مانند سود سرمایه چون قابل ترسیم نبود با عمل بازه بندی سودها باعث خوانایی بیشتر آن شده است.(تمام plot ها با اندازه‌ی مناسب در پوشه‌ی outs/plots قرار داده شده است.)

1. Plot بر اساس درصد:

این نمودار براساس درصد ترسیم شده است که عمل مقایسه را به شکل بهتر و راحت تر میسر می‌کند و در پوشه outs/ plots\_percent قرار گرفته است.(درصد برحسب تعداد کل uppercase ها و درصد برحسب تعداد کل lowercase )

تمام plot ها مربوط به این بخش با اندازه‌ی مناسب در پوشه‌ی outs/plots\_percent قرار داده شده است.



# 

نکته‌ی اضافی در این بخش این بود که در قسمت capital تنوع داده‌ها خیلی زیاد بود و با استفاده از گروه‌بندی کردن قرار داده شده است.

# درخت تصمیم

## توضیح کد و روند کار

در این فایل الگوریتم decision tree پیاده سازی شده است که باید براساس ویژگی‌های دیتا ست تصمیم گیری صورت بگیرد که می‌توان با دادن نام این ویژگی در قالب یک ارایه این عمل را انجام داد.

برای این کار لازم است تا مقدار های ویژگی‌ها به عدد تبدیل شود تا برای الگوریتم قابل شناسایی باشد که به این کار اصطلاحا label encode گفته می‌شود وسپس هر ویژگی که encode شد لازم است تا هر سطر داده در پایگاه داده به صورت یک تاپل در بیاید و درنهایت تمامی ویژگی ها باهم ترکیب بشوند.

درنهایت از دیتا اماده شده 30% دیتا‌ها را به صورت رندوم به عنوان دیتای تست جدا کرده و با بقیه ان مدلمان را آموزش می‌دهیم. (به وسیله‌ی تابع decision\_tree )

سپس دیتای تست را برچسبش را بر حسب مدلمان پیشبینی می‌کنیم تا در خروجی بر روی دیتای تست ما برچسب بزند.(به وسیله‌ی تابع clf.predict)

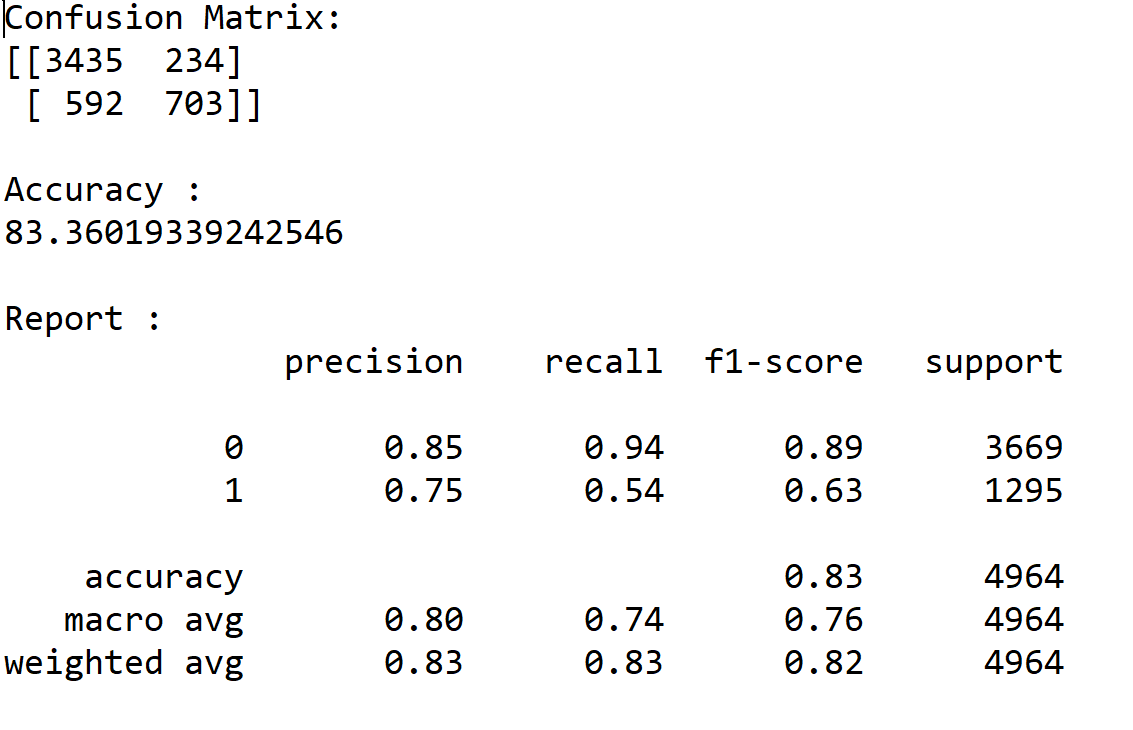
سپس برای مشاهده نتیجه کار تابع cal\_accuracy را صدا می کنیم که در این تابع متریک‌های زیر محاسبه می‌شوند.

* Confusion Matrix
* Accuracy
* Precision
* Recall
* f1-score
* accuracy macro
* accuracy weighted

که در فایل accuracy\_decision\_tree\_report.txt قابل مشاهده است.

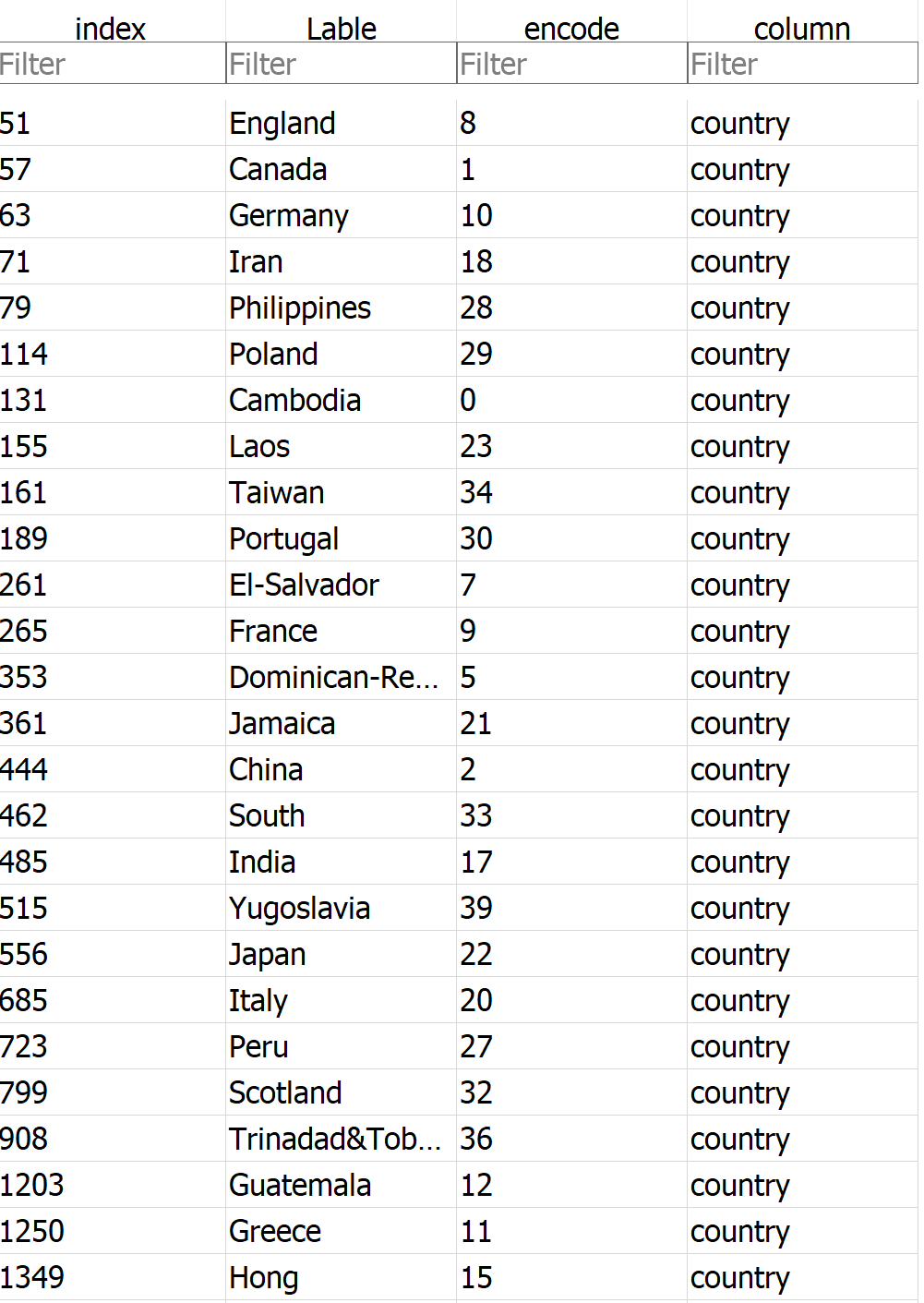
به عنوان خروجی فایلpdf که تصویر درخت تصمیم نیز ترسیم شده است که بر اساس آن ابتدا رابطه و سپس سود سرمایه بیشترین gain را داشته اند. که در پوشه outs با نام decision\_tree.pdf قرار دارد.

همچنین فایل accuracy\_decision\_tree\_report.txt مشخصات مربوط به دقت و محاسبات لازم را نشان می‌دهد.



البته در بالا 0 به عنوان lowercase در نظر گرفته شده است و 1 به عنوان uppercase.

همچنین در دیتابیس information.sqlite جدولی تحت عنوان encoding\_guide درست می‌شود که در آن می‌گوید هر اتربیوت به چه عددی encodeشده است. برای مثال شکل زیر بخشی از داده‌های مربوط به به encode شدن ویژگی کشور را نمایش می‌دهد.



Naïve.py

در این فایل الگوریتم Naïve پیاده سازی شده است که باید براساس احتمالات شرطی تصمیم گیری را انجام می‌دهد

برای این کار لازم است تا مقدار های ویژگی‌ها به عدد تبدیل شود تا برای الگوریتم قابل شناسایی باشد که به این کار اصطلاحا label encode گفته می‌شود وسپس هر ویژگی که encode شد لازم است تا هر سطر داده در پایگاه داده به صورت یک تاپل در بیاید و درنهایت تمامی ویژگی ها باهم ترکیب بشوند.

درنهایت از دیتا اماده شده 30% را به عنوان دیتای تست جدا کرده و با بقیه ان مدلمان را آموزش می‌دهیم.

    # Create a Gaussian Classifier

    gnb = GaussianNB()

    # Train the model using the training sets

    gnb.fit(X\_train, y\_train)

    # Predict the response for test dataset

    y\_pred\_gaussian = gnb.predict(X\_test)

سپس برای مشاهده نتیجه کار تابع

cal\_accuracy(y\_test, y\_pred\_gini)

را صدا می کنیم که در این تابع متریک‌های زیر محاسبه می‌شوند.

* Confusion Matrix
* Accuracy
* Precision
* Recall
* f1-score
* accuracy macro
* accuracy weighted

که در فایل accuracy\_ Naïve\_report.txt قابل مشاهده است.

سوال 2

در خصوص تاثیر هر ویژگی بر تصمیم گیری بر اساس نمودارها

workclass\_\_ income:

بیشتر افراد دارای شغل private هستند که بیشتر افراد دارای این شغل هستند از نظر تعدادی امار افراد کم در امد در این شغل بیشتر و همچنین بیشترین افراد پر درامد نیز از همین دسته هستند.

relationship\_\_ income:

age\_\_ income:

بیانگر این است که در میان سالی بیشترین دامد حاصل می‌شود و هرچه سن بالاتر می‌رود افراد کم درآمد کمتر شده اند و در دوران جوانی افراد دارای درآمد کم بسیار زیاد است.

capital\_\_ income

افرادی که از سود سرمایه بیشتر از 14892 دارند تما در امد بالا دارند و اکثر جامعه در 0 تا 1294قرار دارند که خوب افراد بیشتری درامد کم دارند تا بیشتر.

country\_\_ income  
بیشتر دیتاهای ما در uniated state قرار داشت که تعدا افراد کم درامد زیاد ولی به نسبت افراد پر در امد بیشتری نیز در آنجا قرار دارند و مثلا کشور مکزیک افراد با درامد بالا را در حد خیلی کمی دارد.

education\_num\_\_ income:

به سادگی نشان می‌دهد که مدرک گرایی در همه جا جواب می‌دهد و هرچه گرید تحصیلی افراد بالاتر رفته درامده ها آنها هم زیاد بوده است.

gender\_\_ income:

در مجموع تعداد مردها در دیتاست تقریبا 3 برابر زن‌ها بود که نمودار آن بیان کننده این است که مردها از نظر دامدی وضعیت بهتری دارند.

hours\_per\_week\_\_ income

در این نمودار با 40 ساعت کار در هفته بیشترین تعدا درامد کم و همچنین افرادی با در امد بالا حضور دارند . و تقریبا می‌توان گفت افزایش ساعت کار باعث افزایش در آمد نمی شود.

marital\_status\_\_ income

نشان می‌دهد ازدواج چیز خوبیه و باعث افزایش درامد می‌شود.

و افراد مجرد نیز یا پول ندارند ازدواج کنند یا چون مجردن کم در آمد هستنند .

nation\_\_ income

نشان دهنده این است که افراد سفید پوست بیشتر ثروتمند هستند و به همان نسبت کم در امد نیز دارند

post\_\_ income

نشان دهنده این است ک اکثر افرادی که شغل‌هایی مانند exec manger یا prof دارند درامد خوبی دارند و شغل های که کندل بلند ابی بلند دارند و کندل نارنجی کوتاه است، نشان از در امد کم آن شغل هاست.

workclass\_\_ income

نتیجه گیری زیادی از آن نمی‌توان کرد و تقریبا در هر مورد افراد پر درامد و کم در امد و جود دارد.

relationship\_\_ income

کلا افراد با ازدواج کرده درامد بهتری نسبت به بقیه دارند.

سوال 5

در تحلیل کلی می‌توان گفت که بیشترین تاثیر از ویژگی‌های capital و relation گرفته شده و افراد با سود سرمایه و ازدواج کرده دارای برچسب با در امد زیاد هستند.