نام خدا

تمرین سوم درس هوش مصنوعی

سوال اول

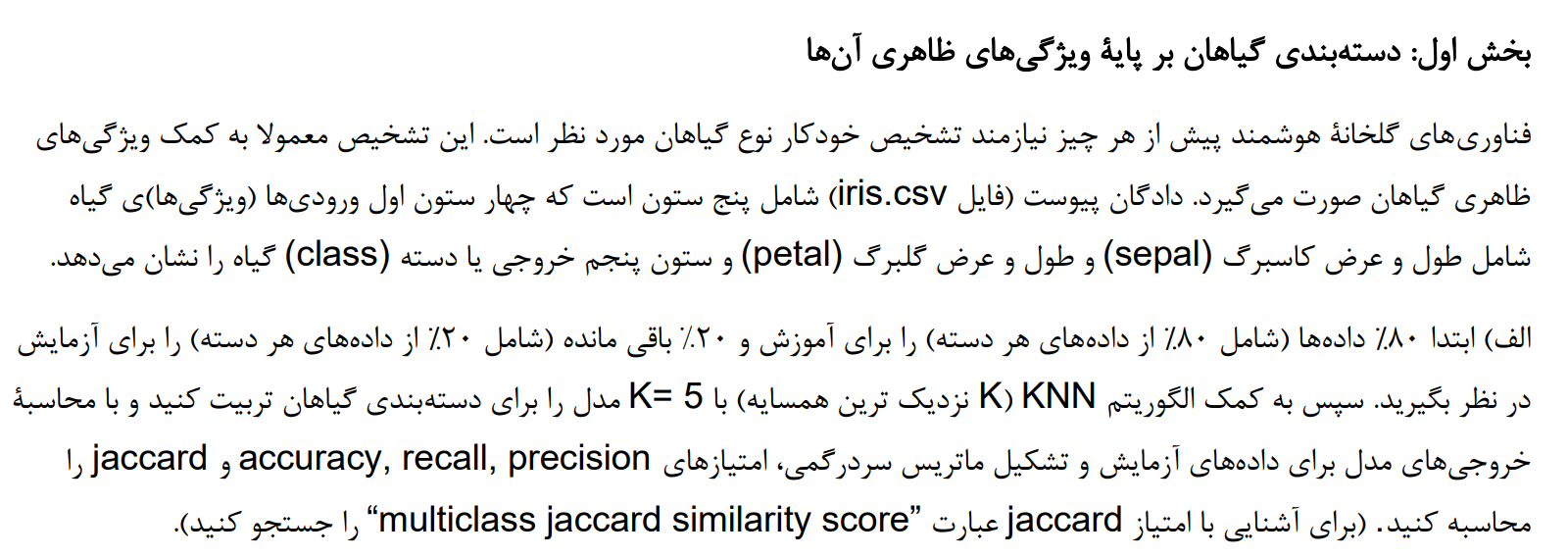
محمد امین سلطانی چم حیدری

810601081

استاد مربوطه:

دکتر مسعود شریعت پناهی

بهار1402





به منظور تربیت مدل برای این سوال ابتدا کتابخانه های مورد نظر در Jupyter notebook فراخوانی شده اند.

matplotlib 🡨 نمایش data و رسم نمودار

pandas 🡨 کار با داده های ساختار یافته مانند جداول،فایل csv و excel و...

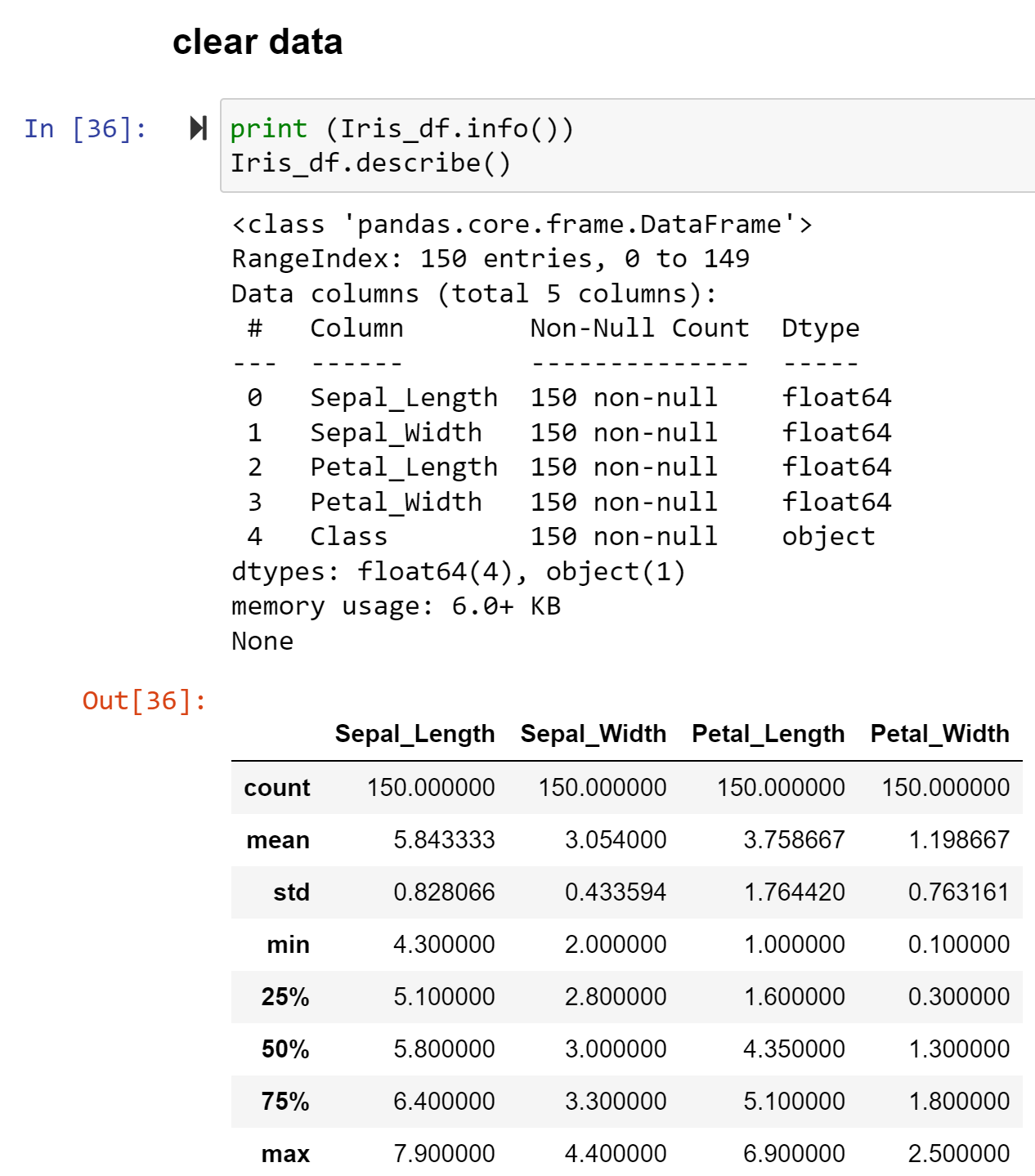
Numpy 🡨 انجام عملیات ریاضی،آماری و عددی

Pylab 🡨 تجسم داده های علمی و عملی.ترکیبی از matplotlib و numpy می باشد

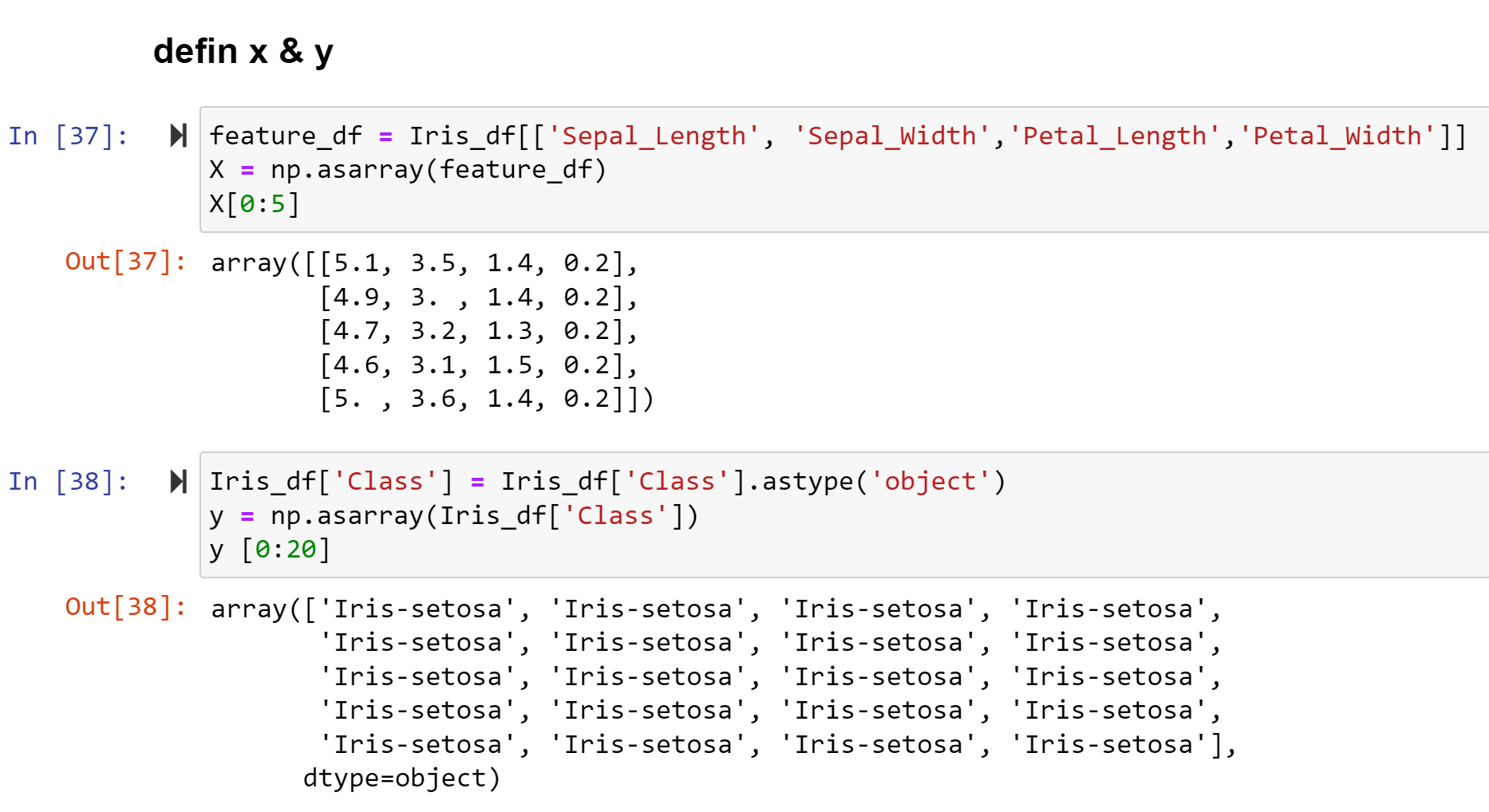
Sickit\_learn 🡨 یادگیری ماشین و تحلیل داده



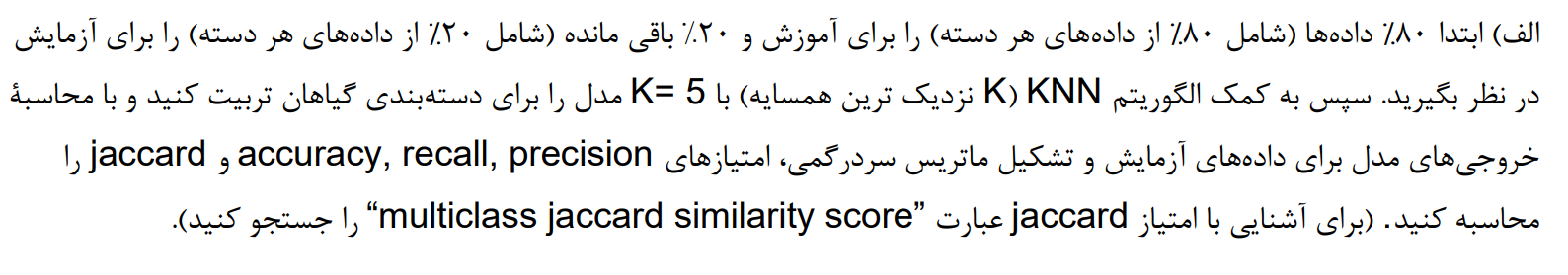
در مرحله بعد داده های مربوط به گیاه شامل ویژگی های ورودی و کلاس خروجی به کمک دستور pd از کتابخانه pandas باز شده اند.

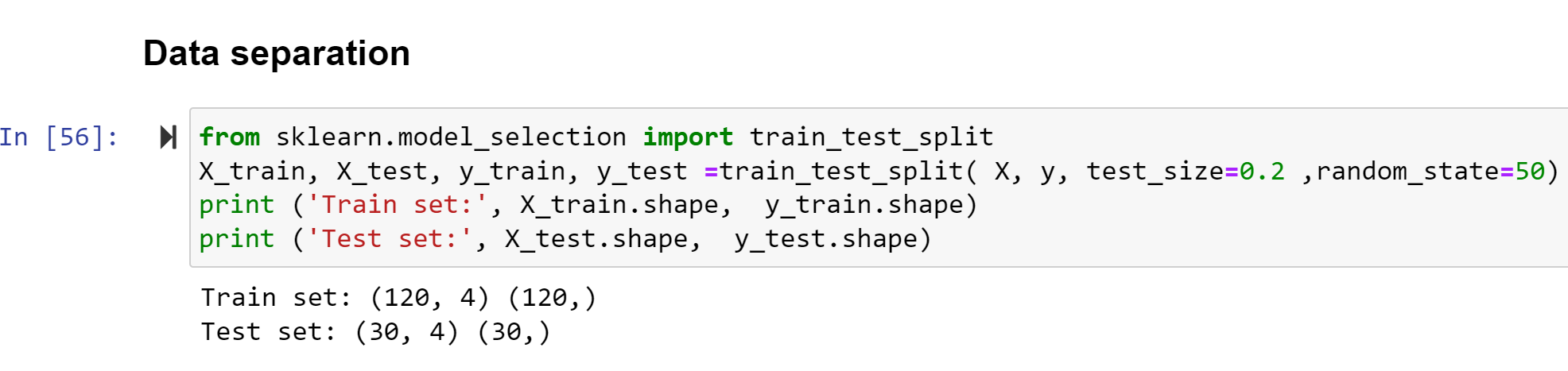


در قسمت بعد توصیفی از داده های ورودی مشاهده می شود.این مرحله به ان علت انجام شده که در صورت وجود هرگونه نقص و عیب در داده های ورودی سطر و ستون مربوط به ان داده ورودی حذف شود که البته در این سوال مشاهده می شود هیچ گونه نقصی در داده های ورودی وجود ندارد.



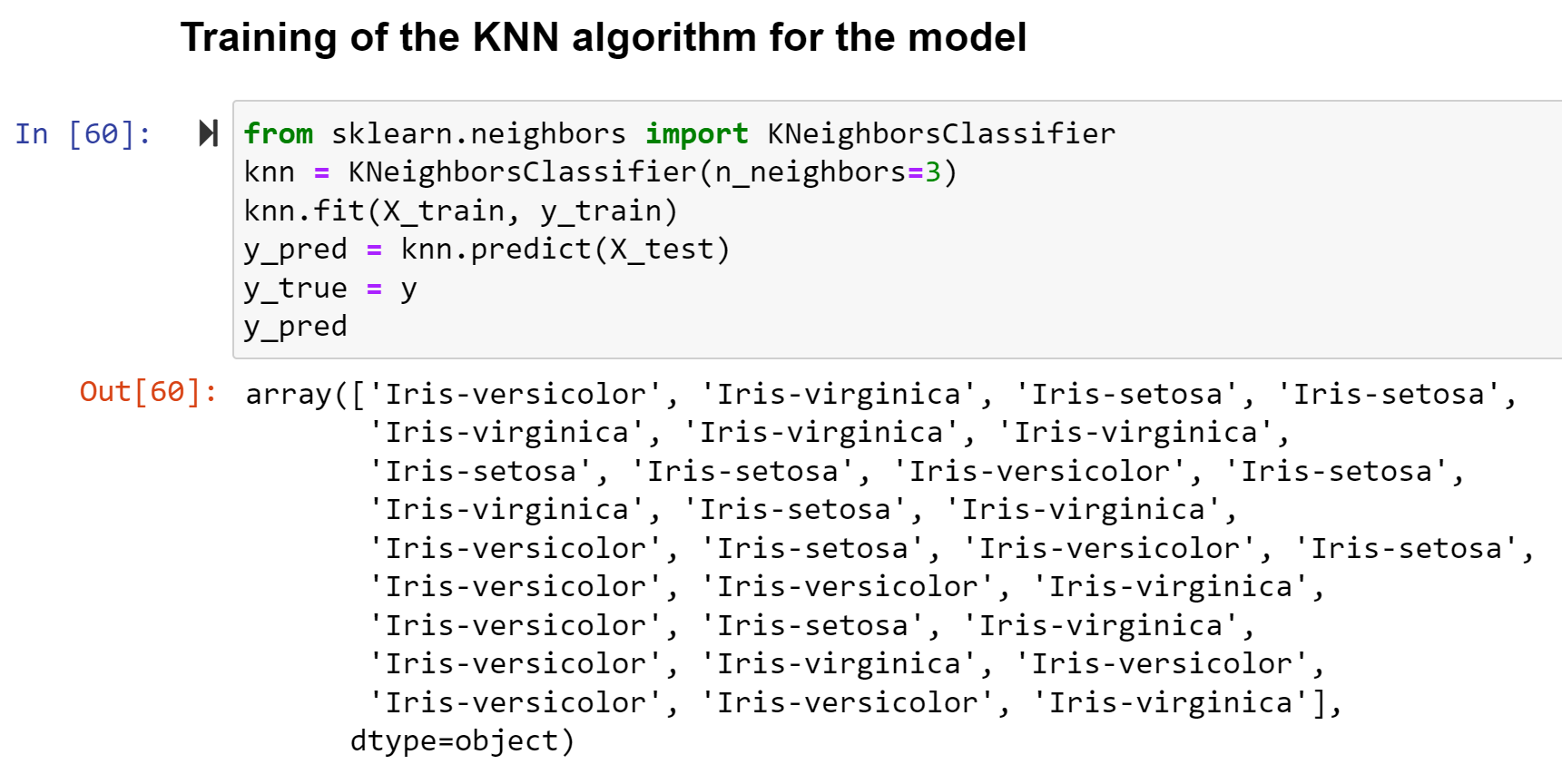
در این قسمت داده های ورودی که در یک فایل شامل 5 ستون قرار داشتند از یکدیگر تفکیک می شوند به این صورت که داده های ستون 1 تا 4 در متغیر X به عنوان ورودی و داده های ستون Class به عنوان خروجی در متغیر y ذخیره می شوند.



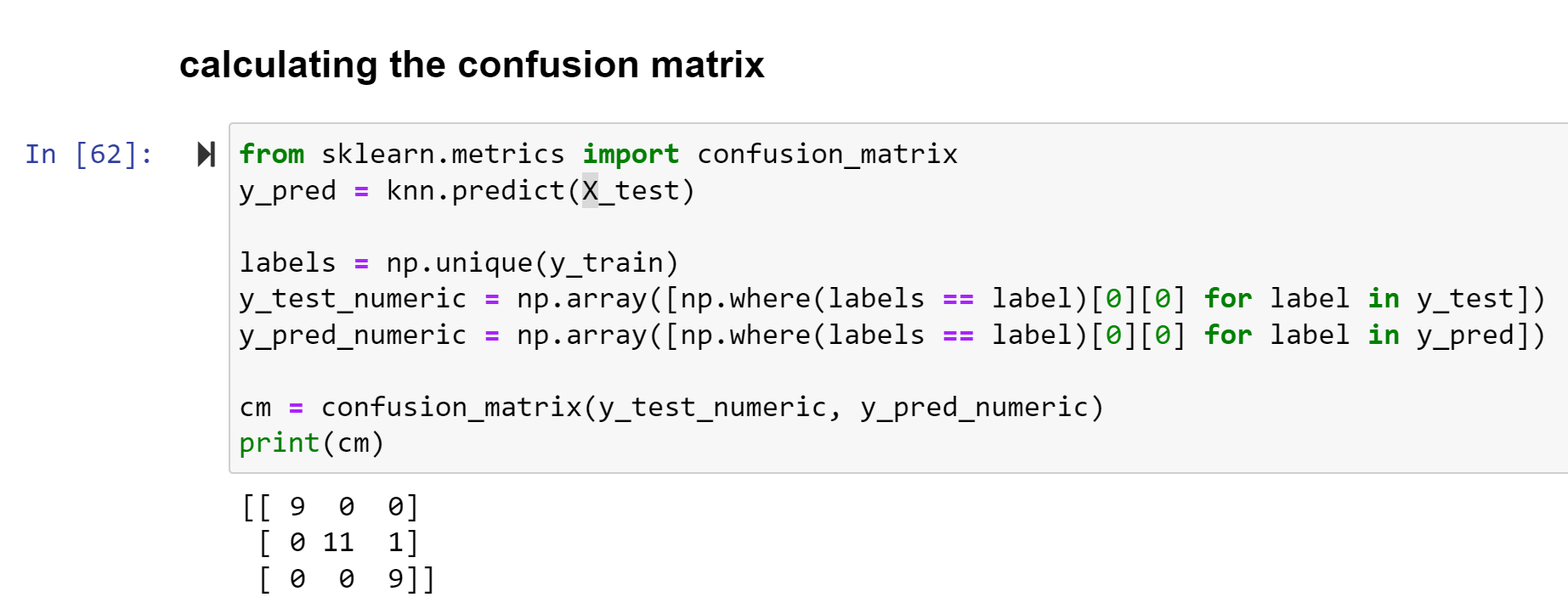


در صورت سوال ذکر شده که 80 درصد از داده ها برای تمرین و 20 درصد از داده ها برای تست در نظر گرفته شوند. برای این کار می توان از کتابخانه scikit learn استفاده کرد یا به صورتی که در کد بالا مشاهده می شود داده ها را به دوگروه تمرین و تست تقسیم کرد.

همچنین به منظور اینکه نتایج گزارش شده در این فایل با نتایج بدست آمده از کد همواره برابر باشد حالت random state به صورت ثابت تعریف می شود.می توان برای تصادفی بدست آمدن داده ها حالت random state را پاک کرد که در این صورت با هربار run کردن کد باتوجه به اینکه داده ها تصادفی در نظر گرفته می شوند نتایج متفاوت خواهد بود.



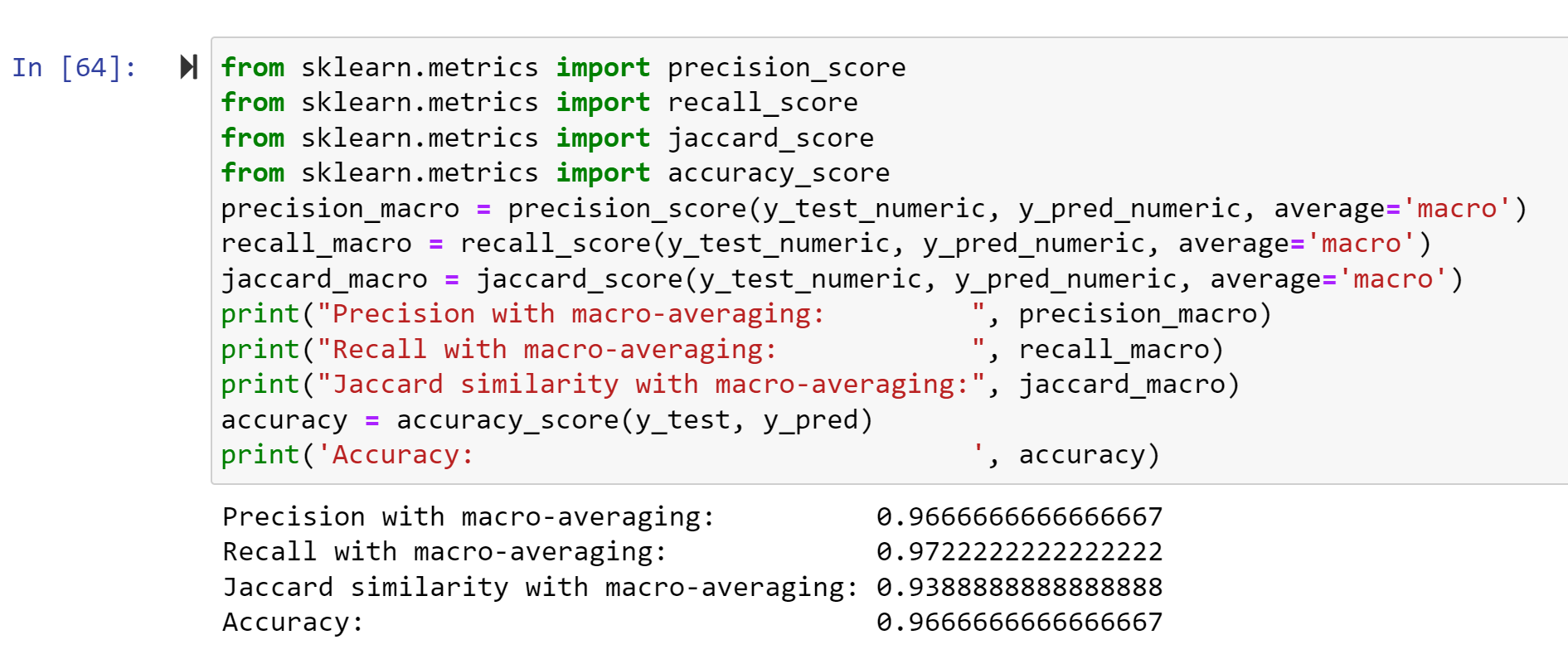
برای تربیت مدل با روش KNN برای دسته بندی گیاهان ابتدا از کتابخانه sklearn.neighbors دستور KNeighborsClassifier فراخوانی می شود. سپس K مربوط به این روش برابر با 3 قرار داده می شود و این مدل روی داده های آموزش fit می شود. پس از اینکه مدل روی داده های آموزش fit شد به کمک این مدل تربیت شده داده های جدا شده برای تست پیشبینی می شوند و در متغیر y\_pred قرار می گیرند.



در مرحله بعد برای محاسبه ماتریس سردرگمی ابتدا دستور مربوط به آن از sklearn.matrix فراخوانی می شود. برای تشکیل ماتریس سردرگمی باید داده های خروجی تست با داده های پیش بینی شده مقایسه شوند. از طرفی ستون خروجی از نوع object می باشد، ولی برای مقایسه ی دو مقدار به مقادیر کمی y\_pred و y\_test نیاز می باشد.

در پایان دستور فراخوانی شده برای تشکیل ماتریس سردرگمی بر روی داده های y\_test\_numeric و y\_pred\_numeric اعمال می شود و ماتریس سردرگمی به صورت شکل بالا یک ماتریس 3\*3 محاسبه می شود.

در این ماتریس مقادیر ستون اصلی مقادیری هستند که به درستی پیش بینی شده اند.(True positive )



مقادیر accuracy , jaccard similarity , Recall , Precision مطابق اعداد بالا بدست آمده اند.

Accuracy :

با توجه به اینکه دقت از مقادیر قطری ماتریس سردرگمی حساب می شود نیاز به میانگین گیری برای آن نیست و برای ماتریس سردرگمی به ازای همه خروجی ها یک accuracy بدست می آید.

Recall :

نسبت تعداد نمونه‌هایی است که الگوریتم به درستی به عنوان مثبت تشخیص داده است به تعداد کل نمونه‌های مثبت در داده‌ها.

Recall = TP / (TP + FN)

Jaccard :

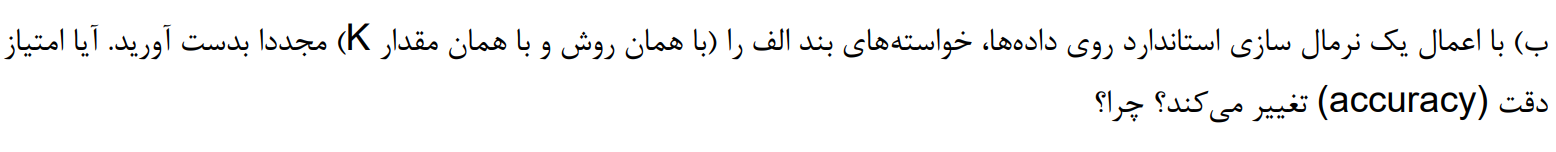
برابر با اندازه‌ی تقاطع دو مجموعه تقسیم بر اندازه‌ی اتحاد آن دو مجموعه است. به عبارت دیگر، این امتیاز نشان می‌دهد که چه مقدار اشتراک بین دو مجموعه وجود دارد.

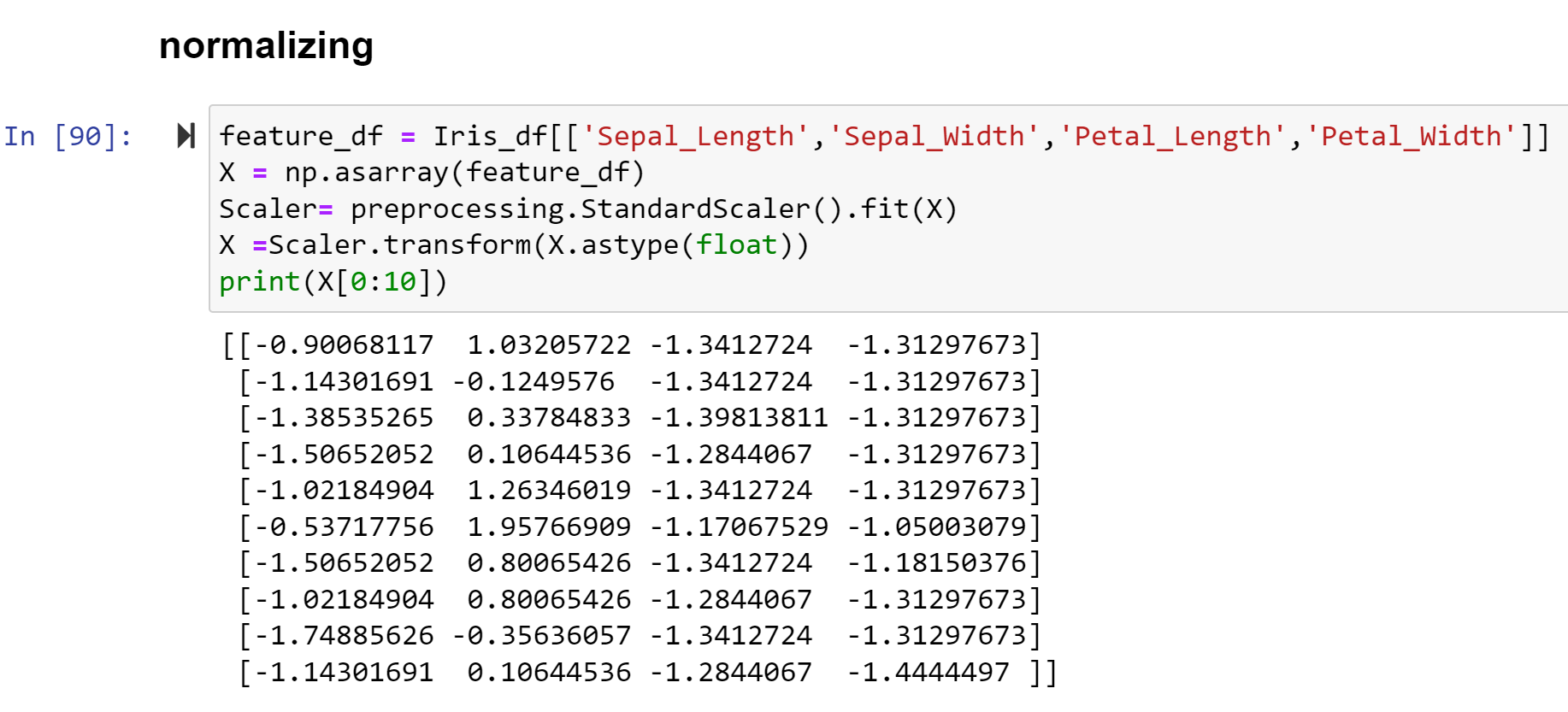
J(A, B) = |A ∩ B| / |A ∪ B|

Precision :

اندازه‌ی تعداد نمونه‌های مثبتی که الگوریتم به درستی به عنوان مثبت تشخیص داده است، تقسیم بر تعداد کل نمونه‌هایی که الگوریتم به عنوان مثبت تشخیص داده است.

Precision = TP / (TP + FP)

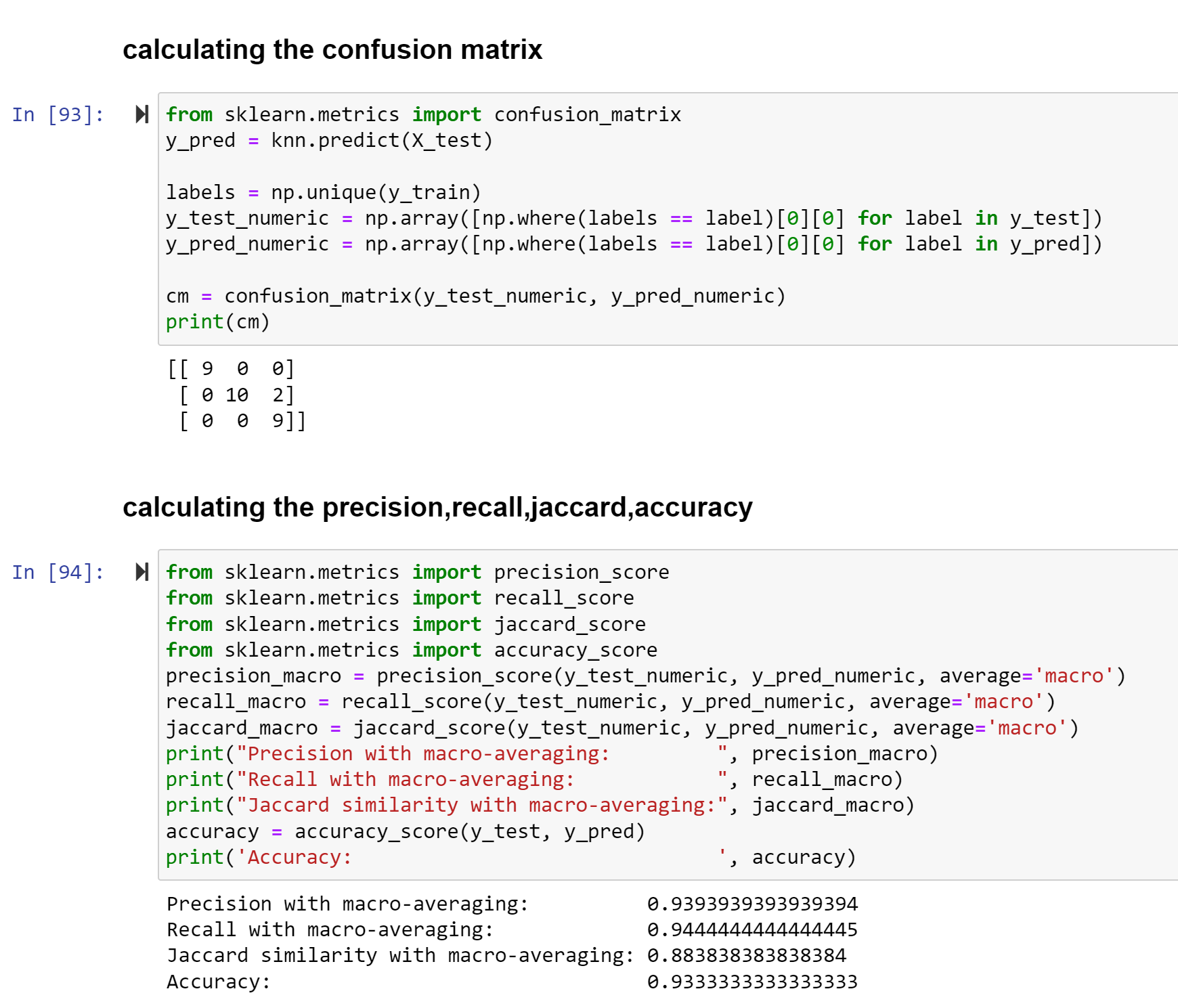




برای قسمت ب ابتدا داده های ورودی با دستور preprocessing از کتابخانه sklearn فراخوانی می شود و به کمک آن داده های ورودی normalize می شوند.

پس از اعمال normalizing سایر مراحل مانند قسمت الف انجام می شود تا مقادیر خواسته شده ی accuracy , jaccard similarity , Recall , Precision برای این سمت نیز محاسبه شوند.

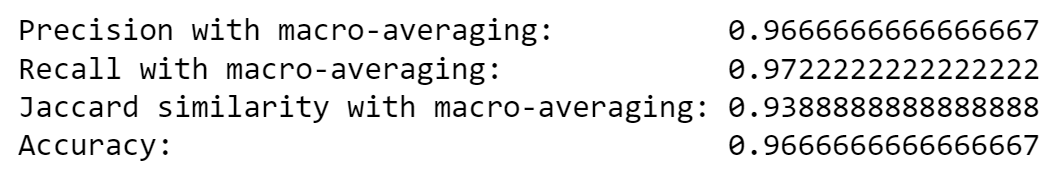
یعنی مانند قسمت الف داده ها به دوقسمت تست و آموزش تقسیم می شوند و مدل با الگوریتم KNN با (K=3) تربیت می شود تا خروجی های مطلوب پیش بینی شوند.

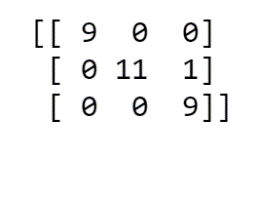


همانطور که مشاهده می شود مقادیر accuracy , jaccard similarity , Recall , Precision در این قسمت نیز پس از normalizing داده های ورودی بدست آورده شده اند.

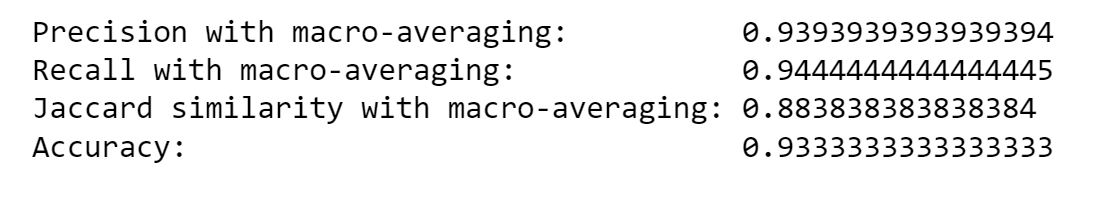
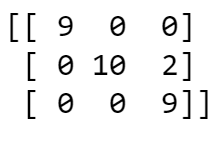
همانطور که مشاهده می شود همه ی این مقادیر پس از normalizing کاهش یافته اند که در ادامه به مقایسه ماتریس سردر گمی و accuracy , jaccard similarity , Recall , Precision قبل و پس از اعمال normalizing پرداخته می شود.

قبل از normalizing :





بعد از normalizing :

همانطور که مشاهده می شود پس از نرمال سازی داده های ورودی دقت پیشبینی مدل کاهش یافته است.

در مدل KNN، نرمال‌سازی داده‌ها می‌تواند در برخی موارد بهبود دقت را به دنبال داشته باشد اما در موارد دیگر، دقت را کاهش می‌دهد. یک دلیل این کاهش دقت پس از نرمال‌سازی داده‌ها در مدل KNN می‌تواند از دست‌رفتن اطلاعات پیرامونی باشد.

در مدل KNN، برای پیدا کردن همسایگان نزدیک به نمونه‌ی جدید، از فاصله‌ی اقلیدسی یا همان فاصله‌ی مربعی استفاده می‌شود. با نرمال‌سازی داده‌ها، مقادیر آن‌ها به یک دامنه‌ی مشخصی تغییر می‌کنند؛ به عبارت دیگر، مقادیر بیشتر و کوچکتر به یک‌دیگر نزدیک‌تر می‌شوند. این تغییر در مقادیر، ممکن است باعث شود که فاصله‌ی اقلیدسی (یا فاصله‌ی مربعی) بین دو نمونه کوچک‌تر یا بزرگ‌تر از واقعیت شود. با این حال، در مدل KNN، فاصله‌ی اقلیدسی (یا فاصله‌ی مربعی) بین نمونه‌ها بسیار مهم است و به عنوان یکی از معیارهای اصلی برای پیدا کردن همسایگان نزدیک استفاده می‌شود. بنابراین، نرمال‌سازی داده‌ها ممکن است باعث از دست رفتن اطلاعات پیرامونی و بهبود دقت در مدل KNN نشود.

به طور کلی، در مدل KNN، نرمال‌سازی داده‌ها باید با دقت انجام شود و اگر نرمال‌سازی بهبودی در دقت مدل نداشت، می‌تواند به عنوان یکی از عوامل کاهش دقت در نظر گرفته شود. بهترین روش برای تعیین اینکه آیا نرمال‌سازی داده‌ها در مدل KNN بهبودی در دقت مدل دارد یا نه، تست مدل با و بدون نرمال‌سازی داده‌ها است.