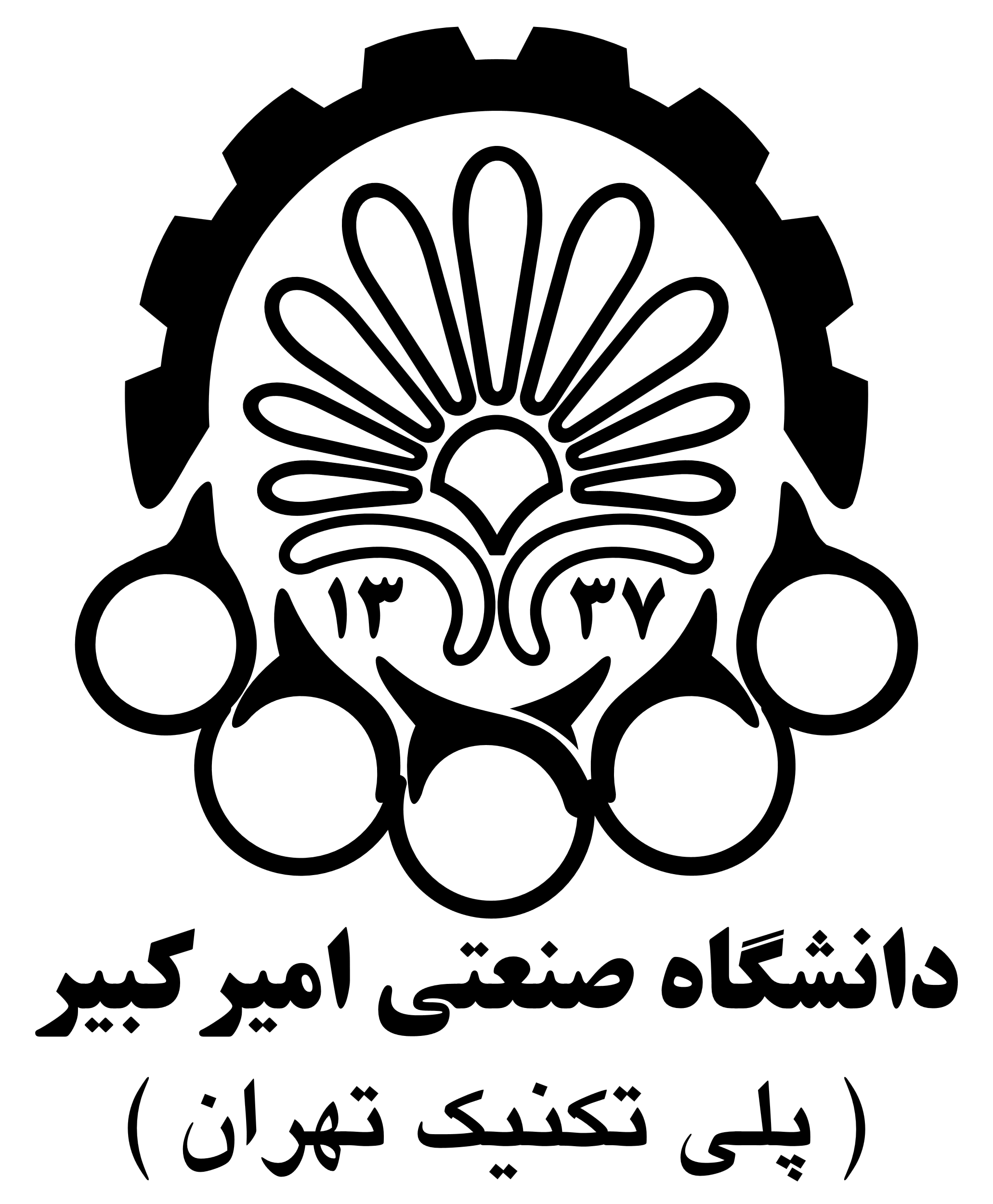
****

**دانشکده مهندسی کامپیوتر**

**تمر ین دوم درس یادگیری ماشین**

**دکتر ناظرفرد**

**غلامرضا دار 400131018**

**پاییز ۱۴۰۰**

فهرست مطالب

[بخش اول: پرسشهای تشریحی 3](#_Toc90605747)

[سوال 1) 3](#_Toc90605748)

[سوال 2) 5](#_Toc90605749)

[سوال 3) 5](#_Toc90605750)

[سوال 4) 7](#_Toc90605751)

[سوال 5) 15](#_Toc90605752)

[بخش دوم: پیاده سازی 18](#_Toc90605753)

[سوال 1) 18](#_Toc90605754)

[سوال 2) 22](#_Toc90605755)

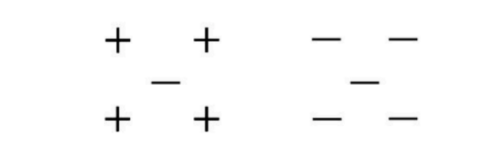
[سوال 3) 26](#_Toc90605756)

[منابع 29](#_Toc90605757)

# بخش اول: پرسشهای تشریحی

## سوال 1)

الف)



**A**

**F**

**I**

**J**

**H H**

**G**

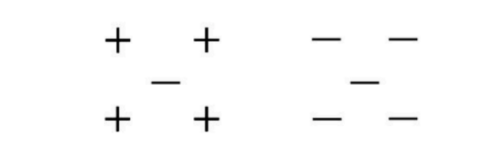
**E**

**D**

**C**

**B**

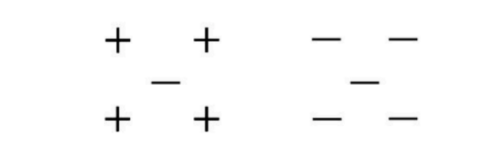
برای آسان تر شدن مسئله به هر کدام از نقاط دیتاست یک اسم میدهیم. در ادامه قرار است 10 بار، هر بار یکی از این نقاط را به عنوان دیتای تست در نظر بگیریم (علامت سوال بگذاریم). این کار را برای مقادیر مختلف K انجام میدهیم. بهترین مقدار برای K مقداری است که کمترین میزان دسته بندیِ اشتباه را داشته باشد.



-

**?**

شکل 1- با فرض K=1 و حذف کردن داده A به عنوان داده تست



+

**?**

شکل - با فرض K=3 و حذف کردن داده A به عنوان داده تست

در شکل های قبل دیدیم که با فرض داده A به عنوان بخش test در روش LOOCV و با فرض K=1 کلاس منفی به A نسبت داده شد که اشتباه است. اما با فرض K=3 چون همسایه های A مثبت بودند کلاس مثبت به A نسبت داده شد پس میتوان گفت برای A مقدار K=3 مناسب تر است.

**نکته**: چون تعداد کلاس های این مسئله 2 است(مثبت و منفی) بهتر است مقادیر **فرد** را برای K در نظر بگیریم تا حالت برابر بودن تعداد همسایه های منفی و مثبت رخ ندهد.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Total Error | **J** | **I** | **H** | **G** | **F** | **E** | **D** | **C** | **B** | **A** | **Data**  **K** |
| 5 | ✔ | ✔ | ✔ | ✔ | ✔ | ✖ | ✖ | ✖ | ✖ | ✖ | **1** |
| 1 | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✖ | ✔ | ✔ | **3** |
| 1 | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✖ | ✔ | ✔ | **5** |
| 5 | ✔ | ✔ | ✔ | ✔ | ✔ | ✖ | ✖ | ✖ | ✖ | ✖ | **7** |
| 4 | ✔ | ✔ | ✔ | ✔ | ✔ | ✖ | ✖ | ✔ | ✖ | ✖ | **9** |

همانطور که در جدول مشاهده میشود، مقادیر K=5 و K=3 کمترین میزان خطا را در حالت های مختلف داشتند. بنابر این بهترین مقدار K عدد 5 یا 3 است.

ب) راهی که در بخش الف استفاده شد. به ازای مقادیر مختلف K و با کمک Cross validation خطای مدل به ازای آن K را بدست می‎آوریم. بهترین K آن K ای است که کمترین خطا را به ازای نقاط مختلفِ داده داشته باشد.

## سوال 2)

هر دو Discriminative هستند زیرا به طور مستقیم یا همان Posterior Probability را یاد میگیرند و اطلاعی از توزیع داده ها ندارند. در واقع را نمیدانند و این باعث میشود نتوانند داده جدیدی تولید کنند.

در مقابل این دو مدل، مدل Naïve Bayes یک مدل Generative است زیرا توزیع داده ها را یاد میگیرد و با کمک آن Posterior Probability را حساب میکند. از آنجایی که این مدل توزیع داده ها را دارد میتواند داده های جدیدی تولید کند.

## سوال 3)

الف)

ب)

ج)

بخاطر اینکه سوال **منفی** Log Likelihood را خواسته.

**د)** در حالتی که مجموعه داده های دو کلاس خطی جداپذیر باشند بیش برازشی که رخ میدهد احتمالا به شکل یک انحراف به سمت یکی از کلاس هاست به این دلیل که ممکن است تعداد داده های دو کلاس یک اندازه نباشند و مدل حین یادگیری به یک کلاس وزن بیشتری بدهد.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Age | Income | Student | Credit | Buy |
| Youth | High | No | Fair | No |
| Youth | High | No | Excellent | No |
| Middle | High | No | Fair | Yes |
| Senior | Medium | No | Fair | Yes |
| Senior | Low | Yes | Fair | Yes |
| Senior | Low | Yes | Excellent | No |
| Middle | Low | Yes | Excellent | Yes |
| Youth | Medium | No | Fair | No |
| Youth | Low | Yes | Fair | Yes |
| Senior | Medium | Yes | Fair | Yes |
| Youth | Medium | Yes | Excellent | Yes |
| Middle | Medium | No | Excellent | Yes |
| Middle | High | Yes | Fair | Yes |
| Senior | Medium | No | Excellent | No |

## سوال 4)

الف)

* X1 = (age = youth, income = high, student = yes, credit = fair)
* X2 = (age = senior, income = low, student = no, credit = excellent)
* X3 = (age = middle-aged, income = medium, student = no, credit = fair)

برای ساده تر شدن حل این مسئله جدول زیر را تولید میکنیم.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature | Value | Count (Buy = Yes) | Count (Buy = No) | Sum |
| Age | Youth | 2 | 3 | 5 |
| Middle | 4 | 0 | 4 |
| Senior | 3 | 2 | 5 |
| Income | Low | 3 | 1 | 4 |
| Medium | 4 | 2 | 6 |
| High | 2 | 2 | 4 |
| Student | No | 3 | 4 | 7 |
| Yes | 6 | 1 | 7 |
| Credit | Fair | 6 | 2 | 8 |
| Excellent | 3 | 3 | 6 |
|  |  | Total = 9 | Total = 5 | Total = 14 |

*X1 = (age = youth, income = high, student = yes, credit = fair)*

رابطه Bayes :

حل مسئله برای "***خریدن فرد***" :

با فرض مستقل بودن ویژگی ها :

از روی جدول داریم :

با جاگذاری مقادیر در رابطه اصلی داریم :

احتمال خریدن فرد X1 برابر 0.483 است.

*X2 =* (age = senior, income = low, student = no, credit = excellent)

رابطه Bayes :

حل مسئله برای "***خریدن فرد***" :

با فرض مستقل بودن ویژگی ها :

از روی جدول داریم :

با جاگذاری مقادیر در رابطه اصلی داریم :

احتمال خریدن فرد X2 برابر 0.362 است.

*X3 =* (age = middle-aged, income = medium, student = no, credit = fair)

رابطه Bayes :

حل مسئله برای "***خریدن فرد***" :

با فرض مستقل بودن ویژگی ها :

از روی جدول داریم :

با جاگذاری مقادیر در رابطه اصلی داریم :

احتمال خریدن فرد X3 برابر 0.805 است.

ب) ابتدا Entropy(S) را محاسبه میکنیم. سپس Information gain همه Feature ها را محاسبه میکنیم. فیچری که بیشترین Information gain را داشته باشد را به عنوان راس درخت انتخاب میکنیم و با کمک آن دیتاست را افراز میکنیم.

ویژگی Age بیشترین Information Gain را دارد پس در مرحله اول از این ویژگی برای تقسیم کردن دیتاست استفاده میکنیم.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Age | Income | Student | Credit | Buy |
| Youth | High | No | Fair | No |
| Youth | High | No | Excellent | No |
| Youth | Medium | No | Fair | No |
| Youth | Low | Yes | Fair | Yes |
| Youth | Medium | Yes | Excellent | Yes |

Middle

Youth

Senior

در این بخش به دنبال بهترین ویژگی برای علامت سوال سمت چپ میگردیم.

در این بخش ویژگی Student بیشترینInformation Gain را دارد پس در این بخش از این ویژگی برای تقسیم کردن دیتاست استفاده میکنیم.

Middle

Senior

Youth

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Age | Income | Student | Credit | Buy |
| Senior | Medium | No | Excellent | No |
| Senior | Low | Yes | Excellent | No |
| Senior | Low | Yes | Fair | Yes |
| Senior | Medium | No | Fair | Yes |
| Senior | Medium | Yes | Fair | Yes |

در این بخش به دنبال بهترین ویژگی برای علامت سوال سمت راست میگردیم.

در این بخش ویژگی Credit بیشترینInformation Gain را دارد پس در این بخش از این ویژگی برای تقسیم کردن دیتاست استفاده میکنیم.

Excellent

Fair

Yes

No

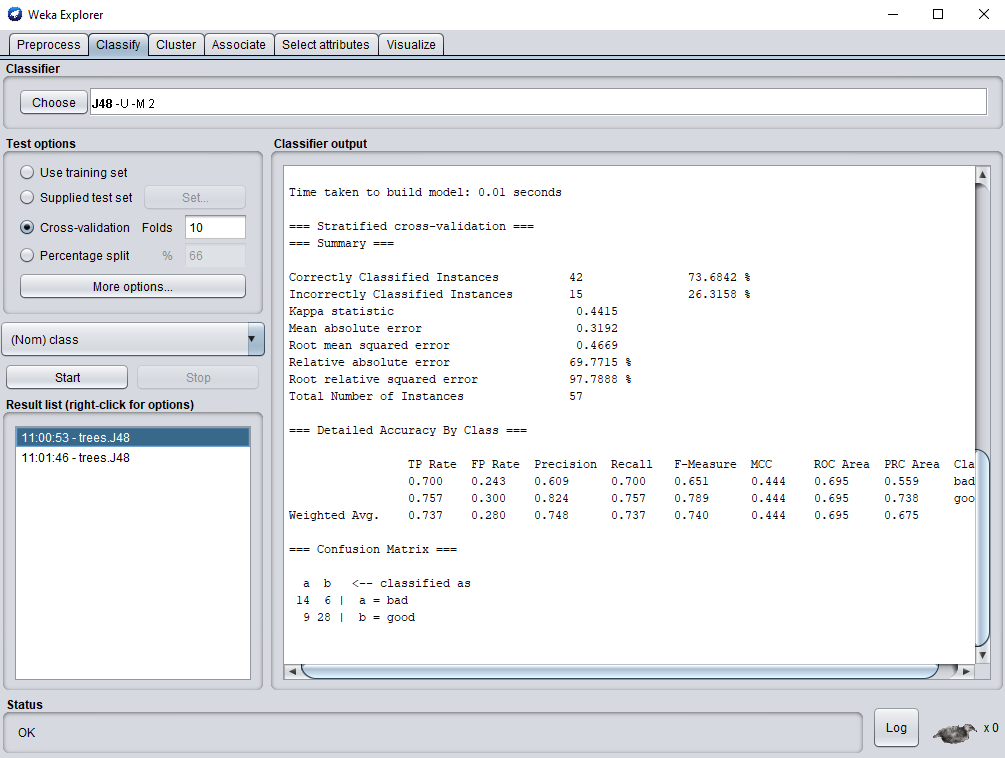
Middle

Senior

Youth

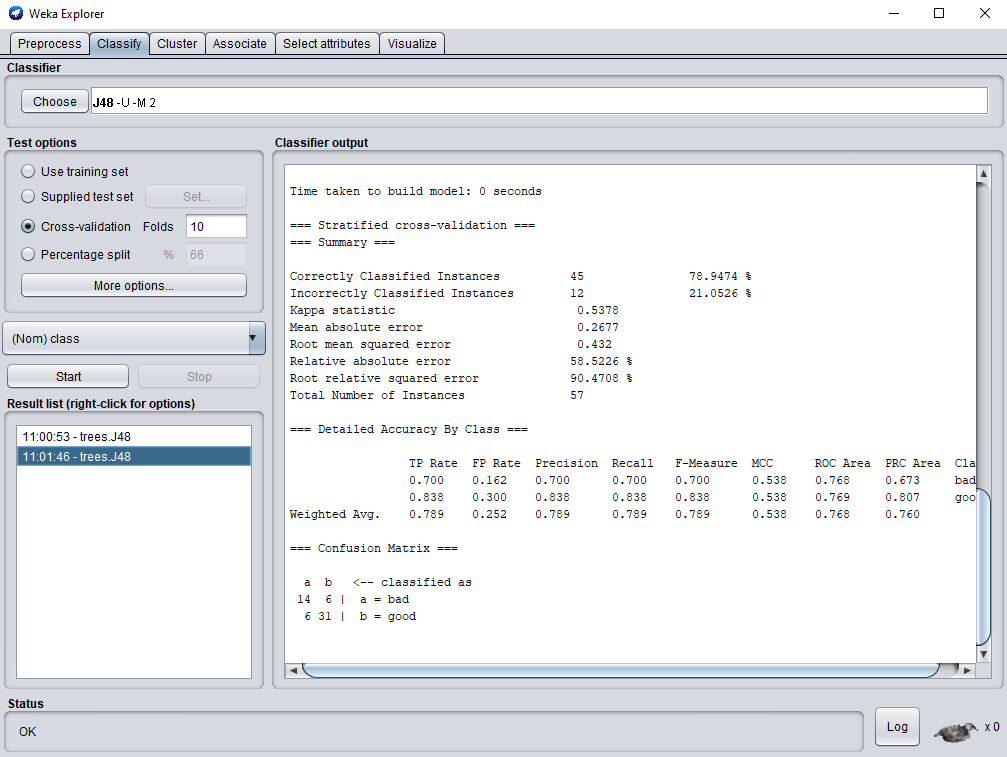
## سوال 5)

**الف)**

****

**ب)**

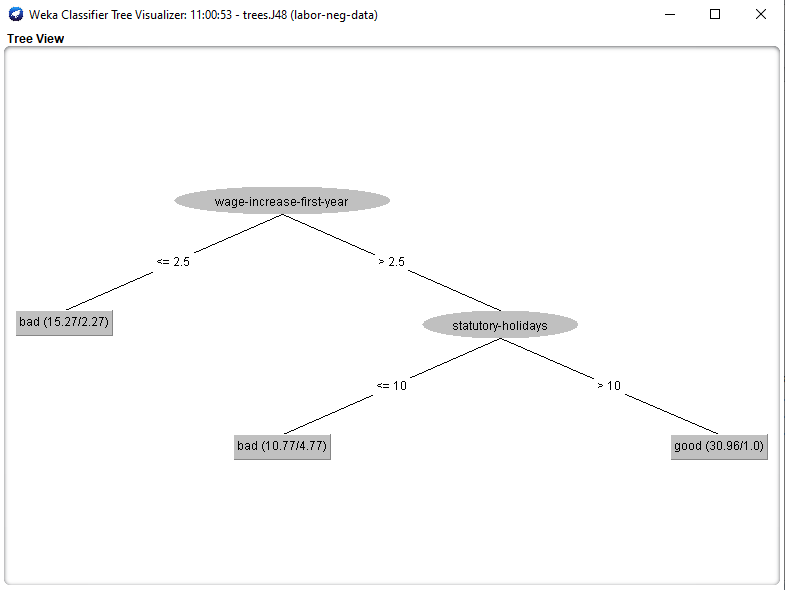
|  |  |  |
| --- | --- | --- |
|  | Classified as Bad | Classified as Good |
| Bad | 14 | 6 |
| Good | 9 | 28 |

**ج)** ****

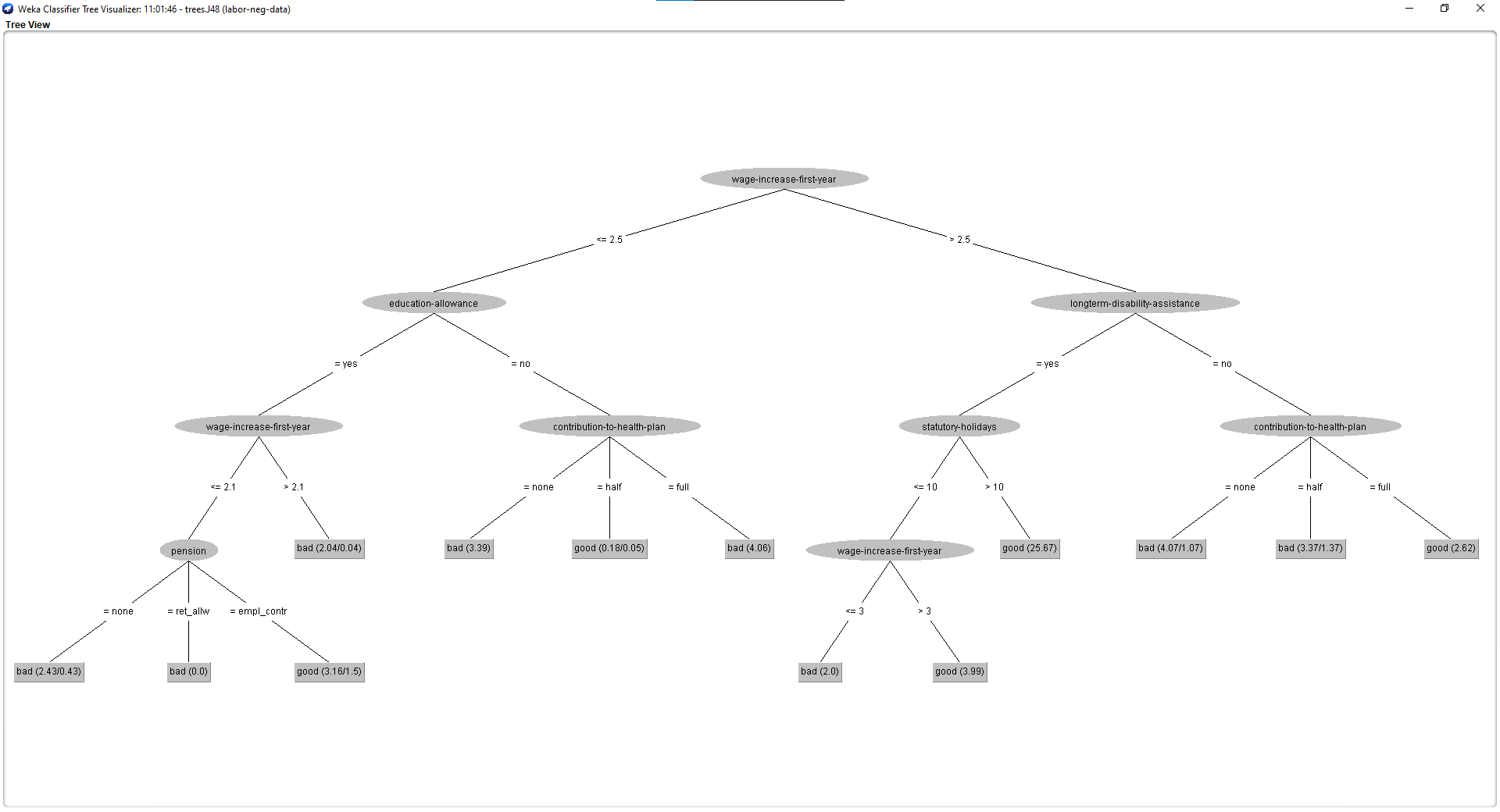
|  |  |  |
| --- | --- | --- |
|  | Classified as Bad | Classified as Good |
| Bad | 14 | 6 |
| Good | 6 | 31 |

**د)**  تفاوتی که بین درخت تصمیم هرس شده و نشده دیده میشود در ارتفاع دو درخت و همچنین میزان تصمیم های اشتباه دو درخت است.

درخت تصمیم هرس شده دارای اشتباهات بیشتری از درخت هرس نشده است اما احتمالا قابلیت generalize شدن بیشتری دارد.

****

شکل - درخت تصمیم هرس شده



شکل - درخت تصمیم هرس نشده

# بخش دوم: پیاده سازی

## سوال 1)

نوت بوک مربوطه به این تمرین :

<https://colab.research.google.com/drive/1hs3OGPk5St9bRO-bbWPr4TjBbgeGxlFn?usp=sharing>

ابتدا داده ها را لود میکنیم. سپس لازم است دیتاست را shuffle کنیم و بعد از آن به نسبت 80-20 به بخش Train, Test تقسیم کنیم.

**الف)**

برای بخش الف سوال ویژگی های گسسته را مستقل از سایر ویژگی ها و ویژگی های پیوسته را تحت یک توزیع چند متغیره نرمال در نظر میگیریم. هدف از این سوال محاسبه Posterior Probability های زیر است:

سپس با مقایسه کردن این دو احتمال به این نتیجه میرسیم که کدام outcome مناسب تر است و آن را به عنوان prediction برای دیتای X (فرد X) در نظر میگیریم.

نکات قابل توجه در این سوال:

* بهتر است برای جلوگیری از float underflow به جای ضرب کردن احتمال های متعدد( که هر کدام ممکن است عدد بسیار کوچکی باشند) از posterior لگاریتم میگیریم تا این ضرب ها به جمع تبدیل شوند. لازم به ذکر است که این کار چون برای هر دو posterior انجام میشود و هدف مقایسه این دو است، تغییری در نتیجه ایجاد نمیکند.
* برای رفع مشکل احتمال 0برای برخی ویژگی ها، از Laplace smoothing استفاده میکنیم به این شکل که به هنگام محاسبه احتمال صورت و مخرج را با 1 جمع میکنیم. این کار باعث میشود صفر شدن احتمال یک ویژگی باعث صفر شدن همه ترم های دیگر نشود.
* چون هر دو posterior بر عدد P(X) تقسیم میشوند، میتوانیم این تقسیم را نادیده بگیریم بدون آنکه در نتیجه تغییری ایجاد کنیم.

برای محاسبه به شکل زیر عمل میکنیم :

*و بخاطر مستقل بودن ویژگی های گسسته داریم :*

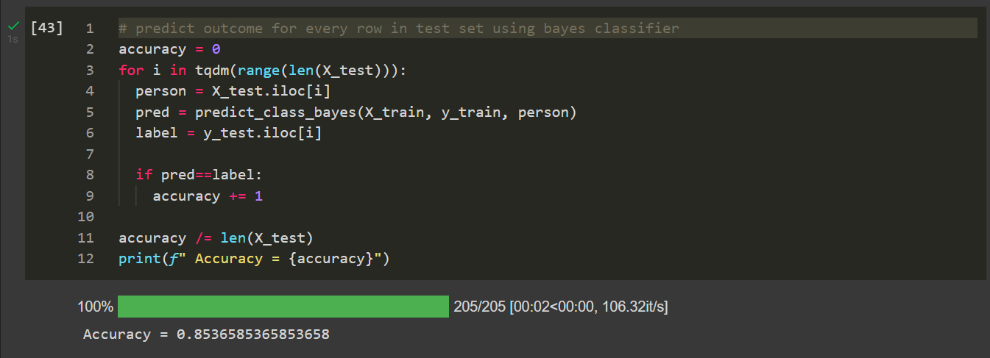
که در آن توزیع چند متغیره گاوسی روی داده های با Label **مثبت** برای ویژگی های پیوسته است.

به طور مشابه برای داریم:

*و بخاطر مستقل بودن ویژگی های گسسته داریم :*

که در آن توزیع چند متغیره گاوسی روی داده های با Label **منفی** برای ویژگی های پیوسته است.

پس از آموزش Bayes classifier روی داده های Train، کارایی مدل روی داده های Test ارزیابی شد که نتیجه، صحت 85% بود.



شکل 5 - آموزش و ارزیابی مدل Bayes Classifier

**ب)**

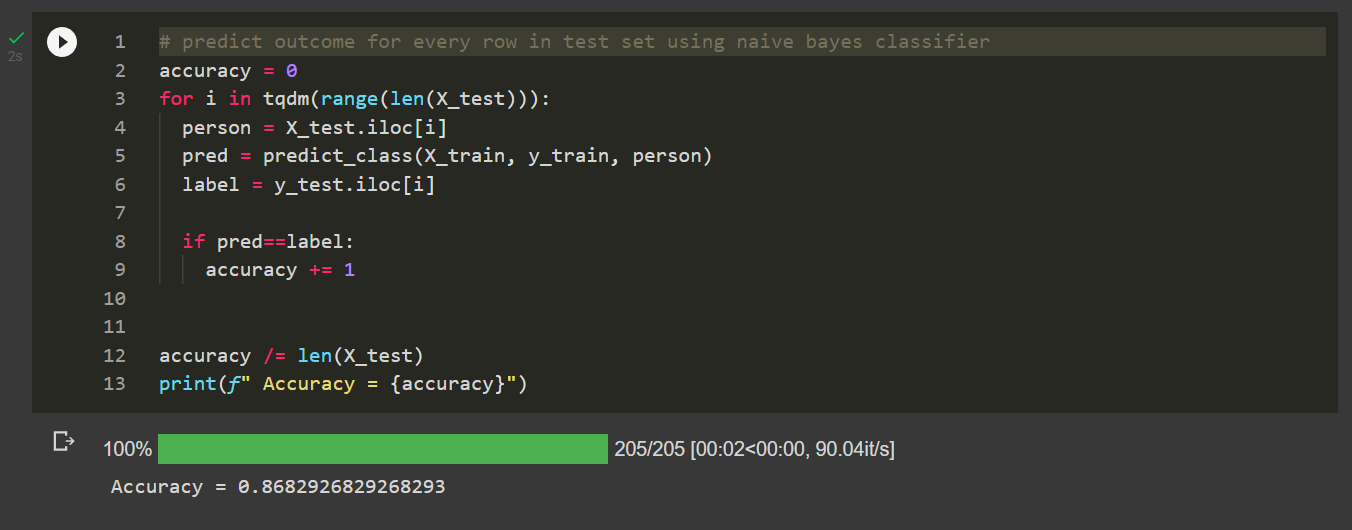
در این قسمت مشابه قسمت الف عمل میکنیم با این تفاوت که در این مثال همه ویژگی ها مستقل هستند نه فقط ویژگی های گسسته. بنابراین محاسبه به شکل زیر انجام می‎شود.

که در آن توزیع نرمال تک متغیری روی هر کدام از ویژگی های پیوسته با داده های label **مثبت** است.

به طور مشابه برای محاسبه داریم:

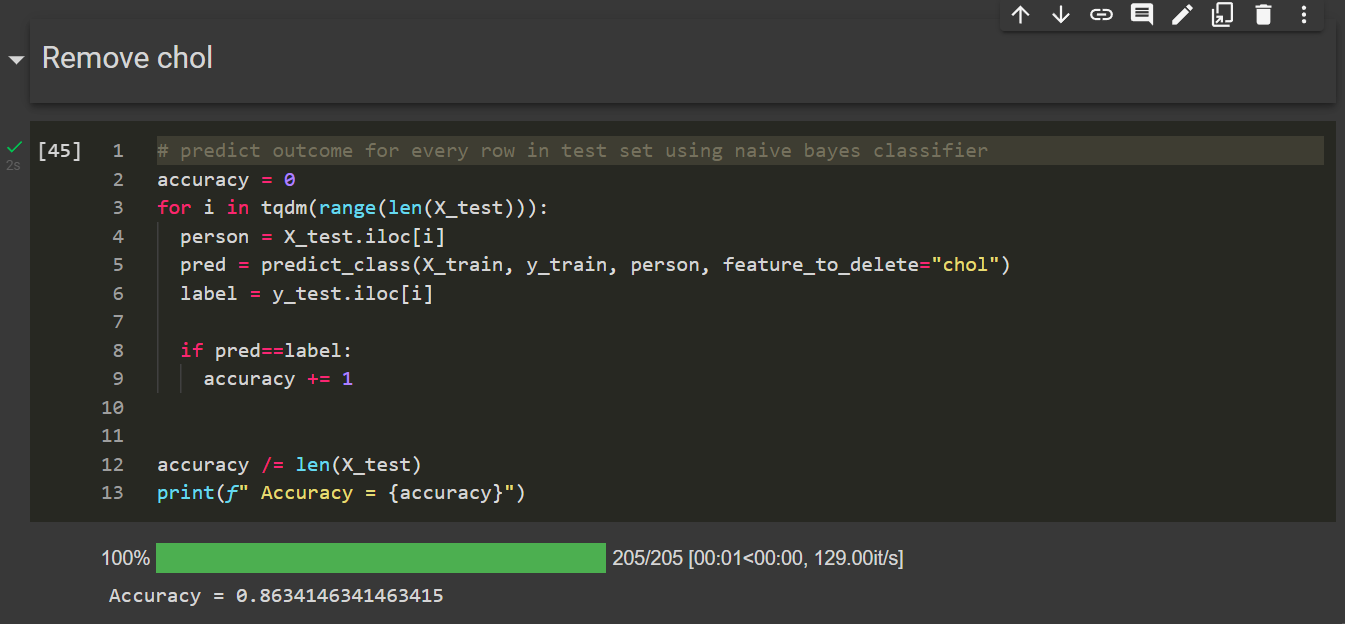
که در آن توزیع نرمال تک متغیری روی هر کدام از ویژگی های پیوسته با داده های label **منفی** است.

در این مورد به صحت 86% رسیدیم.

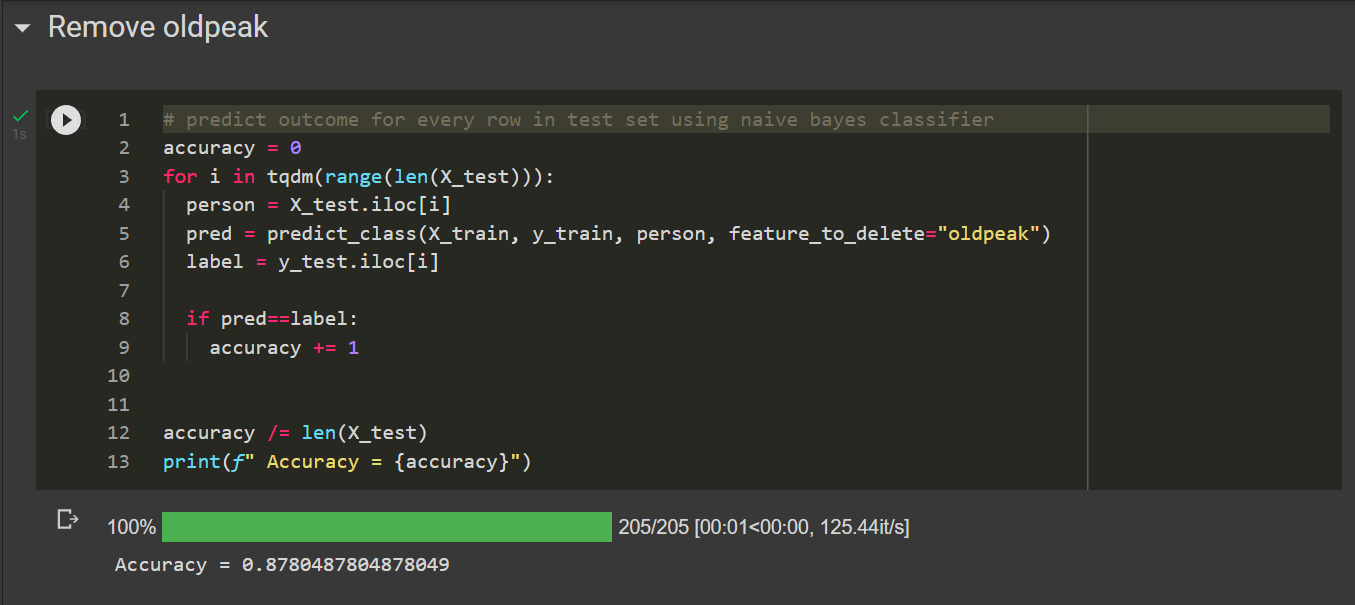


شکل 6 - آموزش و ارزیابی مدل Naïve Bayes Classifier

**ج)** نتیجه مدل قبل با حذف ویژگی های Chol, Oldpeak را به ترتیب در تصاویر زیر مشاهده میکنید.

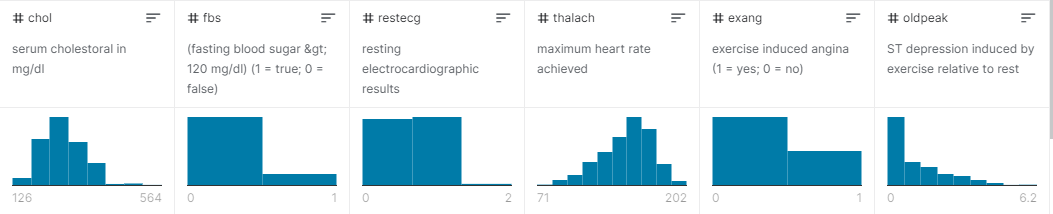


شکل 7- نتیجه ناییو بیز پس از حذف ویژگی Chol



شکل 8 - نتیجه ناییو بیز پس از حذف ویژگی Oldpeak

همانطور که مشاهده میشود، حذف ویژگی Oldpeak بیشترین تاثیر را در صحت مدل می‎گذارد. دلیلی که ممکن است باعث این اتفاق شده باشد این است که مقادیر ویژگی Oldpeak اصلا از توزیع نرمال پیروی نمیکنند. با مراجعه به صفحه دیتاست در وبسایت Kaggle نیز میتوان این مورد را مشاهده کرد.

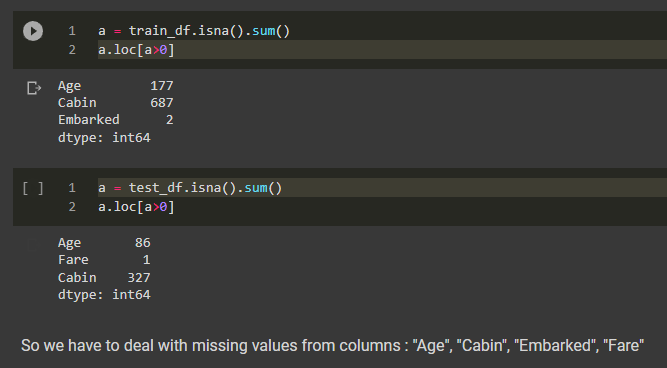


## سوال 2)

نوت بوک مربوطه به این تمرین :

<https://colab.research.google.com/drive/1PQ_OsTzple1dd5EEmL0-kiUc9Ne5xfIR?usp=sharing>

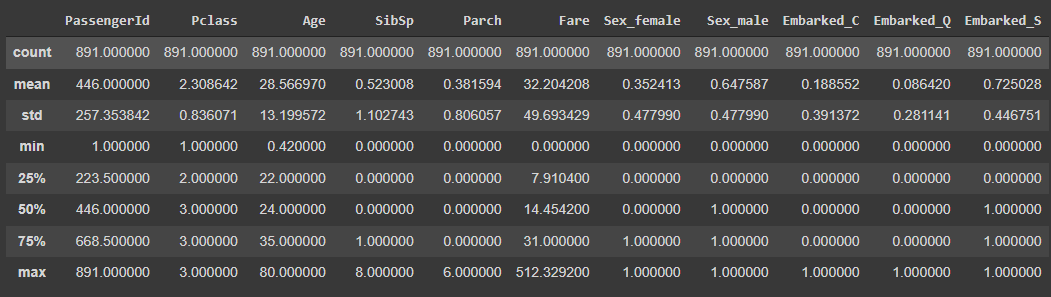
**الف)** ابتدا داده ها را load میکنیم و تعداد مقادیر گم شده هر ستون را پیدا میکنیم.



همانطور که دیده میشود، ستون های Age, Cabin, Embarked, Fare نیاز به بررسی دارند. بقیه ستون ها مقدار گم شده ندارند. لازم است برای این ویژگی ها و یک سری ویژگی های دیگر تصمیماتی بگیریم. بعد از بررسی های انجام شده روی دیتاست تصمیم های زیر گرفته شد:

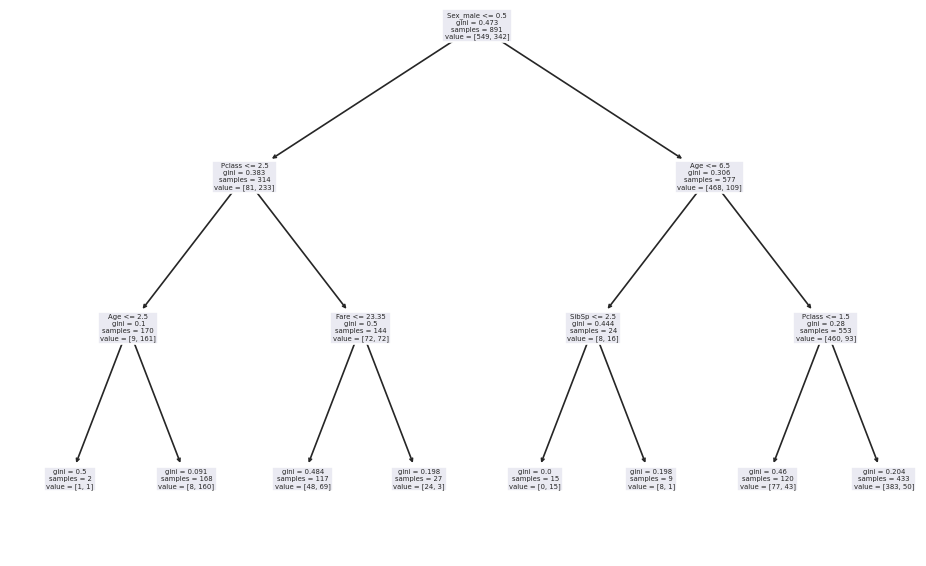
* **ویژگیEmbarked :** جایگزین کردن مُد داده ها به جای مقادیر گم شده.
* **ویژگی Age** **:** جایگزین کردن مُد داده ها به جای مقادیر گم شده (مناسب تر از mean است چون outlier ها را در نظر نمیگیرد)
* **ویژگی Fare** **:** جایگزین کردن میانگین مقادیر با مقادیر گم شده.
* **ویژگی Name** **:** از آنجایی که اسم یک فرد تاثیری در زنده ماندن یا نماندن آن فرد ندارد ستون Name را حذف میکنیم.
* **ویژگی Cabin :** از آنجایی که تعداد بسیار زیادی از مقادیر این ویژگی گم شده است، جایگزین کردن میانگین میتواند مدل را گم راه کند و اطلاعات نادرستی به دیتاست اضافه کند. بنابراین این ویژگی را نیز حذف میکنیم.
* **ویژگی Ticket** **:** این ویژگی احتمالا میتواند پس از یک مرحله استخراج اطلاعات مفید واقع شود( مثلا تحلیل کنیم که اعداد و حروف های روی Ticket به چه معنا هستند) اما برای این سوال این ویژگی را حذف میکنیم.

پس از این مرحله نیاز است مقادیر Categorical را Tokenize کنیم تا به مقادیر عددی تبدیل شوند. ویژگی های Sex, Embarked از این دسته ویژگی ها هستند که به کمک روش one-hot encoding این داده ها را Tokenize میکنیم.



شکل 9 - نمای کلی دیتاست پس از انجام عملیات های پیش پردازش

**ب)** در این مرحله دیتاست ما آماده استفاده به عنوان دیتای ترین یک مدل DecisionTree است.

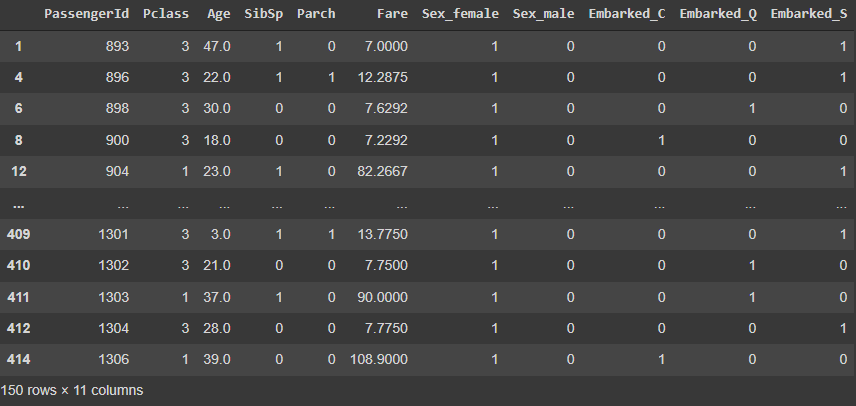


شکل 10 - درخت تصمیم تولید شده برای این مسئله

امتیاز(score) اعلام شده توسط sklearn برای این درخت تصمیم برابر 0.827160493 بود.

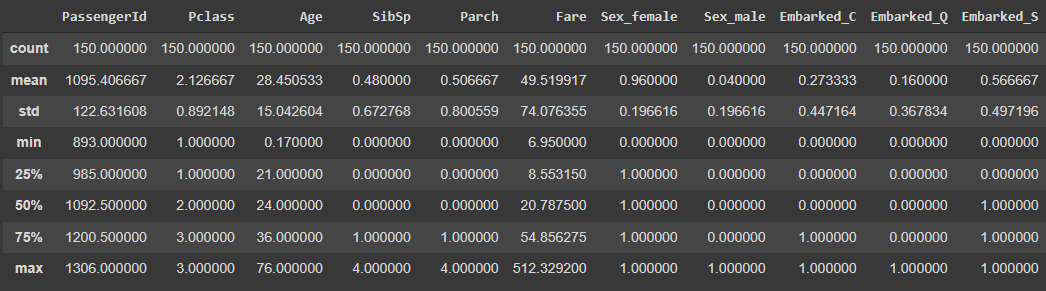
**ج)** نتایج دسته بندی مسافران در دیتاست Test را میتوانید در این بخش مشاهده کنید.

در شکل 7 خلاصه ای از افراد بازمانده را مشاهده میکنید.

****

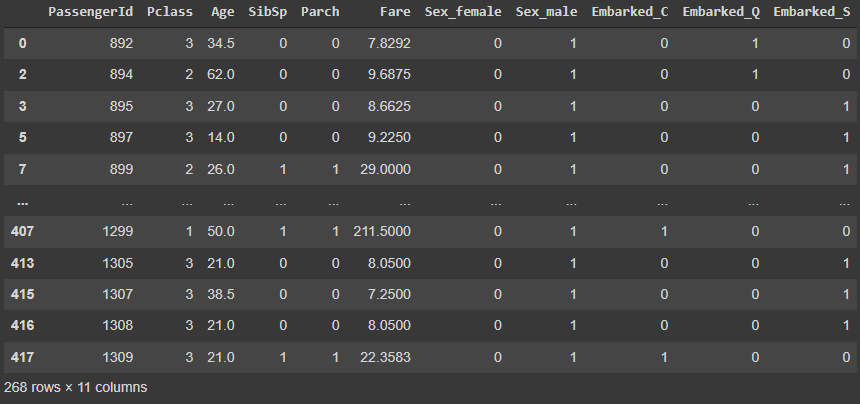
شکل 11 - 150 نفر از مسافران نجات یافتند

همچنین در این بخش میتوانیم ببینیم افراد بازمانده چه ویژگی هایی داشته اند یا مقادیر ویژگی های مختلف آنها در چه بازه ای یا دارای چه میانگینی بوده است که میتواند بسیار مفید باشد.



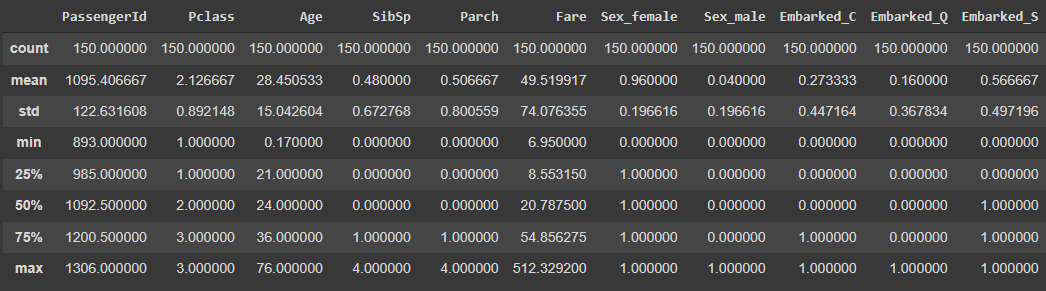
شکل 8 - خلاصه ای از ویژگی های افراد بازمانده

در شکل 9 خلاصه ای از افراد نجات نیافته را مشاهده میکنید



شکل 12 - 268 نفر از مسافران نجات نیافتند

همچنین در این بخش میتوانیم ببینیم افراد نجات نیافته چه ویژگی هایی داشته اند یا مقادیر ویژگی های مختلف آنها در چه بازه ای یا دارای چه میانگینی بوده است که میتواند بسیار مفید باشد.



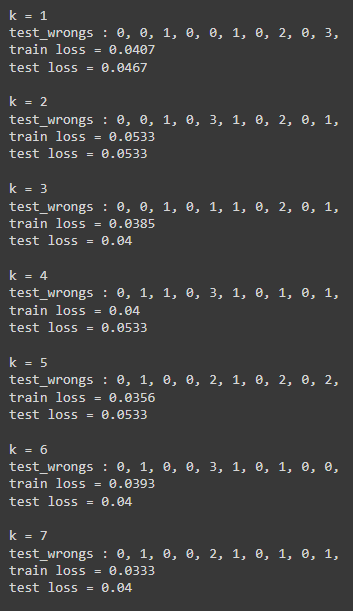
شکل 8 - خلاصه ای از ویژگی های افراد نجات نیافته

## سوال 3)

نوت بوک مربوطه به این تمرین :

<https://colab.research.google.com/drive/1oOhQ-PRDrJndli3CoxiARigyVg-83Wgv?usp=sharing>

**الف)** در این بخش با استفاده از Cross validation با 10 fold بهترین مقدار K را برای KNN پیدا میکنیم. روند کار به این صورت است که به ازای هر K ، 10 بار مدل را ارزیابی میکنیم به شکلی که هر بار یک دهم دیتاست را به عنوان داده تست و 9 دهم دیتاست را به عنوان داده ترین استفاده میکنیم. روند آموزش و ارزیابی مدل را در تصویر زیر مشاهده میکنید.



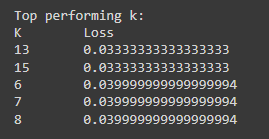
شکل 13 - روند آموزش مدل KNN

پس از اتمام آموزش نمودار K - avg\_loss را رسم میکنیم. در این نمودار محور افقی مقدارK ، منحنی آبی Training\_loss و منحنی نارنجی Validation\_loss میباشد. این نمودار به ما کمک میکند بهترین مقدار K را پیدا کنیم. بهترین مقدار K مقداری است که کمترین validation loss را داشته باشد.



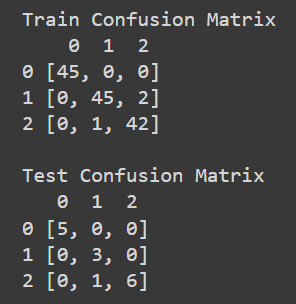
شکل 14 - نمودار K - avg\_loss برای پیدا کردن بهترین مقدار K

بهترین مقادیر K را به ترتیب در تصویر زیر مشاهده میکنید.



شکل 15- جدول بهترین مقادیر K

**ب)** در نهایت برای مقدار K=13 ماتریس درهم ریختگی (Confusion Matrix) را رسم میکنیم.



# منابع

تمرین پیاده سازی اول

<https://colab.research.google.com/drive/1hs3OGPk5St9bRO-bbWPr4TjBbgeGxlFn?usp=sharing>

تمرین پیاده سازی دوم

<https://colab.research.google.com/drive/1PQ_OsTzple1dd5EEmL0-kiUc9Ne5xfIR?usp=sharing>

تمرین پیاده سازی سوم

<https://colab.research.google.com/drive/1oOhQ-PRDrJndli3CoxiARigyVg-83Wgv?usp=sharing>

لینک گیتهاب

<https://github.com/Gholamrezadar/machine-learning-exercises>

Naïve Bayes

<https://www.machinelearningplus.com/predictive-modeling/how-naive-bayes-algorithm-works-with-example-and-full-code/#5naivebayesexamplebyhand>