**به نام خدا**

**دانشگاه صنعتی امیرکبیر**

**دانشکده مهندسی کامپیوتر**

# پاسخ تمرین سری دوم یادگیری ماشین

**استاد:**

**دکتر احسان ناظرفرد**

**دانشجو:**

**حلیمه رحیمی**

**شماره دانشجویی:**

**۹۹۱۳۱۰۴۳**

**پاییز ۱۳۹۹**

## سوالات تشریحی

**۱- برای یافتن بهترین مقدار پارامتر K در الگوریتم K-نزدیکترین همسایه چه راهکاری را پیشنهاد می کنید؟**

استفاده از Cross Validation گزینه ی مناسبی است. می توان با رسم نمودار K-Accuracy (یا هر معیار سنجش دیگری که مورد نظر شماست) بهترین مقدار را برای پارامتر K پیدا کرد. جایی از منحنی که مقدار دقت به بالاترین (یا کمترین بسته به معیار سنجش) حد خود برسد و با توجه به نیاز به بایاس یا واریانس بالاتر، K را انتخاب می کنیم.

**۲- با توجه به شکل زیر به سوالات پاسخ دهید.**

**الف) بهترین مقدار K برای الگوریتم K-نزدیکترین همسایه زمانی که از روش LOOCV استفاده می شود را محاسبه کنید. دقت الگوریتم را به ازای این K گزارش نمایید.**

نقاط را به ترتیب زیر نامگذاری کرده ام.

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | (1,5) | **1** | (2,6) | **2** | (2,7) | **3** | (3,7) | **4** | (3,8) | **5** | (4,8) | **6** | (5,9) |
| **7** | (5,1) | **8** | (6,2) | **9** | (7,2) | **10** | (7,3) | **11** | (8,3) | **12** | (8,4) | **13** | (9,5) |

برای سهولت در نوشتار، مجذور فاصله نقاط با یکدیگر ($d^2$ ) را در ماتریس زیر آورده ام.

$$\begin{bmatrix}
0 & 2 & 5 & 8 & 13 & 18 & 32 & 32 & 34 & 45 & 40 & 53 & 50 & 64 \\
2 & 0 & 1 & 2 & 5 & 8 & 18 & 34 & 32 & 41 & 34 & 45 & 40 & 50 \\
5 & 1 & 0 & 1 & 2 & 5 & 13 & 45 & 41 & 50 & 41 & 52 & 45 & 53 \\
8 & 2 & 1 & 0 & 1 & 2 & 8 & 40 & 34 & 41 & 32 & 41 & 34 & 40 \\
13 & 5 & 2 & 1 & 0 & 1 & 5 & 53 & 45 & 52 & 41 & 50 & 41 & 45 \\
18 & 8 & 5 & 2 & 1 & 0 & 1 & 50 & 53 & 45 & 52 & 41 & 32 & 34 \\
32 & 18 & 13 & 8 & 5 & 2 & 0 & 64 & 50 & 53 & 40 & 45 & 34 & 32 \\
32 & 34 & 45 & 40 & 53 & 50 & 64 & 0 & 2 & 5 & 8 & 13 & 18 & 32 \\
34 & 32 & 41 & 34 & 45 & 40 & 50 & 2 & 0 & 1 & 2 & 5 & 8 & 18 \\
45 & 41 & 50 & 41 & 52 & 45 & 53 & 5 & 1 & 0 & 1 & 2 & 5 & 13 \\
40 & 34 & 41 & 32 & 41 & 34 & 40 & 8 & 2 & 1 & 0 & 1 & 2 & 8 \\
53 & 45 & 52 & 41 & 50 & 41 & 45 & 13 & 5 & 2 & 1 & 0 & 1 & 5 \\
50 & 40 & 45 & 34 & 41 & 32 & 34 & 18 & 8 & 5 & 2 & 1 & 0 & 2 \\
64 & 50 & 53 & 40 & 45 & 34 & 32 & 32 & 18 & 13 & 8 & 5 & 2 & 0
\end{bmatrix}$$

در جدول زیر در هر ستون K، نقاط جدیدی که به نقطه ی مورد نظر نزدیکترند اضافه شده و کلاس داده با توجه به تمامی نقاط تا به آن ستون به دست آمده است.

| | | K | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Points** | **C** | **1** | | **3** | | **5** | | **7** | | **9** | | **11** | | **13** | | | | | | | | | | | | | |
| **0** | **-** | 1 | - | 2,3 | - | 4,5 | - | 6,7 | - | 8,10 | + | 9,12 | + | 11,13 | + | | | | | | | | | | | | |
| **1** | **-** | 2 | + | 0,3 | - | 4,5 | - | 6,8 | - | 7,10 | + | 12,9 | + | 11,13 | + | | | | | | | | | | | | |
| **2** | **+** | 1 | - | 3,4 | - | 0,5 | - | 6,8 | - | 10,7 | - | 12,9 | - | 11,13 | - | | | | | | | | | | | | |
| **3** | **-** | 2 | + | 4,1 | + | 5,0 | - | 6,10 | - | 8,12 | + | 7,13 | + | 9,11 | + | | | | | | | | | | | | |
| **4** | **+** | 3 | - | 5,2 | - | 1,6 | - | 0,10 | - | 12,8 | - | 13,11 | - | 9,7 | - | | | | | | | | | | | | |
| **5** | **-** | 4 | + | 3,6 | - | 2,1 | - | 0,12 | - | 10,13 | + | 8,11 | + | 9,7 | + | | | | | | | | | | | | |
| **6** | **-** | 5 | - | 4,3 | - | 2,1 | - | 0,13 | - | 12,10 | + | 11,8 | + | 9,7 | + | | | | | | | | | | | | |
| **7** | **+** | 8 | + | 9,10 | + | 11,12 | + | 0,13 | + | 1,3 | - | 2,5 | - | 4,6 | - | | | | | | | | | | | | |
| **8** | **+** | 9 | - | 7,10 | + | 11,12 | + | 13,1 | + | 0,3 | - | 5,2 | - | 4,6 | - | | | | | | | | | | | | |
| **9** | **-** | 8 | + | 10,11 | + | 7,12 | + | 13,1 | + | 3,0 | + | 5,2 | + | 4,6 | + | | | | | | | | | | | | |
| **10** | **+** | 9 | - | 11,8 | - | 12,7 | + | 13,3 | + | 1,5 | - | 0,6 | - | 2,4 | - | | | | | | | | | | | | |
| **11** | **-** | 10 | + | 12,9 | + | 8,13 | + | 7,3 | + | 5,1 | + | 6,4 | + | 2,0 | + | | | | | | | | | | | | |
| **12** | **+** | 11 | - | 10,13 | + | 9,8 | + | 7,5 | + | 3,6 | - | 1,4 | - | 2,0 | - | | | | | | | | | | | | |
| **13** | **+** | 12 | + | 11,10 | + | 9,8 | + | 6,7 | + | 5,3 | - | 4,1 | - | 2,0 | - | | | | | | | | | | | | |
| **Accuracy** | | **0.29** | | **0.57** | | **0.71** | | **0.71** | | **0** | | **0** | | **0** | | | | | | | | | | | | | |

همانطور که مشاهده می کنید، مقدار ۵ و ۷ برای K بیشترین دقت را بر می گردانند. ترجیح داده ام برای کاهش حساسیت به داده های پرت و نویز، مقدار ۷ را انتخاب کنم.

**ب) مشکل انتخاب مقدار K خیلی بزرگ و خیلی کوچک چیست؟ توضیح دهید.**

در صورت انتخاب مقدار K خیلی بزرگ، الگوریتم بایاس بالا و در صورت انتخاب مقدار خیلی کوچک، واریانس بالا می شود. بنابراین مقدار خیلی بزرگ موجب می شود یادگیرنده در مقابل داده های جدید واکنش چندانی نشان ندهد و دچار کم برازش شود. برعکس، مقدار خیلی کوچک یادگیرنده را حساس به داده های پرت و نویز می کند و موجب بیش برازش می گردد.

**ج) نقطه (2,1) با توجه به K به دست آمده به کدام کلاس تعلق می یابد؟**

مجذور فاصله این نقطه به هر یک از نقاط را در جدول زیر می بینید.

| **Points** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $d^2$ | 17 | 25 | 36 | 37 | 50 | 53 | 73 | 9 | 17 | 26 | 29 | 40 | 45 | 65 |

با توجه به ۷-نزدیکترین همسایه، این داده جزو کلاس مثبت محسوب می شود. نزدیکترین همسایه ها به این نقطه به ترتیب ۷، ۰، ۸، ۱، ۹، ۱۰، ۲ می باشد که از این تعداد، چهار تایشان از کلاس مثبت بوده است.

**۳- هر کدام از الگوریتم های K-نزدیکترین همسایه و درخت تصمیم را از نظر پارامتریک و غیرپارامتریک بودن بررسی کنید.**

هر دو الگوریتم غیرپارامتریک محسوب می شوند؛ چرا که هیچ یک پارامتری مرتبط با توزیع داده برای یادگیری ندارد. در الگوریتم پارامتریک درصدد یادگیری تعداد بخصوصی از پارامترها هستیم و آمدن داده ی جدید، این تعداد را تغییر نمی دهد.

در K-نزدیکترین همسایه یادگیرنده به فاصله ی بین داده ها توجه می کند و در درخت تصمیم نیز توجهی به خصوصیات توزیع داده نداریم و با توجه به مقدار ویژگی ها به یادگیری می پردازیم. به عبارتی دیگر پارامتری در مورد مجموعه داده ها برای یادگیری نداریم.

**۴- هر کدام از الگوریتم های K-نزدیکترین همسایه و درخت تصمیم را از Generative و Discriminative بودن بررسی کنید.**

هر دو از نوع Discriminative می باشند. در مدل های از نوع Generative به دنبال مدل کردن توزیع هر یک از کلاس ها و استفاده از احتمال مشترک داده و کلاس هستیم، در حالیکه در Discriminative به یادگیری مرز تصمیم و استفاده از احتمال شرطی می پردازیم.

هر دو الگوریتم K-نزدیکترین همسایه و درخت تصمیم با تعیین مرز تصمیم به یادگیری می پردازند. حال آنکه برای مثال، دسته بند بیزین با مدل کردن توزیع کلاس ها و به دست آوردن احتمال مشترک داده و کلاس، آموزش می بینند.

**۵- به چه الگوریتم هایی تنبل گفته می شود؟ K-نزدیکترین همسایه تنبل است یا خیر؟ توضیح دهید.**

الگوریتم هایی که در زمان آموزش یادگیری انجام نمی دهند و تمام بار آموزش به مرحله ی آزمون منتقل می شود. در اینگونه الگوریتم ها پیچیدگی محاسبات در زمان آزمون بیشتر از زمان آموزش می باشد.

الگوریتم K-نزدیکترین همسایه در آموزش، هیچ محاسباتی انجام نمی دهد و یادگیری در آن انجام نمی شود ولی در زمان آزمون محاسبات لازم را به انجام می رساند. بنابراین از نوع الگوریتم های تنبل می باشد.

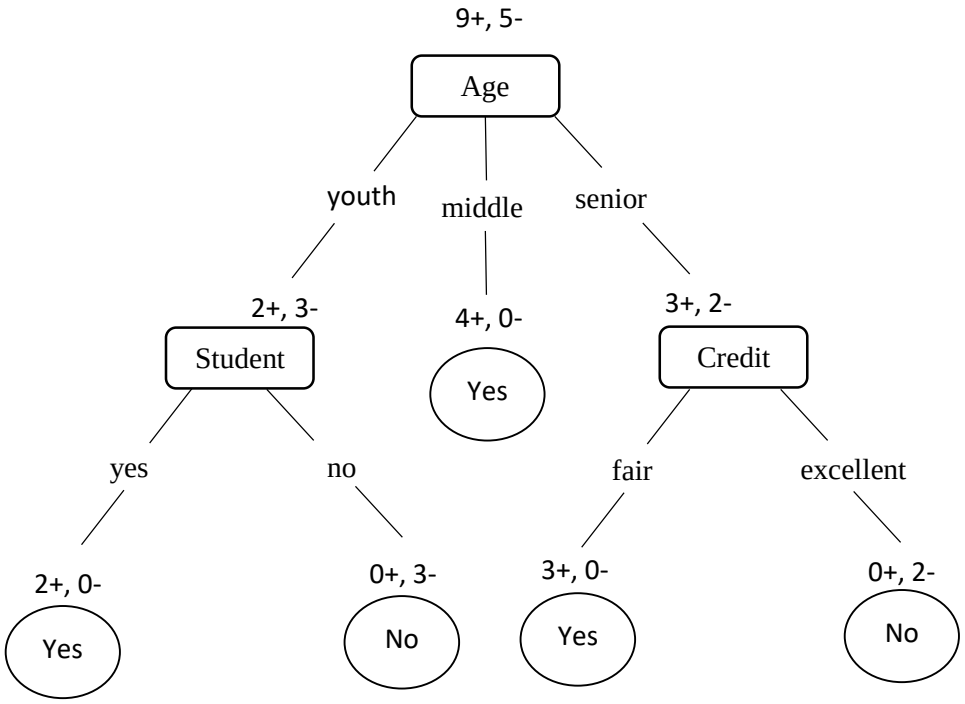
**۶- هرس درخت تصمیم چه تاثیری بر بیش برازش دارد؟ این هرس چه زمانی باید انجام شود؟ توضیح دهید.**

یادگیرنده با عمق کم، بایاس بالا عمل می کند و در مقابل داده جدید تغییر چندانی در مرز تصمیم ایجاد نمی شود. بنابراین با هرس و کمتر شدن عمق درخت، مشکل بیش برازش از بین می رود. این هرس کردن باید به شکلی باشد که موجب کم برازش شدن مدل نشود. به عبارتی هرس نباید معیار کارایی را کاهش قابل محسوسی بدهد.

انواع مختلفی از هرس وجود دارد. می توان هرس را در هنگام پیش روی و با رسیدن به عمق بخصوص یا بهره اطلاعات مورد نظر به انجام رساند (Prepruning). روش کلی دیگر این است که درخت را کامل کنیم و سپس شروع به هرس کردن کنیم و شاخه ها را حذف و با برگ جایگزین کنیم (Postpruning).

**۷- در جدول ۱ مجموعه داده ای نمایش داده شده است که در آن افراد با توجه به ویژگی هایی مثل سن، درآمد و... اقدام به خرید یا عدم خرید یک کالا کرده اند. هدف ما تخمین این است که آیا فرد موردنظر قصد خرید کالا را دارد یا خیر.**

**الف) با توجه به ویژگی آنتروپی و بهره اطلاعات درخت تصمیم بهینه را برای این مجموعه داده بیابید.**




$$\text{Gain}(S, \text{Age}) = 0.94 - \left(\frac{5}{14} \times 0.97 + \frac{4}{14} \times 0 + \frac{5}{14} \times 0.97\right) = 0.25$$

$$\text{Gain}(S, \text{Income}) = 0.94 - \left(\frac{4}{14} \times 0.81 + \frac{6}{14} \times 0.91 + \frac{4}{14} \times 1\right) = 0.03$$

$$\text{Gain}(S, \text{Student}) = 0.94 - \left(\frac{7}{14} \times 0.59 + \frac{7}{14} \times 0.98\right) = 0.15$$

$$\text{Gain}(S, \text{Credit}) = 0.94 - \left(\frac{8}{14} \times 0.81 + \frac{6}{14} \times 1\right) = 0.05$$

با توجه به مقادیر به دست آمده برای بهره اطلاعات، ویژگی Age را در ریشه درخت گذاشته ام.

پس از این برای شاخه با مقدار youth بهره اطلاعات را به دست می آوریم.

$$\text{Gain}(youth, \text{Income}) = 0.97 - \left(\frac{1}{5} \times 0 + \frac{2}{5} \times 1 + \frac{2}{5} \times 0\right) = 0.57$$

$$\text{Gain}(youth, \text{Student}) = 0.97 - \left(\frac{2}{5} \times 0 + \frac{3}{5} \times 0\right) = 0.97$$

$$\text{Gain}(youth, \text{Credit}) = 0.97 - \left(\frac{3}{5} \times 0.91 + \frac{2}{5} \times 1\right) = 0.02$$

بنابراین این بار ویژگی Student در گره قرار می گیرد.

همچنین برای شاخه با مقدار senior داریم:

$$\text{Gain}(senior, \text{Income}) = 0.97 - \left(\frac{2}{5} \times 1 + \frac{3}{5} \times 0.91 + \frac{0}{5} \times 0\right) = 0.02$$

$$\text{Gain}(senior, \text{Student}) = 0.97 - \left(\frac{3}{5} \times 0.91 + \frac{2}{5} \times 1\right) = 0.02$$

$$\text{Gain}(senior, \text{Credit}) = 0.97 - \left(\frac{3}{5} \times 0 + \frac{2}{5} \times 0\right) = 0.97$$

در اینجا ویژگی Credit بیشترین بهره اطلاعات را داشته و در نتیجه به عنوان گره بعدی انتخاب می شود.

از تصویر کاملا مشهود است که داده ها پیش از انتخاب ویژگی Income کاملا از یکدیگر جدا شده اند.

**ب) داده های زیر را با توجه به درخت تصمیم به دست آمده، دسته بندی کنید.**

**$X_1$ = (age = youth, income = high, student = yes, credit = fair)**

**$X_2$ = (age = senior, income = low, student = no, credit = excellent)**

**$X_3$ = (age = middle-aged, income = medium, student = no, credit = fair)**

با توجه به درخت تصمیم، $X_1$ و $X_3$ جزو کلاس مثبت و $X_2$ جزو کلاس منفی می باشد.

**۸- در دسته بندی داده ها با درخت تصمیم می توانیم با افزایش عمق درخت توابع بسیار پیچیده ای بسازیم. آیا برای مسائل جداپذیر خطی مانند شکل زیر می توان دسته بندی را با عمق محدود درخت انجام داد؟**

خیر، مرز تصمیم حاصل از درخت های تصمیم غیرخطی می باشد. بنابراین برای تشکیل چنین خطی نیاز به عمق بالایی داریم تا در نهایت مجموع خطوط حاصل از هر گره، خطی تقریبا مشابه این خط به ما بدهد.

**۹- الگوریتم جنگل تصادفی را مختصر توضیح دهید. چرا این الگوریتم با محدود نگه داشتن عمق درخت می تواند داده ها را جدا کند؟**

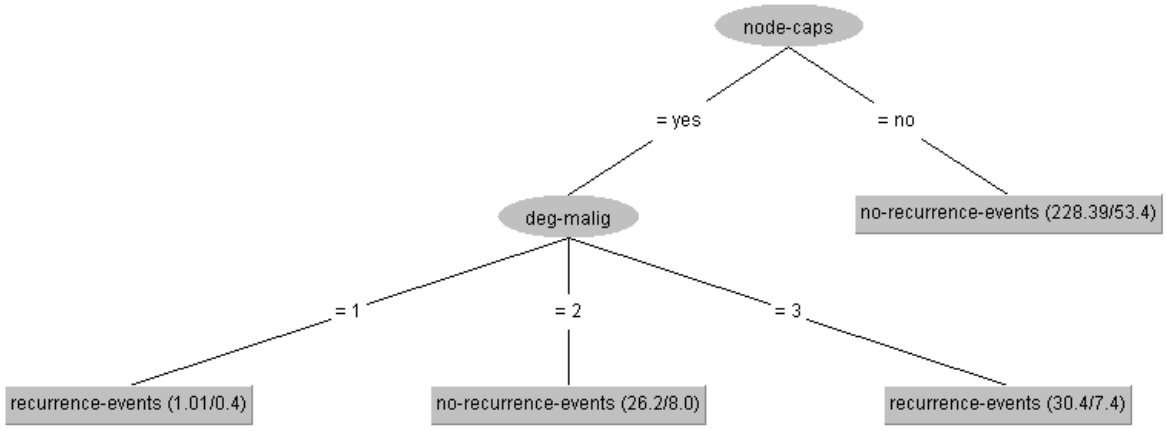
الگوریتم جنگل تصادفی از نوع یادگیری گروهی (Ensemble Learning) می باشد. این الگوریتم شامل چند درخت تصمیم می باشد که برای دسته بندی، از آنها رای گیری به عمل می آید. در اینجا با استفاده از bagging به تقسیم بندی تصادفی داده ها با جایگذاری به ایجاد مجموعه داده های مختلف برای هر یک از این درخت های تصمیم پرداخته می شود.

با توجه به آنکه ویژگی هایی که در هر عمق درخت تصمیم انتخاب می شود بر اساس داده هاست، می توان نتیجه گرفت درخت ها با یکدیگر متفاوت خواهند بود. اگر ویژگی بخصوصی اثر بسیاری بر تصمیم گیری می گذارد، مطمئنا در بیشتر درخت ها در اوایل آنها انتخاب خواهد شد.

با این توضیح می توان فهمید که در هر درخت تاثیرگذارترین ویژگی ها در ابتدا برگزیده می شوند و از آن جهت که مجموعه داده های هر یک از این درخت ها با هم متفاوت خواهند بود، طبیعتا ویژگی ها ترتیب متفاوتی خواهند داشت. بنابراین با محدود نگه داشتن عمق درخت، بایاس در مقایسه با یک درخت عمیق کلی آنچنان افزایش پیدا نخواهد کرد؛ در حالیکه دقت آن (نسبت به هرس در یک درخت عمیق کلی) افزایش خواهد داشت.

**WEKA**

**۱- مجموعه داده ی پیوست شده به نام breast-cancer.arff را بارگذاری کرده و با استفاده از درخت تصمیم (J48) داده ها را دسته بندی کنید.**

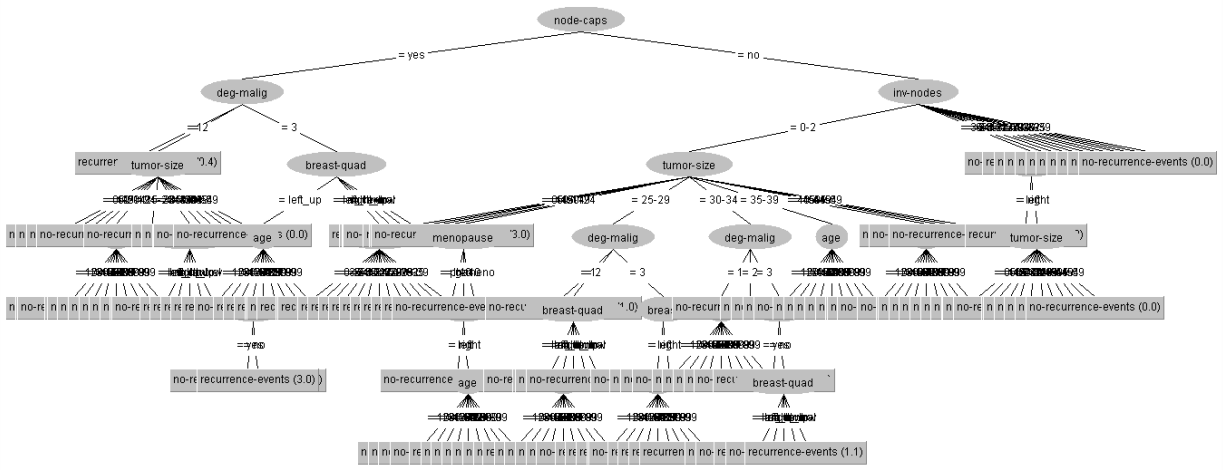



درخت با اندازه ی ۶ شامل ۴ برگ می باشد. مدل این درخت تحت عنوان model1 در میان فایل های ارسالی ثبت شده است. مقدار اعشاری در برگ ها به علت مقادیر نامعلوم می باشد.

| | TP | FP | FN | Precision | Recall | F1measure |
|---|---|---|---|---|---|---|
| **Recurrence-events** | 0.281 | 0.148 | 0.719 | 0.529 | 0.281 | 0.367 |
| **Weighted Average** | 0.640 | 0.506 | 0.360 | 0.616 | 0.640 | 0.606 |

| **Recurrence-events: +**<br>**Non-recurrence-events: -** | | **Predicted Class** | |
|---|---|---|---|
| | | **+** | **-** |
| **Actual Class** | **+** | **9** | **23** |
| | **-** | **8** | **46** |

**۲- پارامتر unpruned چه چیزی را کنترل می کند؟ این مقدار را از false به true تغییر داده و دوباره موارد قسمت ۱ را انجام دهید. درخت آموزش داده شده در این قسمت چه تفاوتی با قسمت ۱ دارد؟ توضیح دهید.**




با پارامتر unpruned هرس کردن درخت کنترل می شود. با قرار دادن آن در حالت True هرس انجام نمی گیرد. در این درخت ویژگی ها چندین بار در شاخه های مختلف استفاده شده اند. همچنین بسیاری از برگ های درخت خالی اند.

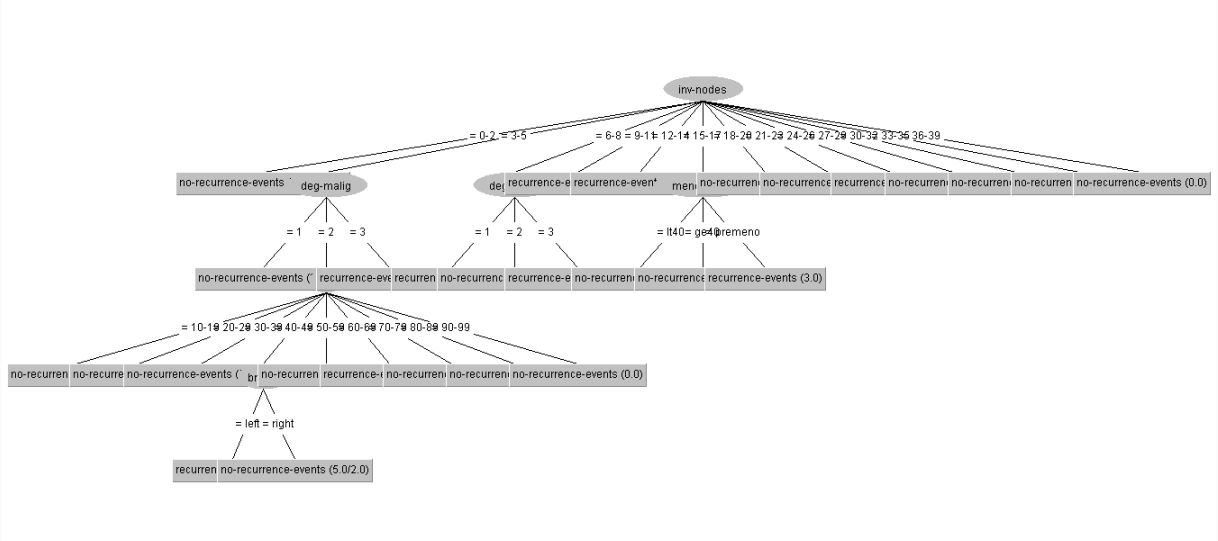
درخت حجیم بوده و با اندازه ی ۱۷۹ شامل ۱۵۲ برگ می باشد. در تصویر به خوبی این درخت مشخص نمی شود. مدل مربوطه را می توان تحت عنوان model2 در میان فایل های ارسالی اینجانب یافت.

| | TP Rate | FP Rate | FN Rate | Precision | Recall | F1measure |
|---|---|---|---|---|---|---|
| **Recurrence-events** | 0.375 | 0.241 | 0.625 | 0.480 | 0.375 | 0.421 |
| **Weighted Average** | 0.616 | 0.482 | 0.384 | 0.601 | 0.616 | 0.604 |

| **Recurrence-events: +**<br>**Non-recurrence-events: -** | | **Predicted Class** | |
|---|---|---|---|
| | | **+** | **-** |
| **Actual Class** | **+** | **12** | **20** |
| | **-** | **13** | **41** |

**۳- حال موارد قسمت ۱ و ۲ را بار دیگر، این بار با اعمال ۱۵ درصد نویز به ریشه ی درخت های دو قسمت قبل تکرار کنید و نتایج حاصل را با نتایج مراحل قبلی مقایسه کرده و بررسی کنید که هرس کردن درخت چه تاثیری در برخورد با نویز دارد.**

در دو قسمت قبل، ویژگی node-caps در ریشه قرار می گرفت. با اعمال نویز به این ویژگی، درخت هرس شده ی زیر حاصل شد. مدل این درخت تحت عنوان model3 در میان فایل های ارسالی ثبت شده است.

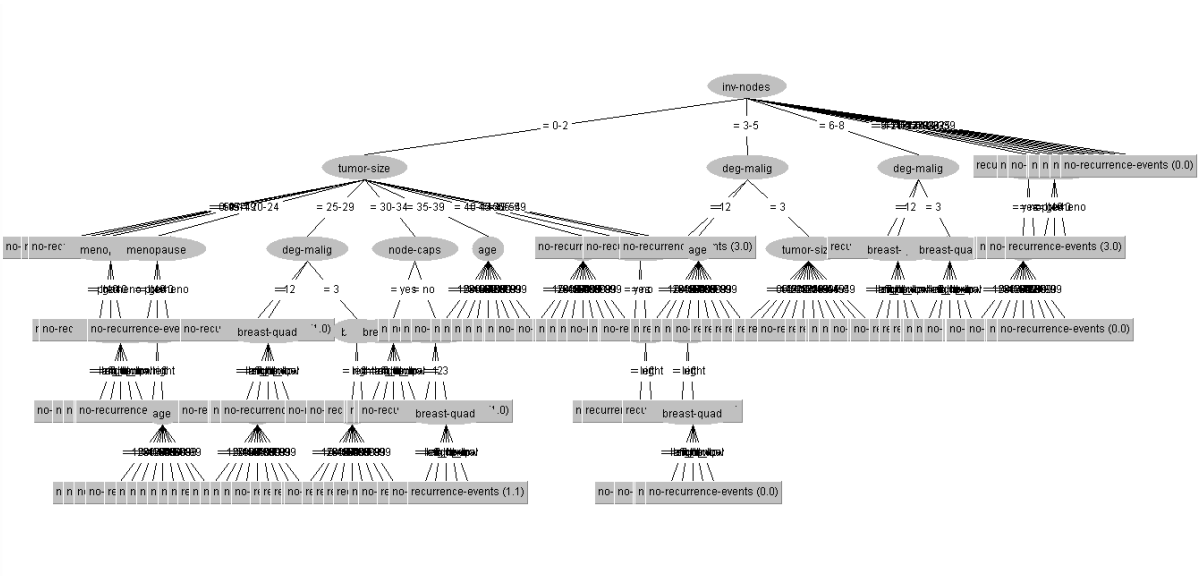



این بار در ریشه inv-nodes قرار گرفته است. درخت با اندازه ۳۴ شامل ۲۸ برگ می باشد.

| | TP | FP | FN | Precision | Recall | F1measure |
|---|---|---|---|---|---|---|
| **Recurrence-events** | 0.281 | 0.130 | 0.719 | 0.563 | 0.281 | 0.375 |
| **Weighted Average** | 0.651 | 0.500 | 0.349 | 0.631 | 0.651 | 0.616 |

| **Recurrence-events: +<br>Non-recurrence-events: -** | | **Predicted Class** | |
|---|---|---|---|
| | | **+** | **-** |
| **Actual Class** | **+** | **9** | **23** |
| | **-** | **7** | **47** |

درخت در حالت بدون هرس، با اندازه ۱۷۱ شامل ۱۴۰ برگ می باشد. در زیر این درخت حجیم را مشاهده می کنید. مدل این درخت تحت عنوان model4 در میان فایل های ارسالی به ثبت رسیده است.




| | TP | FP | FN | Precision | Recall | F1measure |
|---|---|---|---|---|---|---|
| **Recurrence-events** | 0.406 | 0.278 | 0.594 | 0.464 | 0.406 | 0.433 |
| **Weighted Average** | 0.605 | 0.476 | 0.395 | 0.595 | 0.605 | 0.599 |

| **Recurrence-events: +**<br>**Non-recurrence-events: -** | | **Predicted Class** | |
|---|---|---|---|
| | | **+** | **-** |
| **Actual Class** | **+** | **13** | **19** |
| | **-** | **15** | **39** |

با مقایسه حالت بدون نویز و با نویز برای درخت هرس شده، متوجه می شویم که با وجود نویز، الگوریتم همچنان مقادیر برابر یا بهتری برای معیارهای سنجش مختلف دارد.

درحالیکه برای درخت هرس نشده، در برخی موارد بهتر و در برخی بدتر عمل کرده است.

این مسائل نشاندهنده ی آن است که درخت هرس شده به علت عمق کمتر، بایاس بیشتری داشته و در برابر نویز مقاومت بیشتری دارد. درحالیکه درخت هرس نشده با هر نویزی، به راحتی نتایج را تغییر می دهد و واریانس بالایی دارد.

آنچه مسلم است بایاس بودن دیتاست به سمت کلاس منفی است. بنابراین طبیعی است که الگوریتم روی داده های مثبت به خوبی داده های منفی عمل نمی کند.

## سوالات پیاده سازی

**۱- یک تابع بنویسید که با گرفتن ورودی های مجموعه داده، K و معیار فاصله، الگوریتم KNN را اجرا کند. از آن تابع برای دسته بندی مجموعه داده ی mammographic_masses.data استفاده کنید.**

این الگوریتم با استفاده از فاصله، کلاس را پیشبینی می کند، بنابراین از آن جهت که واحد های متفاوت بر فاصله اثرگذارند، داده ها به نرمالیزیشن نیاز دارند.

برای آماده سازی داده، در ابتدا این روش را امتحان کردم که با توجه به مثبت بودن ضریب همبستگی ویژگی ها با کلاس، در جاهایی که کلاس مثبت داشتیم، ماکسیموم مقدار آن ویژگی و در جاهایی که کلاس منفی داشتیم مینیموم مقدار را گذاشتم.

اشکال این روش این بود که مقادیر ویژگی داده ها متناسب با کلاس تنظیم می شد و اجازه تصادفی بودن به مقادیر نمی داد.

سپس از جاگذاری مقدار ویژگی داده ی قبلی یا بعدی به عنوان مقدار ویژگی استفاده کردم. به علت کم بودن تعداد مقادیر خالی و کوچک بودن فاصله اعداد، اثر این حالت با حالت قبلی مشابه یکدیگر بود.

در نهایت بهترین نتیجه را در حالتی داشتم که نمونه های با مقادیر خالی را حذف کردم. به نظر می رسد این مسئله به علت تعداد کم مقادیر نامعلوم نسبت به کل مجموعه داده باشد. نتایجی که آورده ام مربوط به این حالت است.

برای به دست آوردن ماتریس درهم ریختگی از تابع نوشته شده توسط خودم استفاده کرده و برای K-fold Cross Validation از تابع آماده کتابخانه SKLearn بهره برده ام. هنگام استفاده از K-Fold مقدار شافل را در ابتدا برابر False گذاشتم، سپس آن را به True تغییر دادم. نتایج را در پایین می توانید ببینید.

با شافل کردن، دقت برای مقادیر کم K بخصوص مقدار ۱ اثر بیشتری می گذارد درحالیکه ممکن است از ۷ به بعد تغییر کوچکی ایجاد شود که دقت را کم یا زیاد کند. علت این مسئله کاملا مشخص است؛ K کوچکتر به معنای واریانس بیشتر می باشد.

**الف) الگوریتم KNN به ازای مقادیر مختلف ۱،۳،۵،۷،۱۵،۳۰ برای K و با فاصله اقلیدسی اجرا کرده و تأثیر مقادیر مختلف K را تحلیل کنید.**

**(خروجی موردنظر: دقت الگوریتم و جدول درهم ریختگی)**

برای اطمینان از آنچه در مورد شافل کردن گفتم، در زیر مقادیر دقت را برای حالت False شافل آورده ام.

| K | 1 | 3 | 5 | 7 | 15 | 30 |
|---|---|---|---|---|---|---|
| **Accuracy** | 0.73614458 | 0.76385542 | 0.79518072 | 0.80240964 | 0.79156627 | 0.80481928 |

برای این حالت، بهترین مقدار K برابر با ۳۰ شد.

حال مقادیر دقت را برای حالت True شافل می بینید.

| K | 1 | 3 | 5 | 7 | 15 | 30 |
|---|---|---|---|---|---|---|
| **Accuracy** | 0.76144578 | 0.7686747 | 0.79879518 | 0.80722892 | 0.80120482 | 0.80240964 |

مقدار K در هر بار اجرا بسته به داده ها ممکن است تغییر کند. در اینجا برابر با ۷ شده است.

از این پس نتایج بخش های دیگر را به دلیل ثبات در نتایج ذکر شده در گزارش با حالت False شافل مقایسه خواهم کرد.

**ب) به ازای بهترین مقدار K که در قسمت الف یافته اید و با فاصله های اقلیدسی، منهتن و کسینوسی، الگوریتم را اجرا کنید.**

**(خروجی موردنظر: دقت الگوریتم و جدول درهم ریختگی)**

| Distance Criterion | Euclidian | Cosine | Manhattan |
|---|---|---|---|
| **Accuracy** | 0.80481928 | 0.51445783 | 0.79638554 |

به نظر می رسد معیار فاصله بسته به مجموعه داده انتخاب می شود. در اینجا فاصله کسینوسی برای این مجموعه داده معیار مناسبی نیست و کمترین مقدار دقت را می دهد.

**ج) با استفاده از کتابخانه های آماده و به ازای مقادیر مختلف ۱،۳،۵،۷،۱۵،۳۰ برای K و فاصله اقلیدسی، داده ها را دسته بندی کرده و پیاده سازی خود را از نظر دقت و سرعت با این کتابخانه مقایسه کنید.**

**(خروجی موردنظر: دقت الگوریتم، جدول درهم ریختگی و زمان اجرای هر الگوریتم)**

| K | 1 | 3 | 5 | 7 | 15 | 30 |
|---|---|---|---|---|---|---|
| **Accuracy** | 0.73975904 | 0.76024096 | 0.79879518 | 0.8 | 0.79277108 | 0.8060241 |

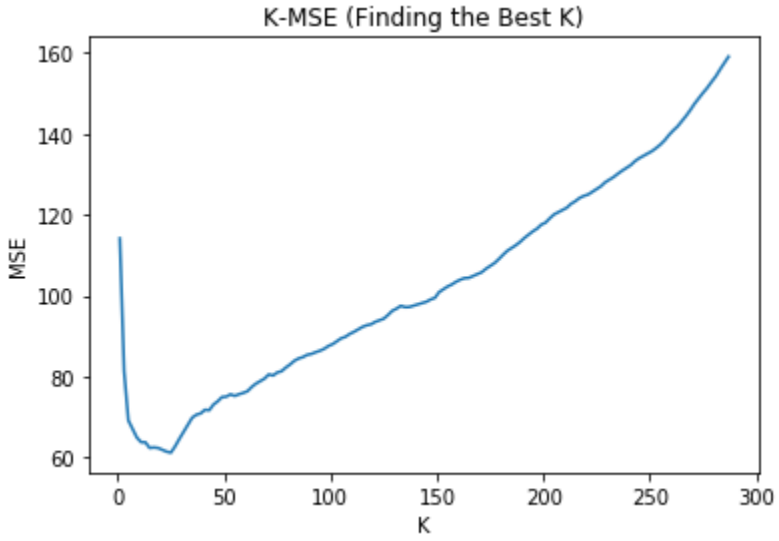
از نظر دقت، تفاوت در حدود ۰/۰۰۳ می باشد. از نظر سرعت، کتابخانه SKLearn بسیار سریعتر عمل می کند و مقداری حدود ۱۰ برابر دارد.

| Library | SKLearn | - |
|---|---|---|
| **Time** | 0.003 | 0.02 |

**۲- در این قسمت الگوریتم KNN برای رگرسیون استفاده نمایید. مجموعه داده ی regression.xlsx را با نسبت ۷۰ به ۳۰ تقسیم کرده و سپس بهترین مقدار برای K را با آزمون و خطا بیابید. خطای MSE را برای این مدل برای هر دو مجموعه ی داده آزمون و آموزش گزارش کنید.**

**(خروجی موردنظر: خطای MSE برای هر دو مجموعه آزمون و آموزش)**

از آن جهت که MSE خواسته شده و با شافل کردن داده این مقدار به شدت تغییر می کرد. این بخش را بدون شافل کردن انجام داده ام.




با در نظر گرفتن ۷۰ درصد اول داده به عنوان آموزش و ۳۰ درصد باقی مانده، مقدار K برابر با ۲۵ و MSE مطابق زیر می باشد.

| Dataset | Training | Test |
|---|---|---|
| **MSE** | 70.42133370242215 | 61.20864960000001 |

با توجه به اینکه برای رگرسیون با این الگوریتم از مقادیر همسایه ها میانگین می گیریم، به همین علت مقدار MSE برای داده های آموزش برابر با صفر نخواهد بود.

به صورت شافل شده نیز این الگوریتم را اجرا کردم. گاهی با توجه به مجموعه داده ها مقدار MSE برای مجموعه آزمون به ۴۰ می رسید.

**۳- درخت تصمیم بهینه را با استفاده از کتابخانه های آماده برای مجموعه داده ی breas-cancer-wisconsin-train.data آموزش داده و مجموعه داده ی breast-cancer-wisconsin-test.data را دسته بندی کنید.**

**(خروجی موردنظر: دقت الگوریتم برای مجموعه آزمون و جدول درهم ریختگی)**

در صورتی که از بهره اطلاعاتی استفاده می شد حتما نیاز بود ستون Sample Code Number را حذف کرد. در واقع این ستون آنچنان عدم قطعیت بالایی دارد که برای تشخیص، هر کدام از نمونه ها یک شاخه در درخت را تشکیل می دهند. در اینجا چنین کاری نیاز نیست.

در ابتدا مقادیر خالی را با مقدار داده ی بعدی برای همان ویژگی پر کردم. نتیجه را در زیر می توانید ببینید. در این حالت، دقت برابر با ٪۹۳ شد. این مقدار بین ۹۰ تا ۹۴ درصد متغیر بود.

| Malignant: +<br>Benign: - | | Predicted Class | |
|---|---|---|---|
| | | + | - |
| Actual Class | + | 49 | 11 |
| | - | 3 | 137 |

جدول درهم ریختگی در هر بار اجرای برنامه تغییر چندانی نمی کند. هر بار کلاس مثبت تعداد اشتباه در همین حدود دارد. با توجه به آنکه موضوع سرطان سینه می باشد، اشتباه کمتر در تشخیص کلاس مثبت بسیار مهم تر از تشخیص در کلاس منفی می باشد.

بنابراین تصمیم گرفتم از همان روشی که در ابتدای این بخش ارائه کردم استفاده کنم. نتیجه این بود که برای کلاس مثبت تعداد اشتباهات کمتر شد و برای کلاس منفی کمی بیشتر. در کل، دقت در تمامی اجراهای بنده در غالب موارد بیشتر و گاه مساوی با دقت حالت پیشین بود. در اینجا مقدار دقت برابر با ٪۹۴ شده و در کل در بین ۹۱ تا ۹۵ درصد متغیر بود.

| Malignant: +<br>Benign: - | | Predicted Class | |
|---|---|---|---|
| | | + | - |
| Actual Class | + | 53 | 7 |
| | - | 5 | 135 |

در نهایت عملی را انتخاب کردم که با دقت ۹۴/۵ درصد یکی از بهترین نتایج را داشت: حذف کردن داده های شامل مقادیر خالی. این مورد در بین ۹۳ تا ۹۶ درصد متغیر بود.

با این وجود تصمیم گرفتم نتایج حالت های دیگر را نیز ذکر کنم. نتیجه ی حذف مقادیر نامعلوم را با دقت ۹۴/۵ درصد در زیر می بینید.

| Malignant: +<br>Benign: - | | Predicted Class | |
|---|---|---|---|
| | | + | - |
| Actual Class | + | 56 | 4 |
| | - | 7 | 133 |