Subject :
Date

یادگیری ماشین تمرین سوم مهدی جمال خواه 810100111

سوال ① :

1 $N=5$: $P = \sum_{i=\frac{N+1}{2}}^{N} \binom{N}{i} \left(\frac{7}{10}\right)^i \left(\frac{3}{10}\right)^{N-i}$

$= \sum_{i=3}^{5} \binom{5}{i} \left(\frac{7}{10}\right)^i \left(\frac{3}{10}\right)^{5-i} = \boxed{0.83}$

2 $N=9$: $P = \sum_{i=5}^{9} \binom{9}{i} \left(\frac{7}{10}\right)^i \left(\frac{3}{10}\right)^{9-i} = \boxed{0.90}$

3 $N \to \infty$: $P = \lim_{N\to\infty} \sum_{i=\frac{N+1}{2}}^{N} \binom{N}{i} q^i (1-q)^{N-i}$

$= \lim_{N\to\infty} (1-q)^N \sum_{i=\frac{N+1}{2}}^{N} \binom{N}{i} \left(\frac{q}{1-q}\right)^i$

* q احتمال درست تشخیص دادن است

$q = \frac{7}{10} \Rightarrow P = \lim_{N\to\infty} (0.3)^N \sum_{i=\frac{N+1}{2}}^{N} \binom{N}{i} \left(\frac{7}{3}\right)^i = \boxed{1}$

به احتمال یک رسیدیم. در واقعیت نمی توان به این دقت رسید چون فرض مستقل بودن نظر هر شخص از دیگران فرض بزرگی است و معمولاً در واقعیت برقرار نیست.

Subject:
Date

4 $q = \frac{1}{2}$, $N = 5$ $\Rightarrow$ $\sum_{i=3}^{5} \binom{5}{i} \left(\frac{1}{2}\right)^5 = \boxed{\frac{1}{2}}$

$$\lim_{N \to \infty} \sum_{i=\frac{N+1}{2}}^{N} \binom{N}{i} \left(\frac{1}{2}\right)^N = \lim_{N \to \infty} \left(\frac{1}{2}\right)^N \underbrace{\sum_{i=\frac{N+1}{2}}^{N} \binom{N}{i}}_{2^{N-1}} = \lim_{N \to \infty} \frac{2^{N-1}}{2^N} = \boxed{\frac{1}{2}}$$

البته بنابراین به محاسبه حالت $N \to \infty$ هم نیست زیرا همان $N=5$ که احتمال تغییری نمی‌کند نشان می‌دهد که با افزایش تعداد اعضا هیچ بهبودی حاصل نمی‌شود. منظور آن هم مشخص است، هیچ شخص اطلاعات اضافه‌ای ندارد و نظر او کاملاً رندوم است. زمانی که هر کس اندکی نظر را بهبود بیخشد (احتمال بیش از $\frac{1}{2}$) درنهایت آن احتمال بیش‌تر می‌شود اما اکنون دقیقاً $\frac{1}{2}$ است و ما تنها یک classifier رندوم عمل می‌کند که هیچ اطلاعات اضافه‌ای به ما نمی‌دهد.

Subject:
Date

سوال ②:

الف) ① در این فضا نمی توانیم طبقه بند Decision Stump ای ایجاد کنیم که دقت

آن بیش از 50% باشد زیرا در هر بعد 3 حالت زیر را داریم که دقت همه 50% است.

−	+	−	+	−	+
+	−	+	−	+	−

acc = 50% acc = 50% acc = 50%

در نتیجه طبقه بند ضعیف ما هیچ اطلاع اضافه‌ای به ما نمی دهد و هیچ وقت دقت را

بهبود نمی دهد در فرمول AdaBoost نیز داریم:

$$\alpha = \frac{1}{2} \ln \frac{1-\varepsilon}{\varepsilon} \xrightarrow{\varepsilon=\frac{1}{2}} \alpha = 0$$

یعنی طبقه‌بند با دقت 50% را حذف می‌کند پس تمام طبقه‌بندها حذف می‌شوند و

نتیجه نهایی همان طبقه‌بند رندوم می‌باشد که دقت آن همان 50% است.

Sr Subject:
Date

$$E = \sum_{i=1}^{N} \omega_i^{(t)} \, e^{-y_i \alpha_t h_t(x_i)}$$

(2)

$$= \sum_{i \in \text{Correct}} \omega_i^{(t)} \, e^{-\alpha_t} + \sum_{i \in \text{Miss}} \omega_i^{(t)} \, e^{\alpha_t}$$

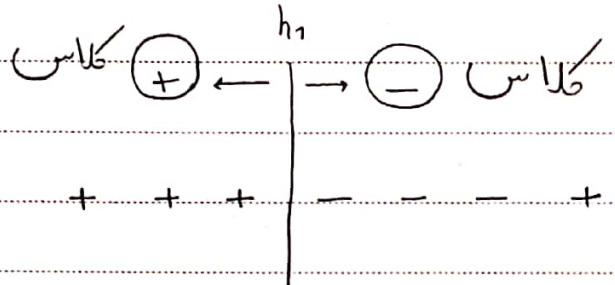
$$= (1 - \varepsilon_t) \, e^{-\alpha_t} + \varepsilon_t \, e^{\alpha_t} \quad ??$$

P4PCO

Subject:
Date

(3)

(ب)



Decision stump از بین ۸ نقطه بین داده ها و اطراف آن ها نقطه ای را انتخاب می کند که بیشترین دقت را داشته باشد زیرا در ابتدا وزن همه ی داده ها مساوی است و در حالت بالا فقط یک داده اشتباه پیش بینی شده است و هر کدام از نقاط دیگر را هم انتخاب کنیم نمی توانیم همه ی داده ها را به درستی پیش بینی کنیم پس h_1 انتخاب می شود.

(4)

$$\varepsilon_1 = \frac{\sum_{i \in M} w_i^{(0)}}{\sum_{i=1}^{n} w_i^{(0)}} = \frac{1/7}{1} = \frac{1}{7} = \boxed{0.14}$$

$$\alpha_1 = \frac{1}{2} \ln \frac{1-\varepsilon_1}{\varepsilon_1} = \frac{1}{2} \ln 6 = \boxed{0.89}$$

$$\text{accuracy} = \frac{6}{7} \times 100 = \boxed{86\%}$$

(5)

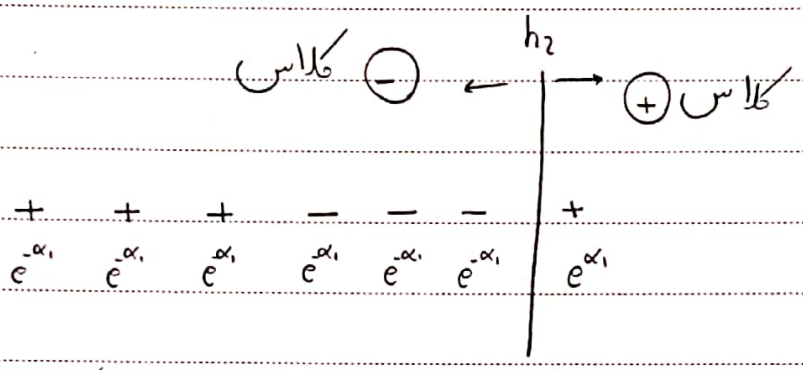
نقاط را از ست چپ از ۱ تا ۷ شماره گذاری می کنیم.

Subject:
Date

$$\omega_i^{(1)} = e^{-y_i F_i(x_i)} = e^{-y_i \alpha_1 h_1(x_i)}$$

$$\omega_i^{(1)} = \begin{cases} e^{-\alpha_1} = e^{-0.89} & i = 1, \dots, 6 \\ e^{\alpha_1} = e^{0.89} & i = 7 \end{cases}$$

(6)



مثل قبل نقطه ای انتخاب می شود که دقتا h_2 بیش تر باشد که مثل بالایی شود البته

در بدست آوردن دقتا وزن نمونه ها مهم است و باید میانگین وزن دار گرفته شود

ازین 8 نقطه برای طبقه بند 2 نقطه آن است که نیاز به محاسبات دارد و مابقی بدیهی است

که نمی تواند باشد. 3 و 4 بین $accuracy = \frac{6e^{-\alpha_1}}{6e^{-\alpha_1} + e^{\alpha_1}} = 50\%$

7 و 6 بین $accuracy = \frac{3e^{-\alpha_1} + e^{\alpha_1}}{6e^{-\alpha_1} + e^{\alpha_1}} = 75\%$ ✓

(البته حالت های دیگری هم وجود دارد که به همین دقتا 75% می رسد)

Subject:
Date

(7)

$$\varepsilon_2 = \frac{3e^{-\alpha_1}}{6e^{-\alpha_1} + e^{\alpha_1}} = \boxed{0.25}$$

$$\alpha_2 = \frac{1}{2}\ln\frac{1-\varepsilon_2}{\varepsilon_2} = \boxed{0.54}$$

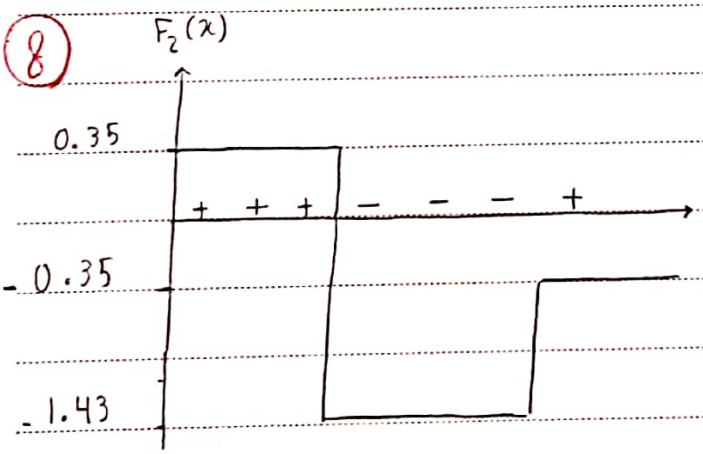
$$w_i^{(2)} = \begin{cases} e^{-\alpha_1}e^{-\alpha_2} = e^{-(\alpha_1+\alpha_2)} = e^{-1.43} & i = 4,5,6 \\ e^{\alpha_1}e^{-\alpha_2} = e^{\alpha_1-\alpha_2} = e^{0.35} & i = 7 \\ e^{-\alpha_1}e^{\alpha_2} = e^{\alpha_2-\alpha_1} = e^{-0.35} & i = 1,2,3 \end{cases}$$

* بنابراین نقاط منفی (4, 5, 6) کمترین وزن را دارند زیرا در هر دو طبقه بند درست پیش بینی شدند.

+ + + − − − +

کمترین وزن

(8)



$$\text{accuracy} = \frac{6}{7} \times 100 = \boxed{86\%}$$

Subject:
Date

خیر دقت بهبود نمی‌یابد اما همان طورکه در نمودار بالا مشاهده می‌کنید $f(x)$ دارد به اینکه داده 7 را درست پیش بینی کند نزدیک می‌شود و هنوز به تعداد کافی طبقه بند به آن اضافه شده است و اگر ادامه دهیم روی داده آموزش می‌توانیم به دقت ۱۰۰٪ برسیم. حتی اگر به همین طبقه بند حاصل یک پاپاس اضافه کنیم نیز به دقت ۱۰۰٪ می‌توانیم برسیم البته در حول کلی AdaBoost چنین ترمی وجود ندارد اما من در مثال های مختلف امتحان کردم و اگر این ترم پاپاس را اضافه کنیم می‌توانیم سریع‌تر دقت را افزایش دهیم و با تعداد کمتر طبقه بند ضعیف به بالاترین دقت ممکن برسیم.

3 Ensemble Learning

In this section, we will use the `credit scoring sample` dataset to predict whether a customer will repay their debt within 90 days or not; in fact, the resulting binary classifier will divide the customers into two categories: good payers and bad payers.

3.1 EDA & Preprocessing

The distribution of target features is as follows:

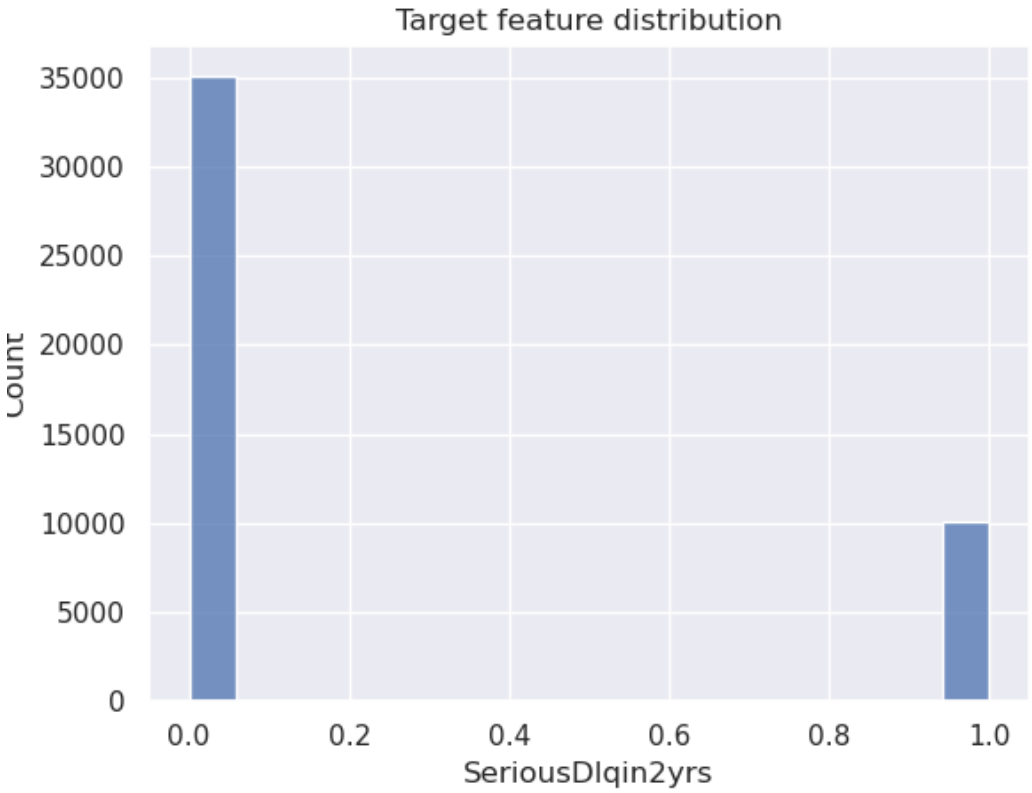


Figure 3.1: Target distribution

We fill the `Nan` values of each column with the median of that column.

3.2 Bootstrapping

We calculate the 90 % confidence interval for the mean of `age` column using Bootstrap, and the results shown in table 3.1.

Original mean	51.21
Bootstrap standard deviation	0.07
90% bootstrap confidence interval	(51.10, 51.32)

Table 3.1: Bootstrap for age mean

3.3 Random Forest

3.3.1 ROC AUC

You can see the ROC curve in the following figure:

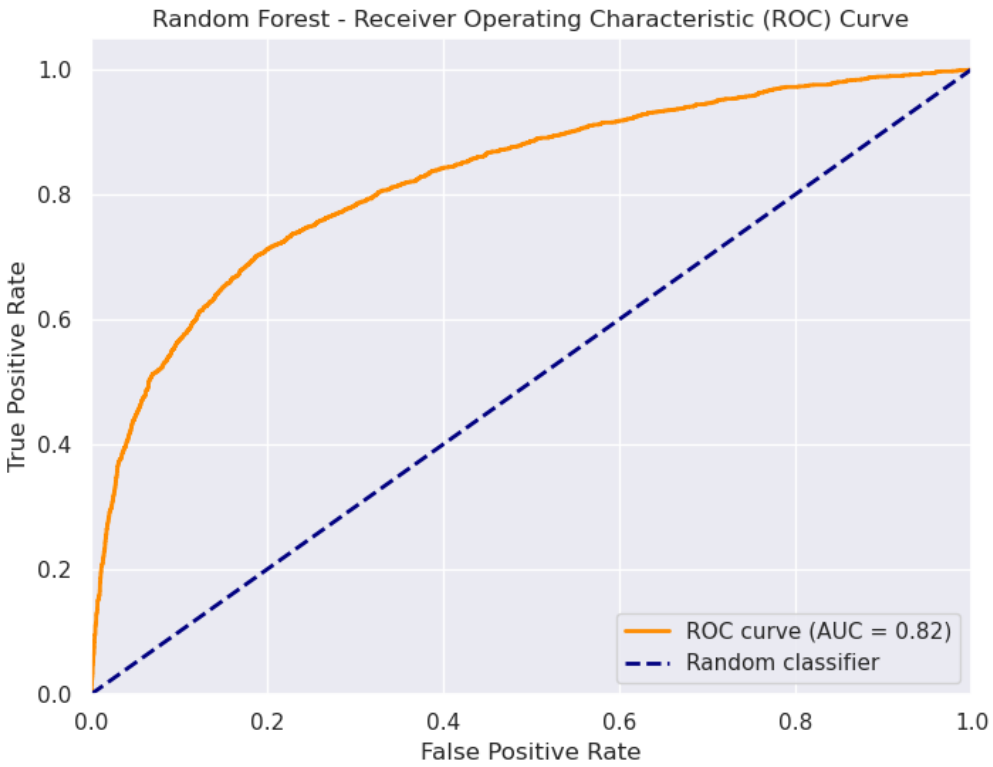


Figure 3.2: ROC curve of random forest

The ROC AUC is 0.89, which suggests that this is a generally robust and effective classifier, regardless of the chosen decision threshold.

3.3.2 Feature importance

In this dataset, the feature `NumberOfDependents` has the least impact on the random forest classifier.

3.4 Bagging

3.4.1 ROC AUC

You can see the ROC curve in the following figure:

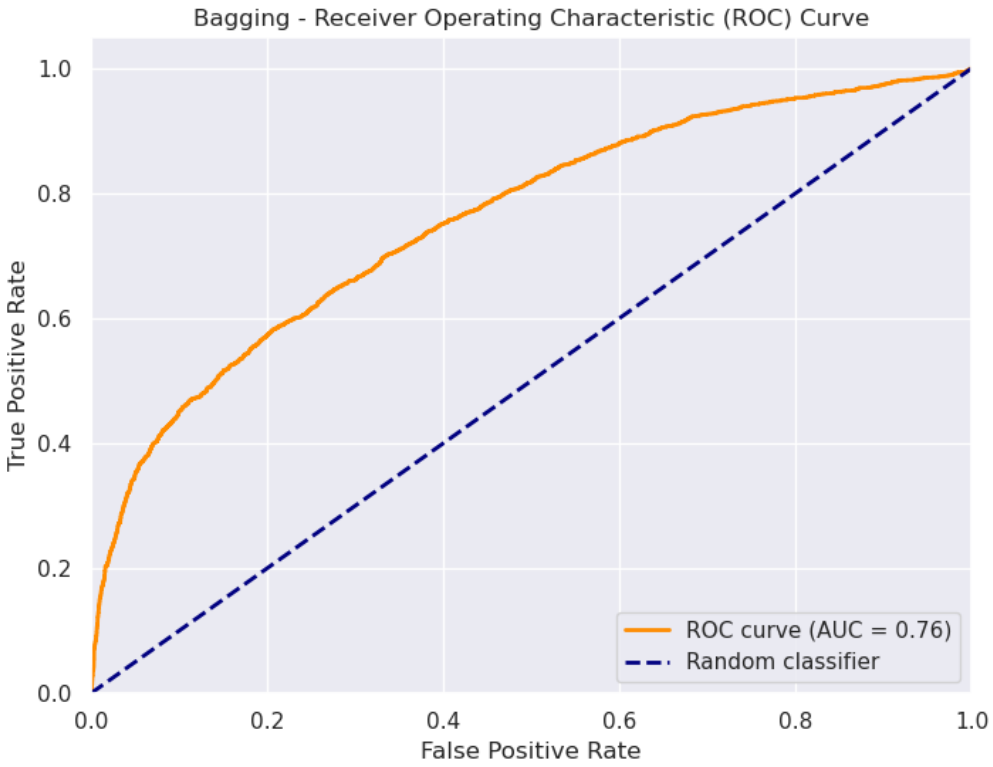


Figure 3.3: ROC curve of bagging

3.4.2 Best Hyper-parameter

max_samples	0.7
max_feature	4
estimator__C	10

Table 3.2: Best hyper-parameter for bagging

It is generally better to use 70 percent of the samples as the training data, which can lead to better generalization. This is because we cannot see all the data, so we should avoid memorizing the entire dataset.

It is also recommended to use almost all the features, which means that the 4 features were all helpful and have relatively low correlation with each other. Each feature can contribute to predicting the label differently and detect different patterns.

Finally, the use of a C value of 10 for the logistic regression estimator implies the use of a regularization term with a coefficient of 0.1. This is likely because with 4 features, there is a tendency to overfit the model. By including this regularization term, we can prevent overfitting and ultimately achieve good accuracy on the test data.

4 AdaBoost

In this section, we will implement an `AdaBoostClassifier` from scratch to classify `iris` dataset. We use a decision tree classifier with depth one as a base estimator.

4.1 Implementation

To begin, we will start with uniform initial weights for each sample and select a new set of samples. We will then train the weak learner and compute the weighted error rate (misclassification rate). Next, we will compute the learner weight using the SAMME algorithm and increase the weights of the misclassified samples. After that, we will randomly choose another set of samples based on the sample weights and repeat the same procedure for the specified number of estimators.

4.2 Evaluation

The following figure shows the confusion matrix of this model's prediction:

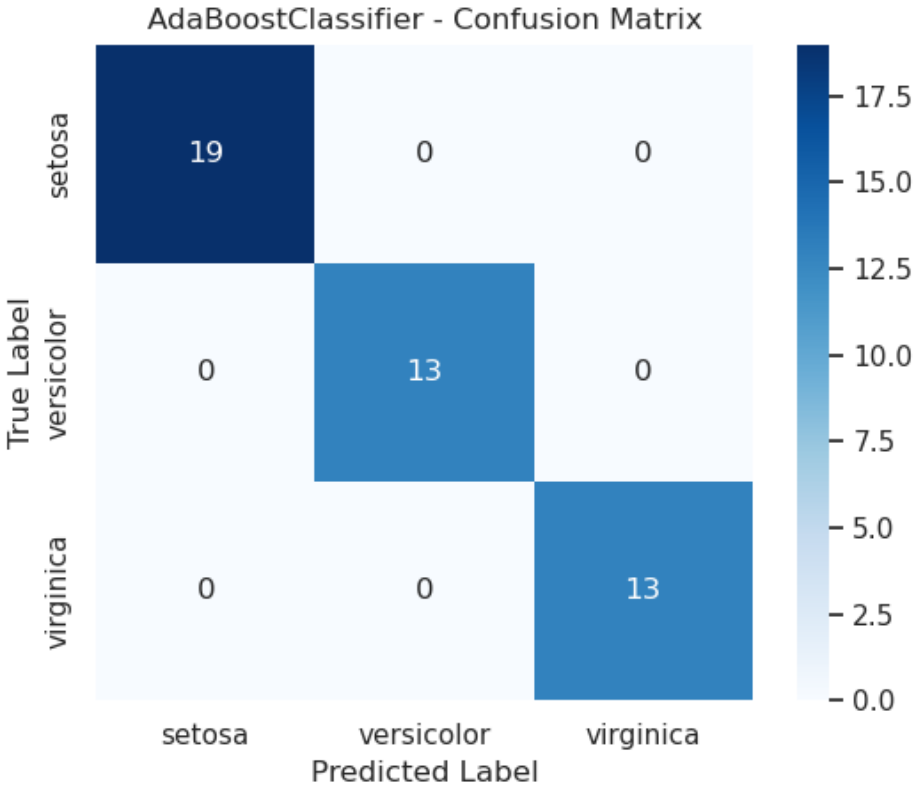


Figure 4.1: Confusion matrix of AdaBoost

The model's accuracy is 1.0, which means it is a perfect classifier for this dataset. This is likely because the dataset has a simple and small feature space that can completely separate each class. For example, if we used a single decision tree with a depth of 3, we could have achieved the same results.

The high accuracy of 1.0 also indicates that the model has learned the underlying patterns in the data extremely well. The reason is that each time we fit a model on the misclassified samples, we try to learn unlearned patterns and make the classifier better in each step.

Subject:
Date

سوال (5): Y ← انسداد شرایین

الف)

		✓	x	
نشا رخون	بله	6	2	$H(Y \mid \text{نشا رخون}) =$
	نه	3	3	$-\frac{8}{14}\left(\frac{6}{8}\log\frac{6}{8}+\frac{2}{8}\log\frac{2}{8}\right)-\frac{6}{14}(1)=\boxed{0.89}$

کلسترول	نرمال	3	2	$H(Y \mid \text{کلسترول})$
	بحرانی	4	0	$-\frac{5}{14}\left(\frac{3}{5}\log\frac{3}{5}+\frac{2}{5}\log\frac{2}{5}\right)-\frac{5}{14}\left(\frac{2}{5}\log\frac{2}{5}+\frac{3}{5}\log\frac{3}{5}\right)$
	بالا	2	3	$=\boxed{0.62}$

سیگار	بله	6	1	$H(Y \mid \text{سیگار}) = -\frac{7}{14}\left(\frac{6}{7}\log\frac{6}{7}+\frac{1}{7}\log\frac{1}{7}\right)$
	نه	3	4	$-\frac{7}{14}\left(\frac{3}{7}\log\frac{3}{7}+\frac{4}{7}\log\frac{4}{7}\right)=\boxed{0.78}$

وزن	نرمال	3	1	$H(Y \mid \text{وزن}) = -\frac{4}{14}\left(\frac{3}{4}\log\frac{3}{4}+\frac{1}{4}\log\frac{1}{4}\right)$
	اضافه وزن	4	2	$-\frac{6}{14}\left(\frac{4}{6}\log\frac{4}{6}+\frac{2}{6}\log\frac{2}{6}\right)-\frac{4}{14}(1)=\boxed{0.97}$
	چاق	2	2	

آن ویژگی که انتروپی کمتری دارد information gain بیش تری خواهد داشت پس

کلسترول انتخاب می شود.

Subject:
Date

نرمال = کلسترول

	✓	×
بله	3	۰
نه	۰	2

فشارخون

$H(\text{کلسترول} = \text{نرمال} \mid \text{فشارخون}) = 0$

چون انتروپی صفر شد پس بیش ترین information gain را بدست می آورد پس نیاز به بررسی سایر ویژگی ها نیست.

بالا = کلسترول

	✓	×
بله	2	۰
نه	۰	3

مصرف سیگار

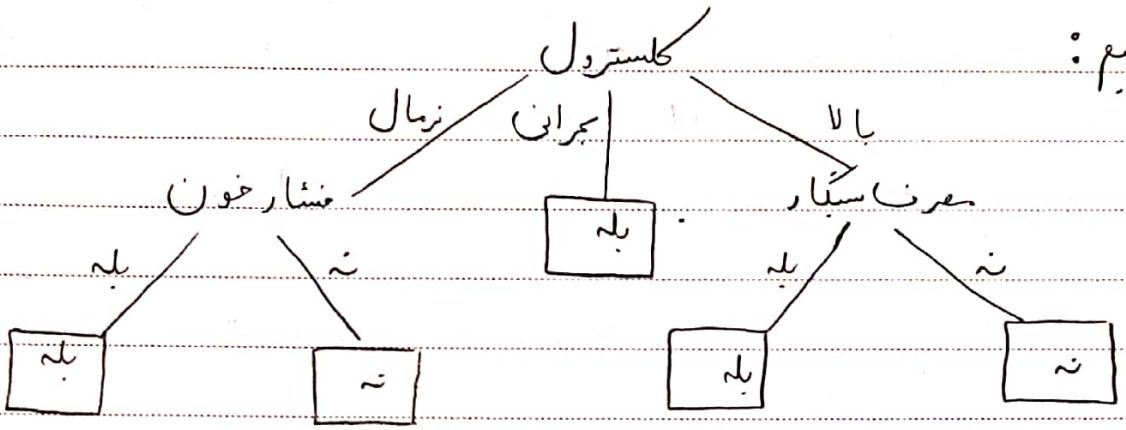
$H(\text{کلسترول} = \text{بالا} \mid \text{مصرف سیگار}) = 0$

مثل بالا نیاز به بررسی سایر ویژگی ها نیست.

برای (کلسترول = بحرانی) نیز چون انتروپی آن صفر است نیاز به شکست بیش تر نیست.

Subject:
Date

درخت تصمیم:



(ب)

	Predicted	True
15 ⟶	بله	بله
16 ⟶	بله	بله
17 ⟶	نه	نه
18 ⟶	بله	نه
19 ⟶	نه	بله

Confusion matrix

True label \ Predicted label	نه	بله
نه	1	1
بله	1	2

$$\text{accuracy} = \frac{3}{5} \times 100 = \boxed{60\%}$$

Subject:
Date

Recall بله بهتر از نه است یعنی بله ها را بهتر تشخیص می دهد و این خوب است برای مساله ما. زیرا ما می خواهیم حتی الامکان بیماری را سالم تشخیص ندهیم. دقت نیز 60٪ درصد است. Recall نیز 50٪ است که اصلا خوب نیست به طور کلی روی این داده ی تست مدل ما عملکرد خیلی خوبی ندارد

ج) زیر ارتفاع درخت همین طور می تواند زیاد شود و این معادل این است که تعداد پارامترهای مدل ما افزایش می یابد، در نتیجه پیچیدگی مدل بیش تری می شود. این پیچیدگی می تواند همین طور افزایش پیدا کند و حتی از پیچیدگی داده ی ما نیز بیش تر شود که این باعث می شود روی داده ی آموزش Overfit کنیم. یعنی به جای یادگیری Pattern داده ها، آن ها را حفظ می کند.

روش های جلوگیری:

① محدود کردن ارتفاع درخت: می توانیم یک مقدار ماکزیمم برای ارتفاع درخت قرار دهیم و اجازه ندهیم ارتفاع درخت بیش از آن رشد کند.

Subject:
Date

② هرس کردن (pruning) : می‌توانیم اجازه دهیم که درخت هر اندازه که می‌خواهد ارتفاعش را زیاد کند و overfit شود بعد از آن از پایین درخت شروع می‌کنیم و split هایی که information gain آن‌ها از یک حدی (که آن را مشخص می‌کنیم) کمتر بود را حذف می‌کنیم.

روش دوم می‌تواند بهتر باشد زیرا ممکن است یک بخش از داده‌ها پیچیدگی بیش‌تری داشته باشند و نیاز به ارتفاع بیش‌تری باشد و بخش دیگر نیاز به ارتفاع کمتری داشته باشد درحالی که ما در روش یک با همه‌ی بخش‌ها و شاخه‌های درخت به یک شکل برخورد می‌کنیم و درخت تقریباً متوازن می‌شود (که ممکن است در واقعیت متوازن نبوده باشد)

6 Decision Tree

In this section, we will implement a `decision tree classifier` from scratch to predict `Recidivism - Return to Prison numeric` feature of `prison_dataset`.

Feature	Unique Values
Fiscal Year Released	[2010, 2013, 2015]
Recidivism Reporting Year	[2013, 2016, 2018]
Race - Ethnicity	['White', 'Black']
Age At Release	['¡45', '¿45']
Convicting Offense Classification	['D Felony', 'Other', 'Felony']
Convicting Offense Type	['Violent', 'Other', 'Drug']
Convicting Offense Subtype	['Other', 'Trafficking']
Main Supervising District	['3JD', '5JD']
Release Type	['Parole', 'Discharged End of Sentence']
Part of Target Population	['Yes', 'No']
Recidivism - Return to Prison numeric	[1, 0] (target)

Table 6.1: Prison Dataset

6.1 Implementation

We use the ID3 algorithm to build the tree. First, we obtain the best feature, which achieves the most information gain, and then split the data into subsets that have the same value for that feature. We continue this approach in each subtree until reach the `max_depth` or no more feature exists to split the data.

6.2 Evaluation

We fit a model with `max_depth` of 3 and the following figure shows the confusion matrix of this model's prediction:

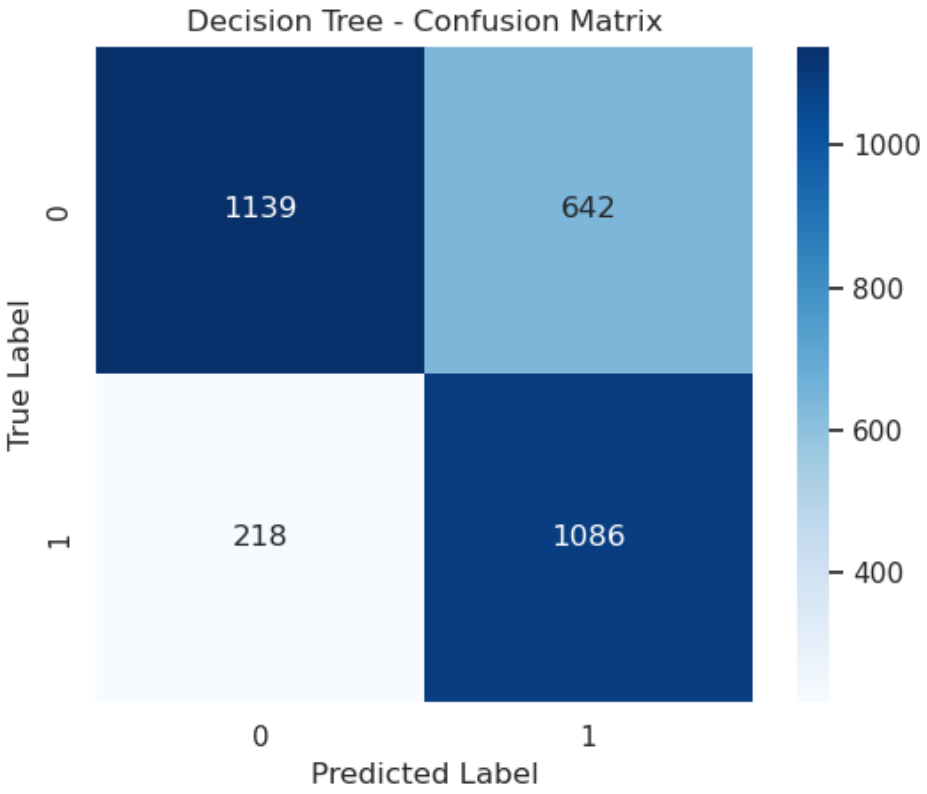


Figure 6.1: Confusion matrix of decision tree

The accuracy of the model is 0.72, which is quite good. As you can see in the confusion matrix the recall of class 1 is much better than class 0. I tried with different values of `max_depth` but the result was almost the same.