باسمه تعالی

عنوان:

Project Report – هپاتیت C

استاد:

دکتر خطیبی

دانشجویان:

محمد خان آبادی

سید حامد صالحی

1.پیش پردازش داده ها:

1.1معرفی دیتاست:

مجموعه داده ها شامل مقادیر آزمایشگاهی اهداکنندگان خون و بیماران هپاتیت C و مقادیر دموگرافیک مانند سن می باشد. این دیتاست شامل 615 مشاهده و 13 متغیر است.

این دیتاست از سایت www.kaggle.com بدست آمده است.

Table 1 معرفی متغیر ها

Feature	Unit	Stands for	Description	Normal range	Type
CATEGORY			سطر هدف		binary
AGE	year				numeric
SEX					binary
ALB	$\frac{g}{L}$	Albumin blood test	آلبومین پروتئین اصلی خون است و در کبد ساخته میشود. در بیماری های کبدی میزان کاهش میابد	34-54 $\frac{g}{L}$	numeric
ALP	$\frac{IU}{L}$	Alkaline phosphatase	آلکالین فسفاتاز یک آنزیم است که بیشتر در کبد، کلیه و استوان ها یافت میشود(برای تشخیص بیماری های کبدی و استخوانی تست آن انجام میشود)	44-147 $\frac{IU}{L}$	numeric
ALT	$\frac{U}{L}$	Alanine transaminase	ALTبه صورت طبیعی در داخل سلول های کبد یافت می شود .البته در مواقعی که کبد آسیب دیده باشد یا دچار التهاب شده باشد این احتمال وجود دارد که ALT وارد جریان خون شود. (برای تشخیص بیماری های کبدی تست آن انجام میشود)	4-36 $\frac{U}{L}$	numeric

AST	$\frac{U}{L}$	Aspartate transaminase	بالاترین غلظت AST در کبد، عضلات، قلب، کلیه، مغز و گلبول های قرمز خونی است .به طور معمول مقدار کمی AST در جریان خون وجود دارد. مقادیر بالاتر از حد طبیعی این آنزیم در خون می‌تواند نشان دهنده مشکلی در سلامتی باشد. سطوح غیر طبیعی این ماده می‌تواند با آسیب های کبدی همراه باشد.وقتی آسیبی به بافت‌ها و سلول‌های حاوی این آنزیم وارد می‌شود، سطوح AST افزایش می‌یابد.	8-33 $\frac{U}{L}$	numeric
BIL	$\frac{g}{dL}$	Bilirubin	بیلیروبین یک رنگدانه زرد-نارنجی است که در طی تجزیه طبیعی گلبول های قرمز خون ساخته میشود.آزمایش بیلیروبین میزان بیلیروبین در بدن را نشان می‌دهد. گاهی کبد نمی‌تواند بیلی روبین بدن را پردازش کند. این امر در نتیجه افزایش بیلی روبین، انسداد یا التهاب کبد رخ می‌دهد.	بستگی به سن افراد دارد	numeric
CHE	$\frac{U}{mL}$	Acetylcholinesterase	CHE یک آنزیم است که عمدتاً در عضلات و اعصاب یافت می شود.کمبود آن باعث بیماری های کبدی میشود.	8-18 $\frac{U}{mL}$	numeric
CHOL	$\frac{mmol}{L}$	Cholestrol	زیاد بون آن باعث بیماریهای قلبی و انباشته شدن چربی در رگ ها میشود	Below 6.18 $\frac{mmol}{L}$	numeric
CREA	$\frac{micromoles}{L}$	Creatinine	کراتینین یک ماده زائد است که وقتی کراتین موجود در ماهیچه شما تجزیه می شود، تشکیل می شود. سطح کراتینین در خون می تواند اطلاعاتی در مورد عملکرد کلیه ها ارائه دهد.	مردان: 65.4-119.3 $\frac{micromoles}{L}$ زنان: 52.2-91.9 $\frac{micromoles}{L}$	numeric
GGT	$\frac{IU}{L}$	Gamma-Glutamyl-Transferase	GGT آنزیمی است که در سراسر بدن یافت می شود، اما بیشتر در کبد یافت می شود. هنگامی که کبد آسیب می بیند، GGT ممکن است به جریان خون نشت کند. سطوح بالای GGT در خون ممکن است نشانه ای از بیماری کبدی یا آسیب به مجاری صفراوی باشد.	0-30 $\frac{IU}{L}$	numeric

numeric	60-83 $\frac{g}{L}$	زیاد بودن آن میتواند ناشی از هپاتیت c باشد	Proteins	$\frac{g}{L}$	PROT

2.1 **شناسایی** Missing values

طبق اطلاعات اولیه ای که از دیتاست میگیریم ، جنس هر متغیر و تعداد missing value های هر ستون قابل تشخیص است :

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 615 entries, 0 to 614
Data columns (total 13 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Category  615 non-null    object
 1   Age       615 non-null    int64
 2   Sex       615 non-null    object
 3   ALB       614 non-null    float64
 4   ALP       597 non-null    float64
 5   ALT       614 non-null    float64
 6   AST       615 non-null    float64
 7   BIL       615 non-null    float64
 8   CHE       615 non-null    float64
 9   CHOL      605 non-null    float64
 10  CREA      615 non-null    float64
 11  GGT       615 non-null    float64
 12  PROT      614 non-null    float64
dtypes: float64(10), int64(1), object(2)
memory usage: 62.6+ KB
```

```
Category     0
Age          0
Sex          0
ALB          1
ALP         18
ALT          1
AST          0
BIL          0
CHE          0
CHOL        10
CREA         0
GGT          0
PROT         1
dtype: int64
```

ما در این دیتاست به 3 روش missing value ها را پر کردیم:

1. میانگین هر ستون ← df_mean
2. میانه هر ستون ← df
3. KNN ← df_knn

به علت تحت تاثیر قرار نگرفتن میانه توسط نقاط پرت و همچنین سادگی، این روش را برای ادامه کد خود استفاده میکنیم.

لازم به ذکر است برای پر کردن داده ها به روش KNN متغیر های numeric را جدا کردیم و با روش KNN Imputer آن ها را پر کردیم. از آن جایی که در متغیر های categorical ، missing value نداشتیم ، این روش را برای متغیر های categorical انجام ندادیم.

3.1 تبدیل متغیر های اسمی به عددی

همانطور که قبلا مشاهده کردیم 2 ستون "sex" و "Category" اسمی هستند . آن ها را به نحو زیر تبدیل به متغیر های عددی میکنیم:

Sex: {"m"=1} , {"f"=0}

Category: {"0=Blood Donor"=1 , "0s=suspect Blood Donor"=1} , {"1=Hepatitis"=0 , "2=Fibrosis"=0 , "3=Cirrhosis"=0}

4.1 Scaling

از آنجایی که در بعضی از مدل ها فاصله بین متغیر ها از هم نیاز است، قبل از مدلسازی مقیاس داده ها را یکسان میکنیم یا به عبارتی scaling انجام میدهیم. ما در این دیتاست به 2 روش scaling را انجام دادیم:

1. Min-max scaling ← df_minmax
2. Standard scaling ← df_zscore

در روش min-max داده ها بین 0 و 1 قرار میگیرند. فرمول روش min-max بصورت زیر است:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

در روش standars scaling عمده داده ها بین 3 و -3 قرار میگیرند و داده هایی که خارج از این باره هستند را میتوان نقاط پرت در نظر گرفت. فرمول روش standard scaling به صورت زیر است:

$$z = \frac{x - \mu}{\sigma}$$

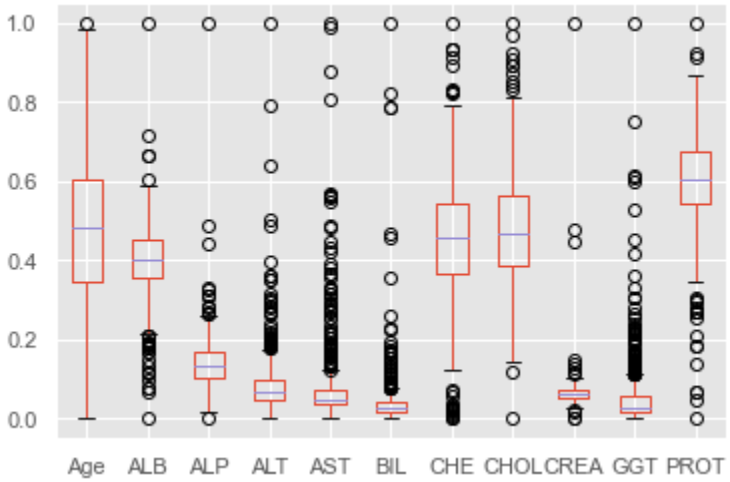
$\mu =$ Mean
$\sigma =$ Standard Deviation

5.1 شناسایی و حذف داده های پرت

2 روش برای شناسایی داده های پرت را ما در این دیتاست استفاده کردیم: 1) box plot 2) z-score

مبنای روش اول استفاده از چارک ها است.

Figure 1- box plot for all numeric variables



همانطور که در تصویر بالا میبینید، box plot تمام متغیر های عددی رسم شده است. خط بالایی هر box plot برابر است با: 1.5*Q3[1] و خط پایینی هر box plot برابر است با: -1.5*Q1[2] . هر داده ای که خارج این محدوده باشد، داده پرت محسوب میشود. خود جعبه box plot نمایانگر Q1 (خط پایینی جعبه) و Q3 (خط بالایی جعبه) است.

مبنای روش دوم استفاده از scaling داده ها (به روش standard scaling) است. همانطور که در بخش scaling توضیح داده شد هر داده ای که خارج از محدوده 3 و -3 باشد، داده پرت محسوب میشود.

بر مبنای همین دو روش، یعنی IQR و z-score حذف داده های پرت انجام شد:

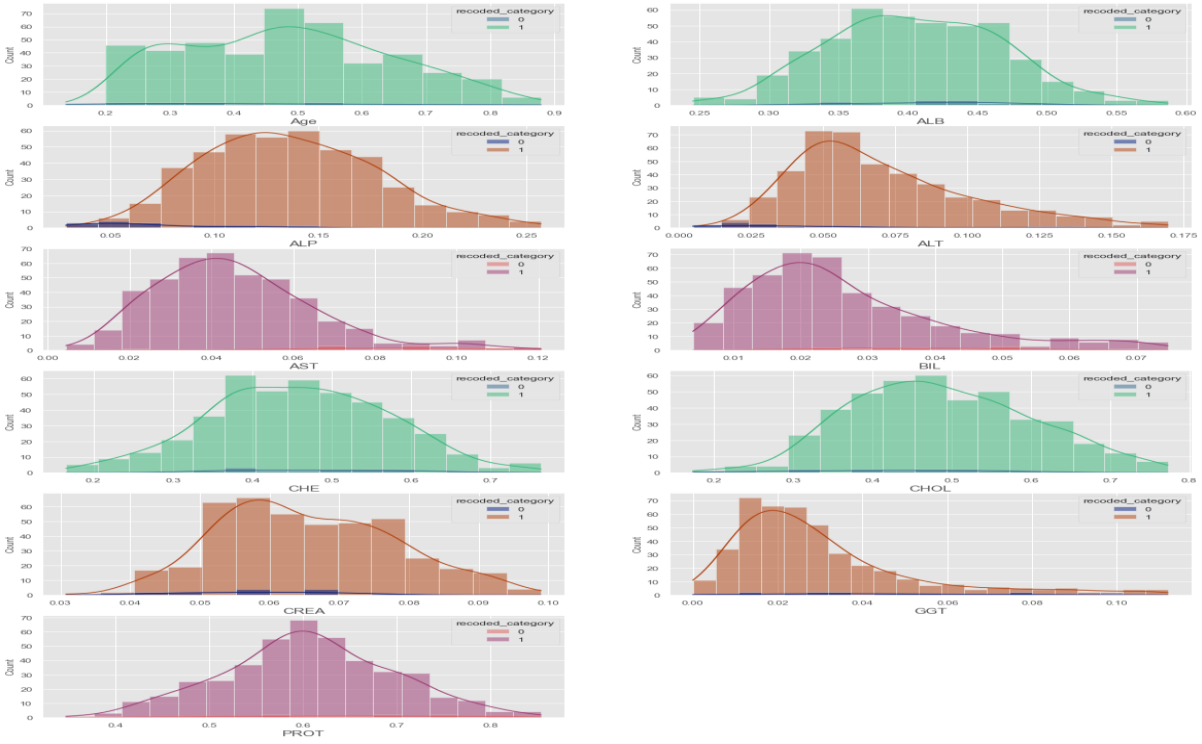
1. حذف بر مبنای z-score ← df_clean1 ← 554 rows
2. حذف بر مبنای IQR ← df_clean2 ← 447 rows

6.1 Visualization

در ابتدا برای این که متوجه شویم که آیا متغیر ها از توزیع نرمال پیروی میکنند یا خیر، هیستوگرام آن ها را رسم کردیم:

[1] چارک سوم
[2] چارک اول

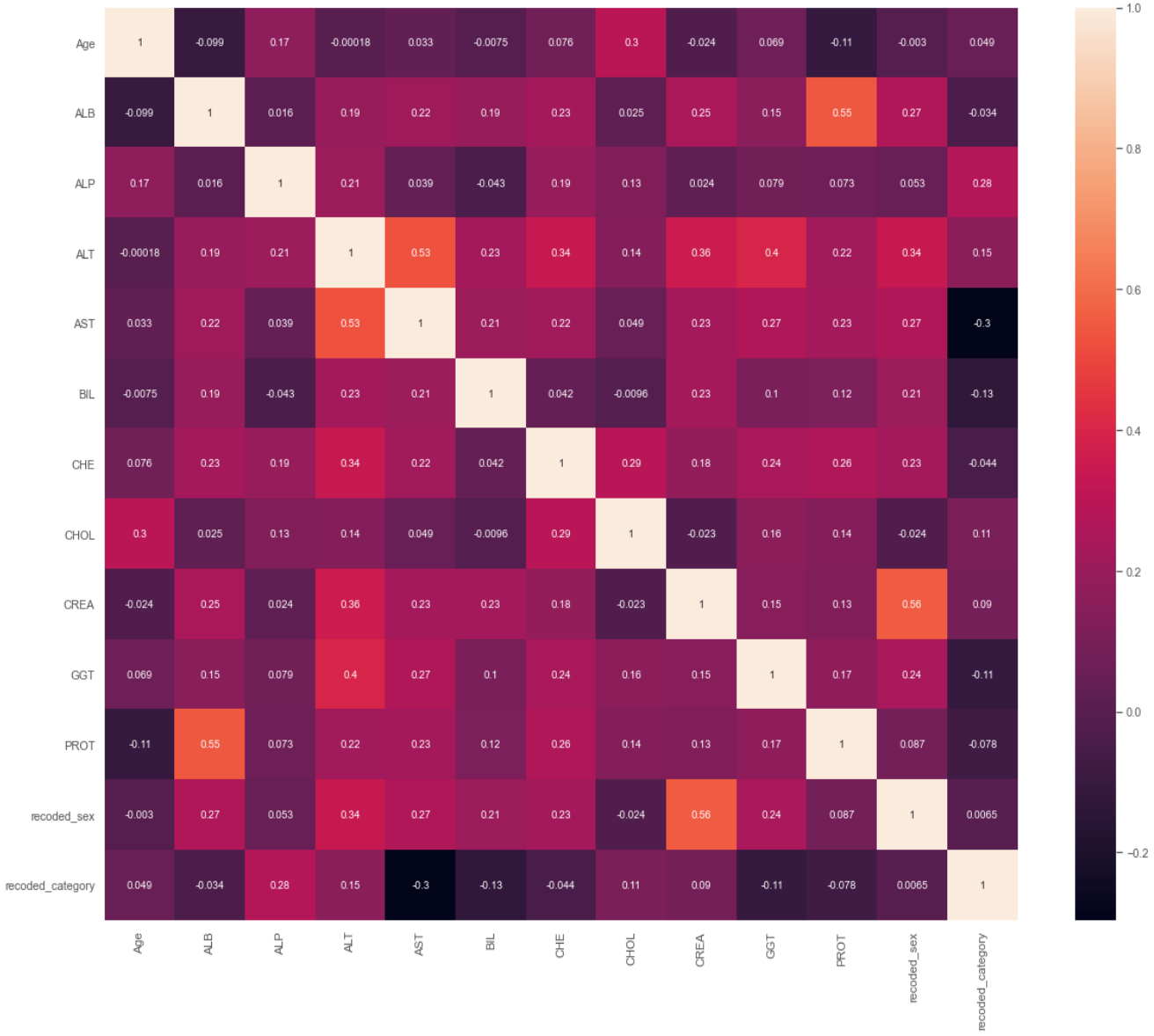
Figure 2- histogram



همانطور که از شکل پیداست اکثر متغیر ها از توزیع نرمال پیروی نمیکنند. برای نرمال کردن آنها میتوان از روش box-cox استفاده کرد که موضوع بحث ما نیست.

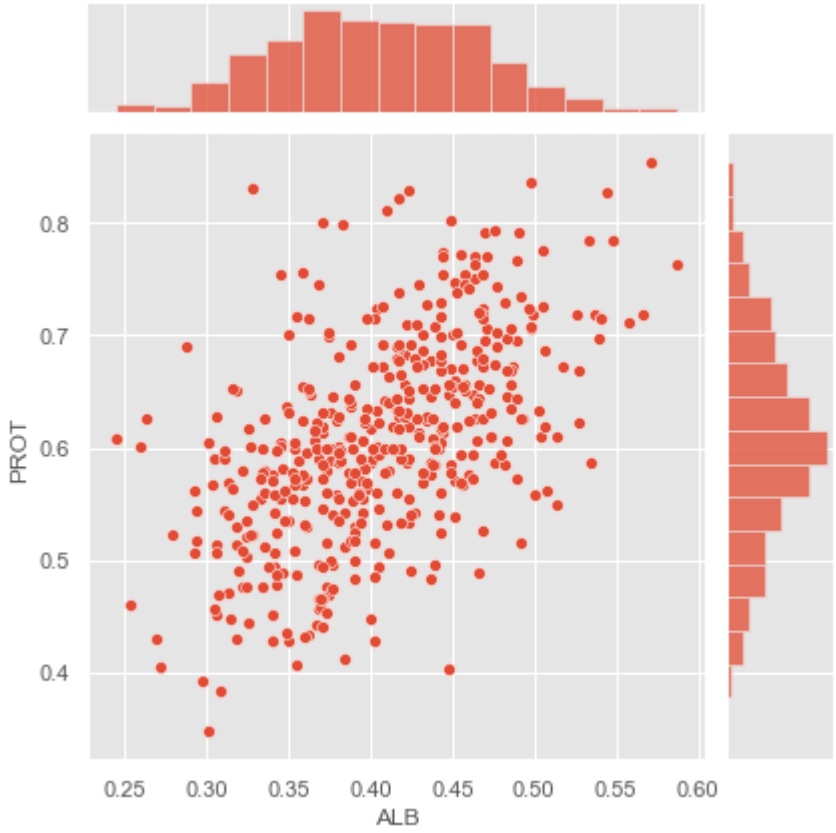
برای بحث feature selection نمودار های heatmap(correlation) و pairplot کمک شایانی به ما میکنند.

Figure 3- heatmap(correlation)



همانطور که از نمودار بالا پیداست 2 داده عددی که با هم بیشترین همبستگی را دارند "ALB" و "PROT" هستند با corr=0.55 که همبستگی آن ها از نوع مثبت است. برای درک بهتر این موضوع نمودار joint plot این دو متغیر نیز رسم شده است.

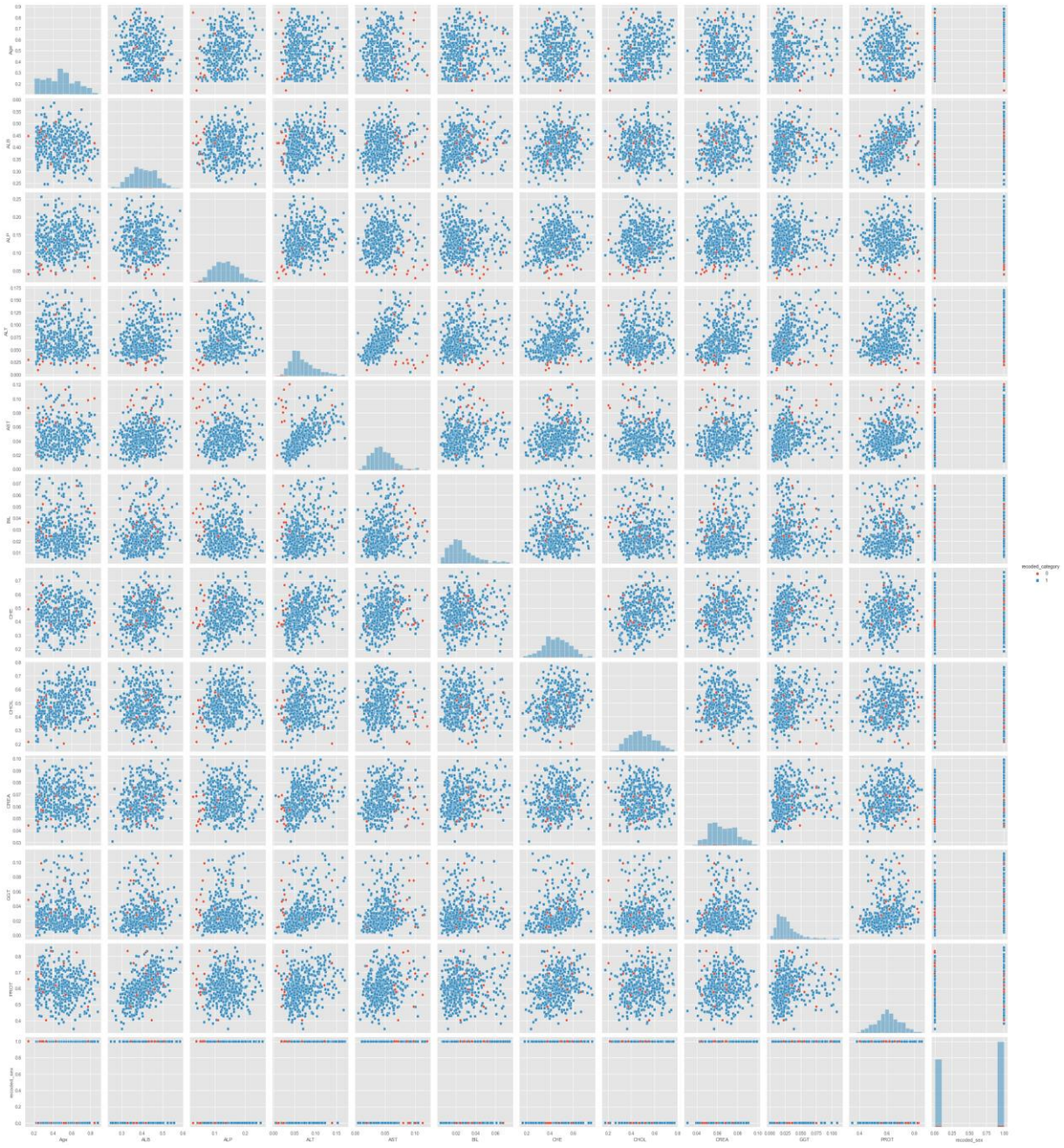
Figure 4 – joint plot for "PROT" & "ALB"



همانطور که از نمودار بالا پیداست همبستگی دو متغیر مثبت است، یعنی میتوان گفت که تقریبا با افزایش "ALB" "PROT" افزایش پیدا میکند.

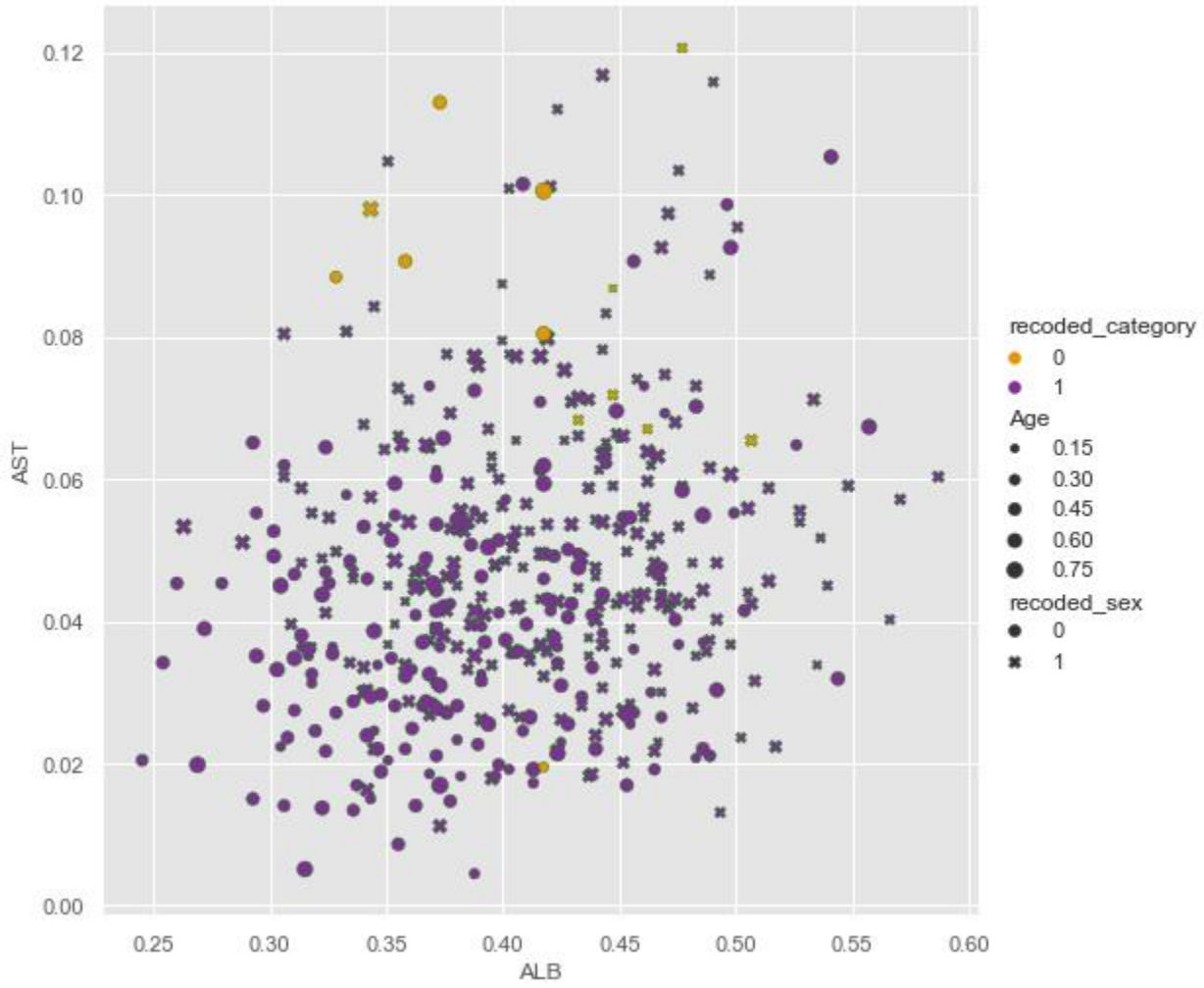
نمودار pairplot نیز همانند نمودار joint plot رابطه دو به دو ی متغیر ها را نشان میدهد و میتوان به کمک آن همبستگی بین دو متغیر را متوجه شد. تصویر آن در زیر آورده شده است:

Figure 5- pairplot



یک نمودار مفید دیگری که از آن استفاده کردیم rel plot است که میتوان با تغیر آرگمان های آن به اشکال مختلفی آن را رسم کرد. برای مثال یک نمونه از آن را در تصویر زیر میبینید:

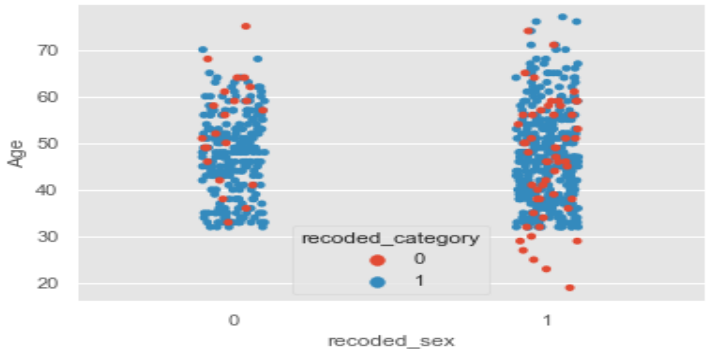
Figure 6- rel plot



از نمودار بالا میتوان متوجه شد که اکثر کسانی که بیمار هستند (در کلاس صفر قرار میگیرند) ، دارای "AST" بالایی هستند اما در رابطه با "ALB" آن ها نمیتوان اظهار نظر کرد. همچنین در رابطه با جنسیت و سن آنها نمیتوان اظهار نظر کرد.

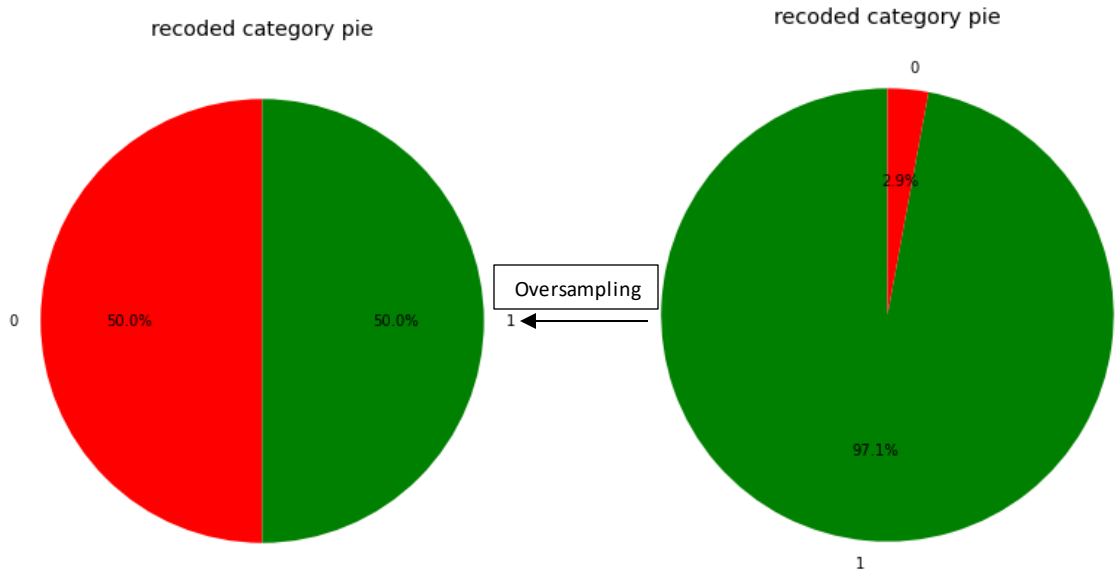
نکته دیگری که در این دیتاست وجود دارد این است که تمامی مردان زیر 30 سال بیمار هستند (درکلاس 0 قرار میگیرند). این نکته بوسیله نمودار strip plot که در زیر تصویر آن آمده است، قابل برداشت است.

Figure 7-strip plot for "Age"



7.1 رفع مشکل بالانس نبودن داده

بالانس نبودن داده باعث میشود که دقت هایی که از مدل های مختلف میگیریم دقیق نباشد. به همین دلیل قبل از مدلسازی بوسیله Oversampling داده ها را بالانس میکنیم. از آن جایی که تعداد کلاس 0 ما 13 تا و تعداد کلاس 1 ما 434 بود از undersampling استفاده نکردیم. در این دیتاست موضوع بالانس نبودن داده ها را بوسیله Pie chart متوجه شدیم.




```
0    434
1    434
```

```
1    434
0     13
```

Figure 8 - pie chart for showing the imbalancing of data

8.1 تقسیم داده ها به test و train

در این مطالعه نسبت تقسیم داده ها 0.25 است. یعنی 0.25 داده ها test و 0.75 داده ها train هستند.

x_test=(217,12) و x_train=(651,12)

2.مدلسازی

برای classification داده ها از دسته بندهای: KNN – decision tree – SVM – bagging – boosting (Ada boosting) استفاده شده است که هر کدام به صورت جداگانه توضیح داده میشود.

یکی از چالش های دسته بند ها تعیین هایپر پارامتر ها است. در ابتدا برای هر دسته بند دقت را روی داده های test و train نمایش میدهیم (این کار را قبل از تعیین هایپر پارامتر ها میکنیم که دقت دسته بند با هایپر پارامتر های پیش فرض پایتون بدست بیاوریم) و بعد از تعیین هایپر پارامتر ها بوسیله grid search نیز دوباره دقت دسته بند را محاسبه میکنیم و بوسیله هیت مپ و report آن را نمایش میدهیم.

KNN 1.2

KNN را نمیتوان جزو مدل ها حساب کرد زیرا مدلی نمیسازد . به همین دلیل به آن دسته بند KNN گفته میشود. این دسته بند تنبل است و تا وقتی داده های تست را نداشته باشد کار انجام نمیدهد.

دقت اولیه مدل:

```
Accuracy on testing data:  0.967741935483871
Accuracy on training data:  0.9662058371735791
```

هایپر پارامتر ها:

یکی از هایپر پارامتر های مهم و تاثیرگذار در KNN ، N_neighbors[3] است. برای تعیین آن دو نمودار رسم کردیم که یکی دقت test و train را به ازای K های مختلف رسم میکند و دیگری خطا را به ازای K های مختلف برای این دسته بند رسم میکند.

[3] K value

Figure 9- accuracy for KNN

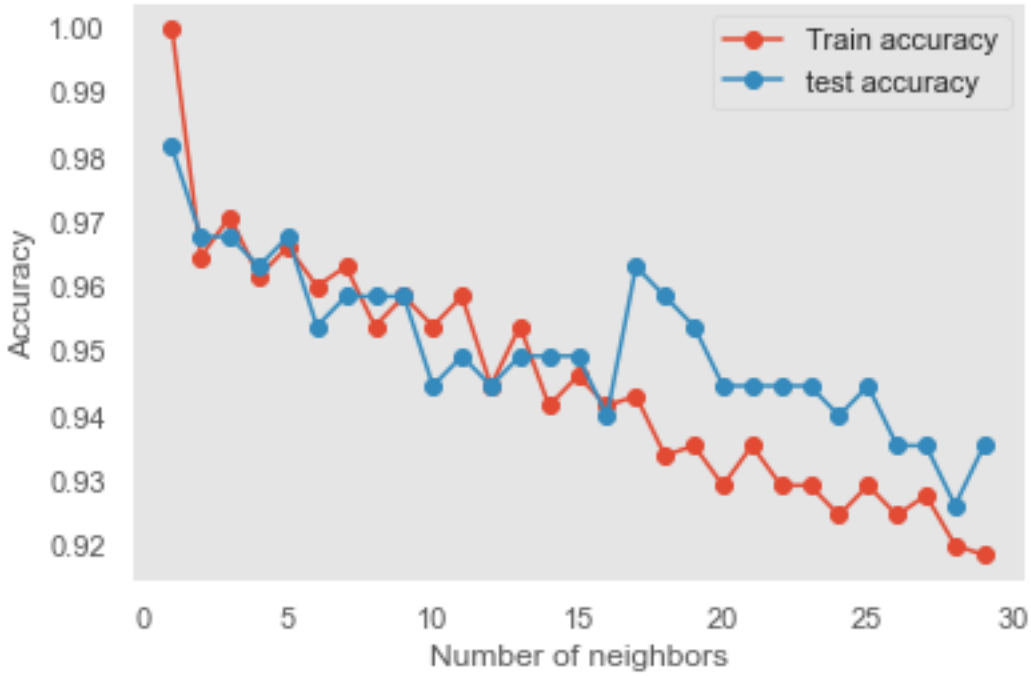
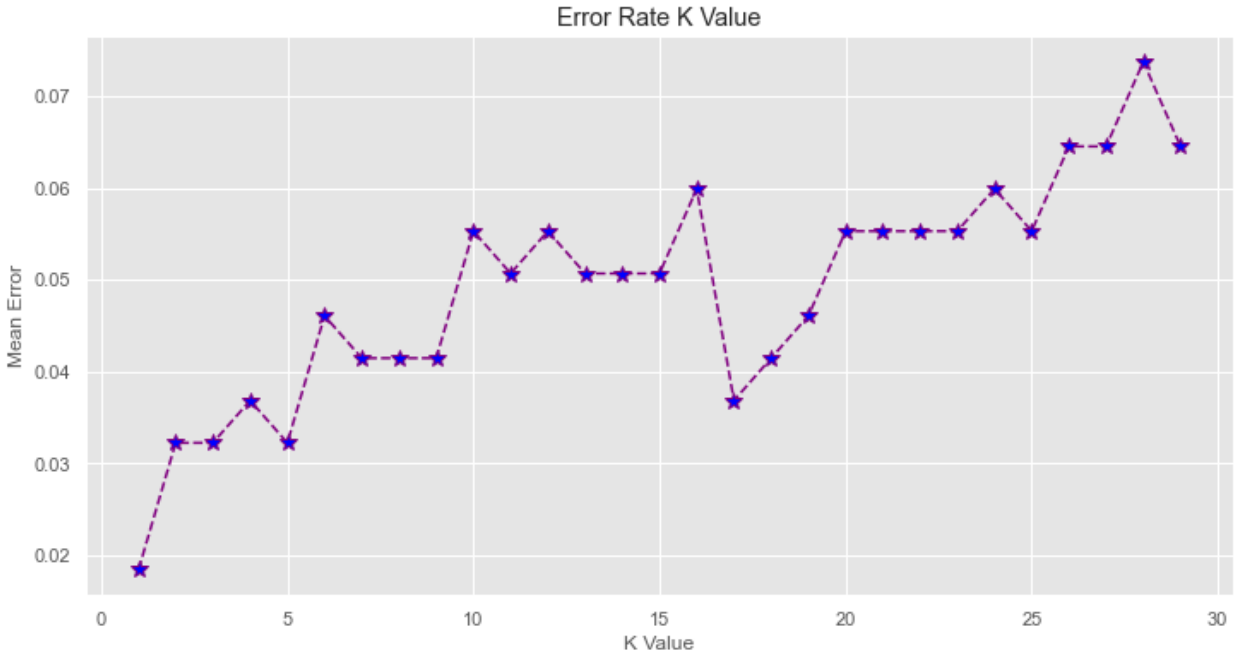


Figure 10-error for KNN



از شکل های 9 و 10 برداشت میشود که K=5 , K=3 مناسب است زیرا در این 2 نقطه overfitting و underfitting اتفاق نیفتاده است و مدل نیز دارای کمترین خطا است (در K=1 کمترین خطا رخ داده است اما از آنجا

که مدل بسیار محدود میکند از این نقطه صرف نظر میکنیم). برای انتخاب N_neighbors نهایی و همچنین نوع فاصله ای[۴] که دسته بند استفاده کند تا حداکثر دقت حاصل شود از grid search استفاده کردیم.

N_neighbors: تعداد همسایه‌هایی که برای تعیین کلاس داده مورد نظر استفاده میشود.

Metric: متریک فاصله ای که برای محاسبات استفاده میشود. متریک پیش فرض minkowski است با p=2 که معادل متریک استاندارد اقلیدسی است.

گزارش دهی دقت دسته بند بعد از تعیین هایپر پارامتر ها بوسیله grid search:

همانطور که توضیح داده شد 2 هایپر پارامتر را وارد grid search کردیم که K و metric هستند. بر طبق grid search بهترین مقادیر به صورت زیر است:

```
Best: 0.976970 using {'metric': 'manhattan', 'n_neighbors': 1}
```

همانطور که قبلا توضیح داده شد K=1 را نمیتوانیم انتخاب کنیم به همین دلیل به سراغ رتبه دوم میرویم:

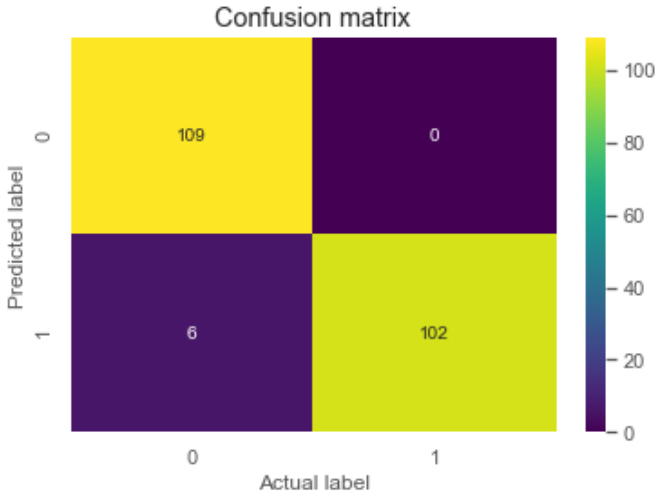
```
0.971841 (0.019452) with: {'metric': 'manhattan', 'n_neighbors': 3}
```

نتایج دسته بند در 2 قالب زیر آورده شده است(بعد از تعیین هایپر پارامتر ها):

	precision	recall	f1-score	support
0	0.95	1.00	0.97	109
1	1.00	0.94	0.97	108
accuracy			0.97	217
macro avg	0.97	0.97	0.97	217
weighted avg	0.97	0.97	0.97	217

[4] metric

Figure 11-heatmap for KNN



Decision tree 2.2

درخت تصمیم یک ساختار فلوچارت مانند است که در آن هر گره داخلی یک آزمون را بر روی یک ویژگی نشان می دهد (به عنوان مثال اینکه آیا یک سکه شیر می آید یا خط)، هر شاخه نشان دهنده نتیجه آزمایش است، و هر گره برگ نشان دهنده یک برچسب کلاس . این دسته بند نسبت به بقیه دسته بند ها بیشتر مستعد overfitting شدن است.

دقت اولیه مدل:

```
Accuracy on testing data:  0.9953917050691244
Accuracy on training data:  1.0
```

اگر دقت مدل test پایین تر بود میشد برداشت کرد که مدل دچار overfitting شده است اما از آنجا که بهم نزدیک هستند مشکلی در مدل نیست.

هایپر پارامتر ها:

هایپر پارامتر های max-depth , min_samples_split , ccp_alpha , criterion برای این دسته بند در نظر گرفته شده است.

Criterion: عملکرد اندازه گیری کیفیت یک تقسیم. معیارهای پشتیبانی شده "gini" برای ناخالصی Gini و "آنتروپی" برای information gain هستند.

Ccp_alpha: پارامتر پیچیدگی مورد استفاده برای هرس که حداقل هزینه-پیچیدگی را دارد. به طور پیش فرض هرس انجام نمی شود.

Min_samples_split: حداقل تعداد نمونه مورد نیاز برای تقسیم internal node.

Max_depth: حداکثر عمق درخت. اگر None باشد، گره‌ها تا زمانی که همه برگ‌ها خالص شوند یا تا زمانی که همه برگ‌ها کمتر از min_samples_split نمونه‌داشته باشند، گسترش می‌یابند.

گزارش دهی دقت دسته بند بعد از تعیین هایپر پارامتر ها بوسیله grid search:

بهترین مقادیر به صورت زیر است:

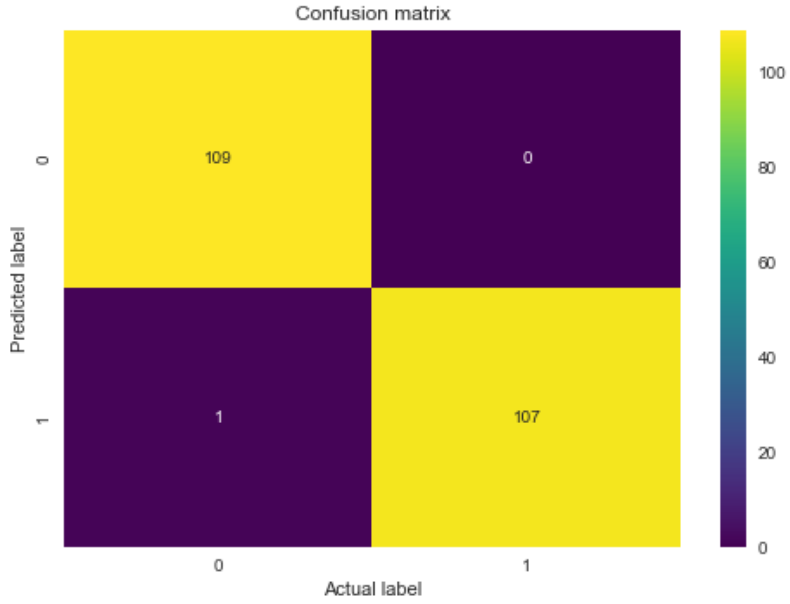
```
Best: 0.984134 using {'ccp_alpha': 0.002, 'criterion': 'gini', 'max_depth': 8, 'min_samples_split': 3}
```

توجه: همانطور که قابل مشاهده است در این دسته بند دقت مدل بعد از تعیین پارامتر ها کمتر از دقت مدل پیش از تعیین پارامتر ها است. استدلال ما این است که از آنجایی که ما در grid search تنها مقادیر محدودی را به آن میدهیم تا چک کند، امکان دارد که بهترین دقت در آن مقادیر محدودی که ما دادیم نباشد.

نتایج دسته بند در 2 قالب زیر آورده شده است(بعد از تعیین هایپر پارامتر ها):

	precision	recall	f1-score	support
0	0.99	1.00	1.00	109
1	1.00	0.99	1.00	108
accuracy			1.00	217
macro avg	1.00	1.00	1.00	217
weighted avg	1.00	1.00	1.00	217

Figure 12-heatmap for decision tree



توجه: در این دسته بند از آنجایی که از فاصله بین داده ها استفاده نمیکند،نیاز به استفاده از دیتاست نرمال شده نیست.

3.2 SVM:Support Vector Machine

ماشین‌های بردار پشتیبان (SVM) مجموعه‌ای از روش‌های یادگیری تحت نظارت هستند که برای طبقه‌بندی، رگرسیون و تشخیص نقاط پرت استفاده می‌شوند. ... از زیرمجموعه ای از نقاط آموزشی در تابع تصمیم گیری استفاده می کند (به نام بردارهای پشتیبان)، بنابراین در استفاده از حافظه نیز کارآمد است.

دقت اولیه مدل:

```
Accuracy on testing data:  0.9769585253456221
Accuracy on training data:  0.9708141321044547
```

هایپر پارامتر ها:

هایپر پارامتر های kernel , gamma , degree برای این دسته بند در نظر گرفته شده است.

Kernel: نوع kernel مورد استفاده در الگوریتم را مشخص می کند. اگر هیچ کدام داده نشود، از «rbf» استفاده می شود.

Gamma: ضریب kernel برای" rbf"، "poly" و "sigmoid".

Degree: درجه تابع چند جمله ای ("poly"). توسط تمام kernel های دیگر نادیده گرفته شده است.

گزارش دهی دقت دسته بند بعد از تعیین هایپر پارامتر ها بوسیله grid search:

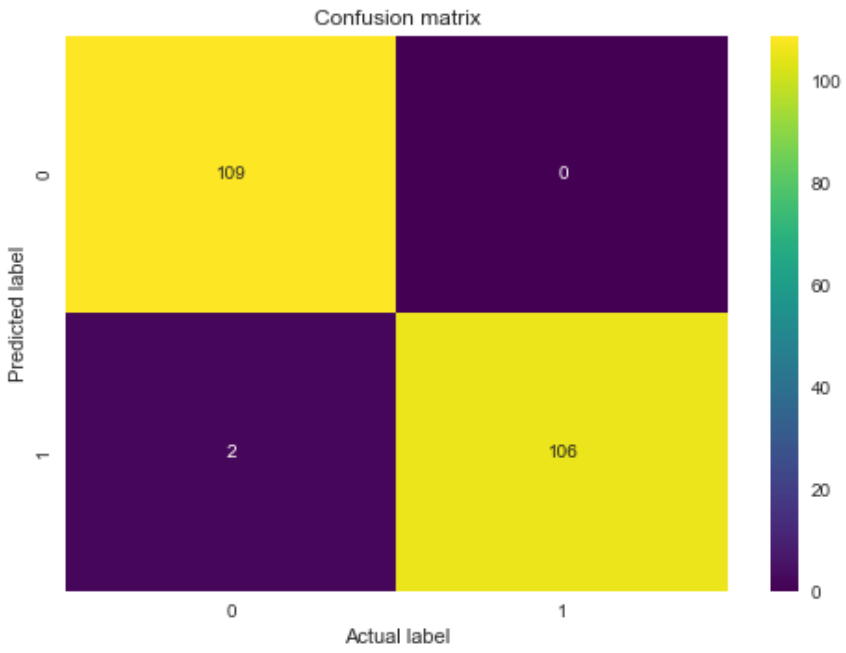
بهترین مقادیر به صورت زیر است:

```
Best: 0.990785 using {'degree': 6, 'gamma': 'scale', 'kernel': 'poly'}
```

نتایج دسته بند در 2 قالب زیر آورده شده است(بعد از تعیین هایپر پارامتر ها):

	precision	recall	f1-score	support
0	0.98	1.00	0.99	109
1	1.00	0.98	0.99	108
accuracy			0.99	217
macro avg	0.99	0.99	0.99	217
weighted avg	0.99	0.99	0.99	217

Figure 13-heatmap for SVM



4.2 **Bagging**

bagging، همچنین به عنوان bootstrap aggregation شناخته می شود، روش یادگیری مجموعه ای است که معمولا برای کاهش واریانس در یک مجموعه داده noisy استفاده می شود. در بسته بندی، یک نمونه تصادفی از داده ها در یک مجموعه آموزشی با جایگزینی انتخاب می شود - به این معنی که نقاط داده فردی را می توان بیش از یک بار انتخاب کرد.

دقت اولیه مدل:

```
Mean Accuracy: 0.987
Std :0.009
Accuracy on testing data:  0.9815668202764977
Accuracy on training data:  0.9969278033794163
```

هایپر پارامتر ها:

هایپر پارامتر های n_estimatores , max_samples برای این دسته بند در نظر گرفته شده است.

Max_samples: تعداد نمونه‌هایی که باید از X برای آموزش هر تخمین‌گر پایه (با جایگذاری به طور پیش‌فرض) ترسیم شود.

N_estimatores: تعداد برآوردگرهای پایه در مجموعه.

گزارش دهی دقت دسته بند بعد از تعیین هایپر پارامتر ها بوسیله grid search:

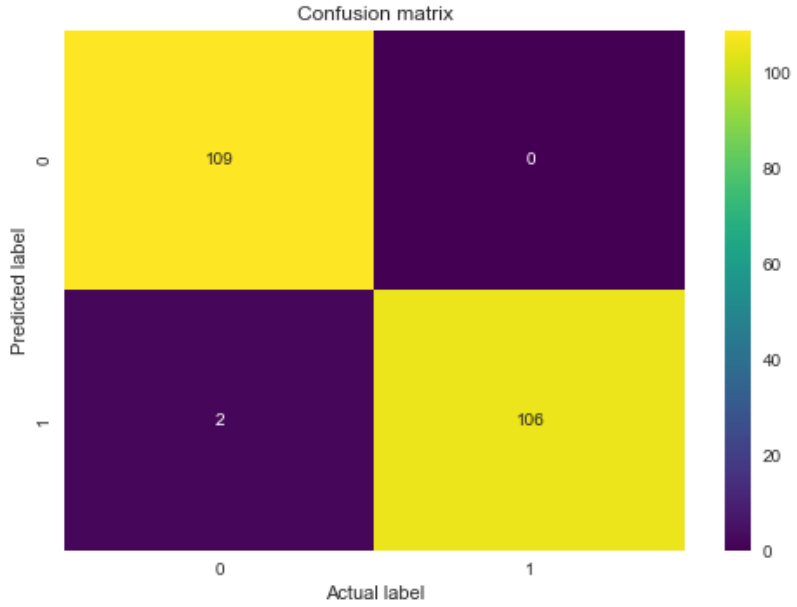
بهترین مقادیر به صورت زیر است:

```
Best: 0.988741 using {'max_samples': 0.75, 'n_estimators': 25}
```

نتایج دسته بند در 2 قالب زیر آورده شده است(بعد از تعیین هایپر پارامتر ها):

	precision	recall	f1-score	support
0	0.98	1.00	0.99	109
1	1.00	0.98	0.99	108
accuracy			0.99	217
macro avg	0.99	0.99	0.99	217
weighted avg	0.99	0.99	0.99	217

Figure 14- heatmap for bagging



Ada Boosting 5.2

می توان از آن در ارتباط با بسیاری از انواع دیگر الگوریتم های یادگیری برای بهبود عملکرد استفاده کرد. خروجی سایر الگوریتم های یادگیری یادگیرندگان ضعیفدر یک جمع وزنی ترکیب می شود که خروجی نهایی طبقه بندی کننده تقویت شده را نشان می دهد. AdaBoost به این معنا تطبیقی است که یادگیرندگان ضعیف بعدی به نفع مواردی که توسط طبقه‌بندی‌کننده‌های قبلی به اشتباه طبقه‌بندی شده‌اند بهینه‌سازی می‌شوند. در برخی مسائل، نسبت به سایر الگوریتم‌های یادگیری، می‌تواند کمتر مستعد مشکل بیش‌برازش باشد. تک تک یادگیرندگان می توانند ضعیف باشند، اما تا زمانی که عملکرد هر یک کمی بهتر از حدس زدن تصادفی باشد، می توان ثابت کرد که مدل نهایی به یک یادگیرنده قوی همگرا می شود.

دقت اولیه مدل:

```
Accuracy
Mean : 0.989
Std : 0.009
Accuracy on testing data:  0.9907834101382489
Accuracy on training data:  1.0
```

هایپر پارامتر ها:

هایپر پارامتر های n_estimatores learning_rate , algorithm برای این دسته بند در نظر گرفته شده است.

N_estimatores: حداکثر تعداد برآوردگرهایی که boosting در آنها خاتمه یافته است. در صورت تناسب کامل، روند یادگیری زودتر متوقف می شود.

Learning_rate: وزن اعمال شده برای هر طبقه بندی کننده در هر تکرار boosting. نرخ یادگیری بالاتر سهم هر طبقه بندی کننده را افزایش می دهد.

Algorithm: اگر "SAMME.R" باشد از الگوریتم real boosting استفاده می کند. اگر "SAMME" باشد، از الگوریتم descrete boosting استفاده میکند. الگوریتم SAMME.R معمولاً سریعتر از SAMME همگرا می شود و خطای تست کمتری را با تکرارهای تقویتی کمتر به دست می آورد.

گزارش دهی دقت دسته بند بعد از تعیین هایپر پارامتر ها بوسیله grid search:

بهترین مقادیر به صورت زیر است:

```
Best: 0.988236 using {'algorithm': 'SAMME.R', 'learning_rate': 1.0, 'n_estimators': 500}
```

نتایج دسته بند در 2 قالب زیر آورده شده است(بعد از تعیین هایپر پارامتر ها):

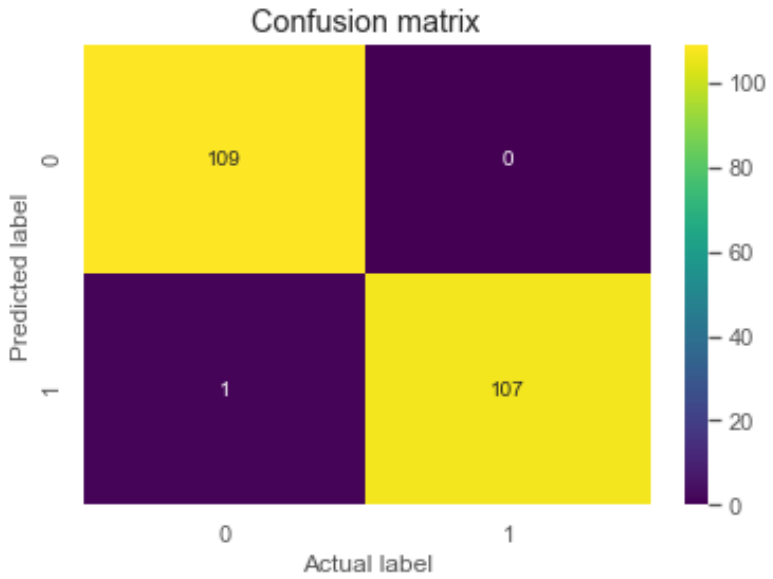
```
              precision    recall  f1-score   support

           0       0.99      1.00      1.00       109
           1       1.00      0.99      1.00       108

    accuracy                           1.00       217
   macro avg       1.00      1.00      1.00       217
weighted avg       1.00      1.00      1.00       217

1.0
0.9953917050691244
```

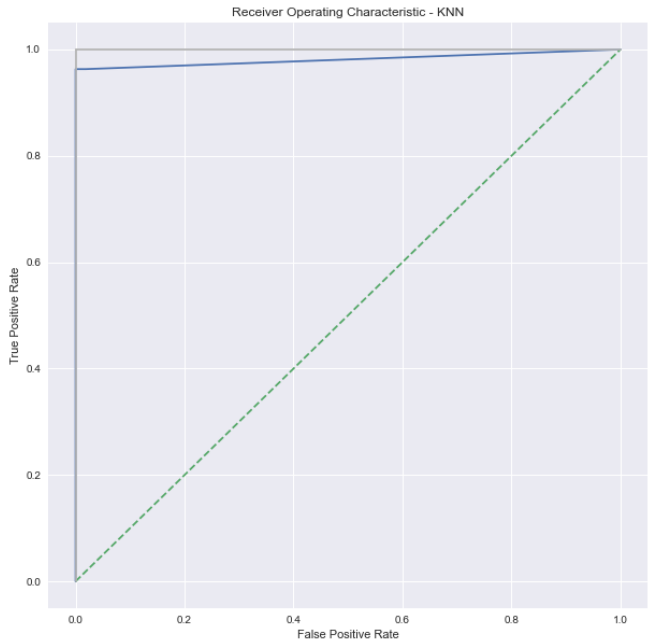
Figure 15- heatmap for boosting

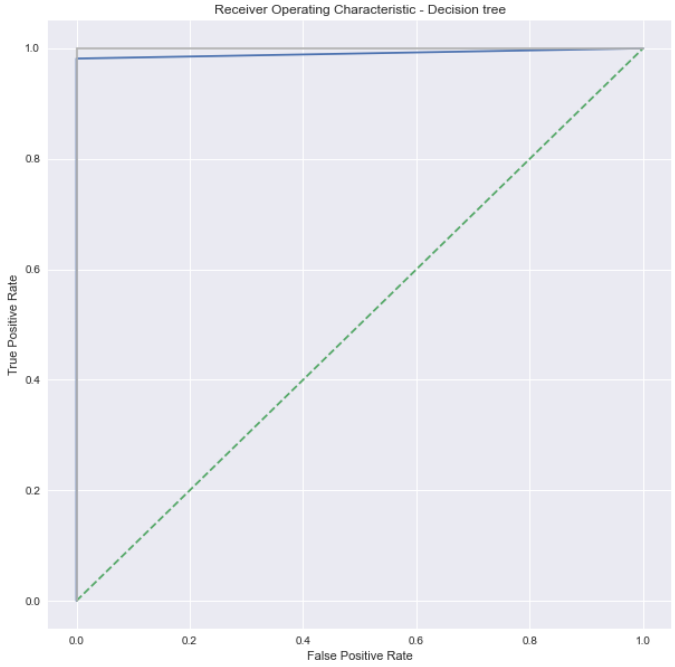


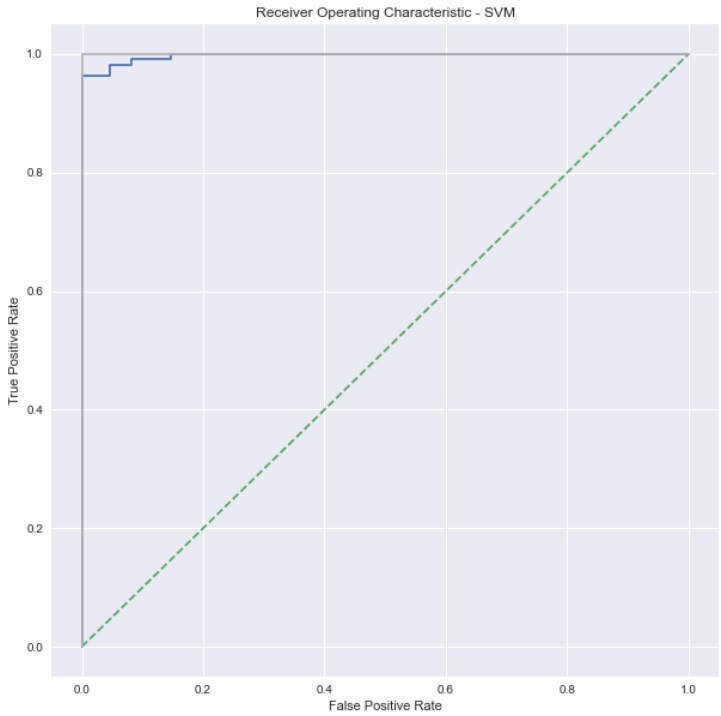
استدلال: از آنجایی که دقت تمام دسته بند ها نسبتا بالا است اینطور برداشت میشود که داده های ورودی از کیفیت بالایی برخوردار بوده اند.

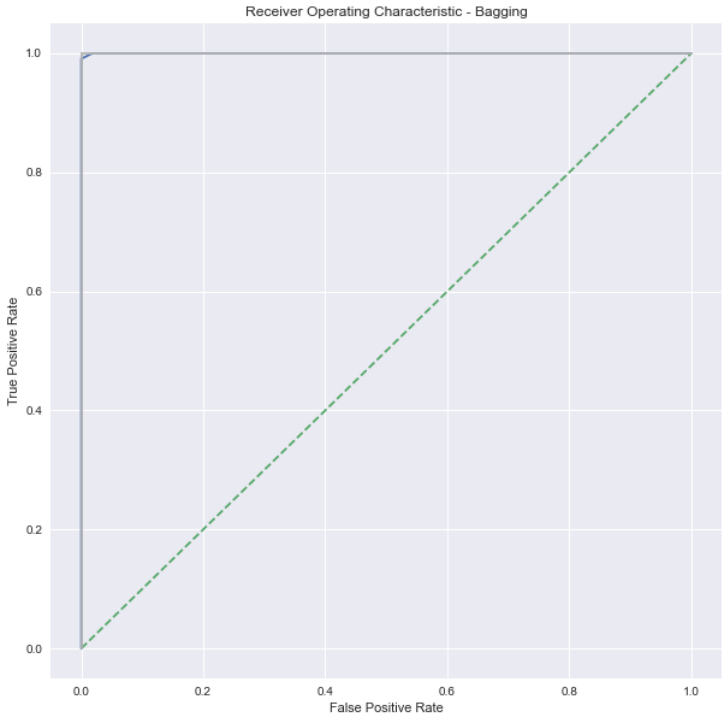
3.مقایسه بین دسته بند ها

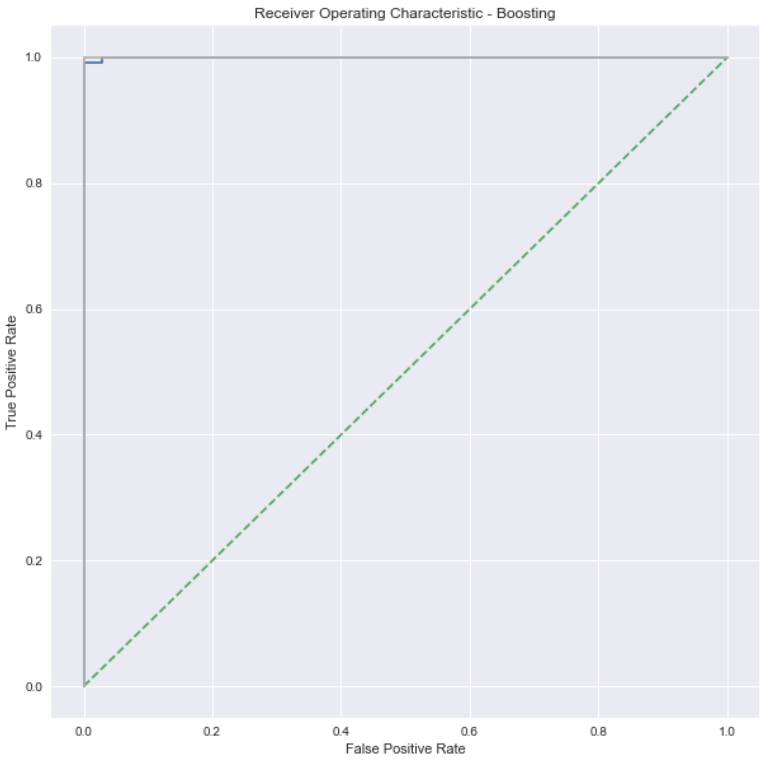
برای مقایسه بین دسته بند ها در ابتدا نمودار های ROC آن ها رسم شده است و سپس بر طبق هر نمودار یک امتیاز دریافت کرده اند (AUC) که مبنای مقایسه را همان قرار میدهیم. نمودار ها در زیر آورده شده است.


Receiver Operating Characteristic - KNN
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0


Receiver Operating Characteristic - Decision tree
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0


Receiver Operating Characteristic - SVM
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0


Receiver Operating Characteristic - Bagging
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0



تحلیل نمودار های ROC:

یکی از روش‌های بررسی و ارزیابی عملکرد دسته‌بندی دو دویی، نمودار مشخصه عملکرد Receiver Operating Characteristicیا به اختصار منحنی ROC است .کارایی الگوریتم‌های «دسته‌بندهای دو دویی Binary Classifier معمولا توسط شاخص‌هایی به نام حساسیت (Sensitivity) یا بازیابی (Recall) سنجیده می‌شود. اما در نمودار ROC هر دوی این شاخص‌ها ترکیب شده و به صورت یک منحنی نمایش داده می‌شوند. اغلب برای بررسی کارایی الگوریتم‌های دسته‌بندی یا ایجاد داده‌های رسته‌ای از منحنی ROC استفاده می‌کنند. این موضوع در شاخه یادگیری ماشین با نظارت (Supervised Machine Learning)، بیشتر مورد توجه قرار گرفته است.

از آنجایی که تمام دسته بند های ما دقت بالایی داشتند در تمام نمودار ها منحنی ما در ناحیه مطلوب است و نزدیک به نقطه (0 , 1) است.

امتیاز های داده شده بر اساس مساحت زیر هر نمودار:

```
roc_auc_score for KNN:  0.9811416921508664
roc_auc_score for DecisionTree:  0.9907407407407407
roc_auc_score for SVM:  0.9970268433571186
roc_auc_score for Bagging:  0.9999150526673461
roc_auc_score for Boosting:  0.9997451580020388
```

طبق امتیاز های بالا bagging بهترین دسته بند ما است.

یک نمودار میله ای نیز طبق امتیاز های بالا کشیده شده است که بهتر قابل مشاهده باشد: