سوال ۶)

۱ . تحلیل و آماده سازی داده ها :

در این بخش هر یک از ستون ها را مورد بررسی قرار داده و تغییرات مورد نیاز را روی آن ها انجام می دهیم.

۱ . ۱. شناسه مسافر: این عدد شناسه هر مسافر است و با توجه به اینکه هیچ اهمیتی در زنده ماندن یا مرگ فرد ندارد ، این ستون را حذف می کنیم.

۱ . ۲. زنده ماندن : این متغیر همان متغیر کلاس است بنابراین آن را حذف کرده و در زمان آموزش از آن استفاده خواهیم کرد.

۱ . ۳. Pclass: این متغیر یک عدد بین ۱ و ۲ و ۳ می باشد و نیازی به ایجاد تغییر در آن نیست.

۱ . ۴. اسم: با توجه به اینکه اسم شخص هیچ اثری در زنده ماندن یا مرگ فرد ندارد ، این ستون را حذف می کنیم.

۱ . ۵. جنسیت: در این ستون مقادیر male را با ۱ و مقادیر female را با ۰ عوض می کنیم.

۱ . ۶. سن: با توجه به اینکه مقدار عددی سن مناسب این کار نیست. سن را به ۴ بازه زیر ۱۶ سال ، زیر ۳۲ سال ، زیر ۴۸ سال و بالا ۴۸ سال تقسیم می کنیم. که مشخصا این تقسیم بندی ارتباط مستقیمی با توان بدنی افراد در آن سن برای زنده ماندن دارد.

۱ . ۷.Sib sp و parch: این دو متغیر هر دو نشان دهنده وجود افراد دیگر به همراه این فرد می باشند. هر دو را ترکیب کرده و متغیر جدیدی به نام isAlone می سازیم که در صورتی که فرد حداقل یک نفر دیگر همراه خود داشته باشد، مقدارش ۱ خواهد بود. در غیر این صورت نیز مقدار ۰ را خواهد داشت.

۱ . ۸. بلیت : با وجود آنکه منطقا این ویژگی نیز تاثیری ندارد اما آن را حذف نکرده ایم. مقدار عددی هر بلیت را بدست آورده (در بلیت های دارای حروف بخش عددی آن ها در نظر گرفته شده) سپس مقدار آن را به دو بازه کوچکتر از ۱۰۰ هزار و بزرگتر از ۱۰۰ هزار تقسیم می کنیم. به صورت کلی افرادی که شماره بلیت شان بیش از ۱۰۰هزار است محدود تر هستند.

۱ . ۹. Fare: متغیر بسیار مهمی است. مقادیر این متغیر را به سه بازه کمتر از ۱۵ ، کمتر از ۵۰ و بزرگتر از ۵۰ تبدیل می کنیم. غالب افراد گروه اول مرده اند و همچنین اکثر افراد گروه سوم نیز زنده مانده اند.

۱ . ۱۰.کابین: اکثر مسافران کابین ندارند بنابراین این متغیر را به حالت ۰ و ۱ در نظر می گیریم و به کسانی که کابین دارند مقدار ۱ و به کسانی که کابین ندارند مقدار ۰ را نسبت می دهیم. اکثر افراد دارای کابین زنده مانده اند.

۱ . ۱۱. Embarked: جایی است که هر یک از مسافران وارد کشتی شده اند. منطقا تاثیری نباید داشته باشند ولی این متغیر را نیز در نظر گرفته و به s مقدار ۰ ، c مقدار ۱ و q مقدار ۲ را نسبت می دهیم.

۲ . آموزش و دسته بندی :

پس از تبدیل کردن داده ها به فرم مناسب ، این درخت تصمیم را با استفاده از کتابخانه sklearn می سازیم. سپس داده های تست را به درخت داده و برچسب های خروجی را از آن می گیریم.

با انجام تعداد زیادی تست که هر یک از آن ها با انجام تغییرات مختلف در متغیر ها مانند تغییر بازه بندی ها ، تغییر ماکسیمم تعداد برگ ها (عمق) ، ادغام متغیر ها ، حذف متغیر ها و همچنین تغییر معیار بین gini و entropy بوده است. به دقت های مختلفی رسیده ایم این دقت ها از حدود ۷۳ تا ۷۹ متغیر بوده اند. در نهایت نیز متغیر ها را به نوعی که در قسمت قبلی گفته شد تبدیل کرده و تست ها را انجام داده ایم که دقت های آن در شکل زیر آمده است.

در ساخت درخت به جای حداکثر عمق ، از حداکثر تعداد برگ استفاده شده است که نتایج بهتری می دهد.

Submission and Description	Public Score
output.csv a few seconds ago by erfannnnnnnn add submission details	0.78708
output.csv 2 minutes ago by erfannnnnnnn add submission details	0.78229
output.csv 3 minutes ago by erfannnnnnnn add submission details	0.78229
output.csv 5 minutes ago by erfannnnnnnn add submission details	0.78947

اولی از پایین مربوط به ماکسیمم برگ برابر با ۱۲ و معیار gini می باشد.

دومی مربوط به ماکسیمم برگ ۱۲ و معیار entropy می باشد.

سومی ماکسیمم برگ ۹ و معیار entropy

و چهارمی ماکسیمم برگ ۹ و معیار gini

همان طور که دیده می شود بهترین نتیجه در تمامی نتایج مربوط به ماکسیمم برگ ۱۲ و معیار gini می باشد. که دقت آن نزدیک به ۷۹ درصد است.

سوال ۷)

۱ . تحلیل و آماده سازی داده ها :

۱ . ۱. همان طور که در صورت سوال نیز بیان شده است داده های مربوط به ۳۰۳ بیمار در اختیار ما می باشد. با توجه به اینکه این داده ها بر اساس target مرتب شده هستند ، آنها ها را به صورت دستی به دو بخش train و test تقسیم می کنیم. (اگر فقط داده های انتهایی را در نظر بگیریم ، تمامی داده ها تست بر چسب صفر خواهند داشت.) پس برای این کار به این صورت عمل می کنیم که

داده های از شماره ۲ تا ۱۳۳ بعلاوه از شماره ۱۶۷ تا ۲۷۷ را برای train در نظر می گیریم و داده های ۱۳۴ تا ۱۶۶ و ۲۷۸ تا ۳۰۴ را نیز برای test در نظر می گیریم. در این تقسیم بندی نسبت ۸۰ به ۲۰ نیز تا حد خوبی رعایت شده است.

۱ . ۲. با بررسی مجموعه داده ها متوجه می شویم که مقدار null وجود ندارد. همچنین تمامی مقادیر عددی هستند پس نیازی به عددی کردن نیز نیست.

۱ . ۳. با توجه به اختلاف بین داده های هم نوع کم است و اختلاف بین داده های غیر همنوع زیاد (مثلا slope بین ۰ تا ۲ در حالی که سن حدودا بین ۳۰ تا ۸۰) پس نیاز به نرمال سازی داده ها کاملا ضروری است. برای نرمال سازی داده ها از روش های مختلف که در قسمت preprocessing از sklearn موجود است استفاده می کنیم. این روش ها عبارتند از :

۱ . ۳. ۱. استفاده از متد scale :

در این روش برای نرمال سازی میانگین را از تمامی داده های موجود کم کرده سپس بر انحراف معیار تقسیم می کنیم.

۱ . ۳. ۲. استفاده از متد min-max scale :

در این روش مینمم مقدار بین داده ها را از هر یک از داده ها کم می کنیم. سپس حاصل را بر اختلاف ماکسیمم و مینیمم داده ها تقسیم می کنیم. (تمامی اعداد بین ۰ و ۱ قرار می گیرند.)

۱ . ۳. ۳. استفاده از متد robust scale :

این روش مانند روش min max است با این تفاوت که به جای کل داده ها ، بازه بین چارک اول تا سوم استفاده می کنیم (یعنی ۲۵ درصد از بالا و ۲۵ درصد از پایین استفاده نمی شود.)

۱ . ۳. ۴. استفاده از متد normalizer :

این روش به حالت عادی نرمال سازی را انجام می دهد یعنی مقدار عددی داده را به اندازه آن در فضای n بعدی تقسیم می کند. (تمامی اعداد بین ۰ و ۱ خواهند شد)

۲ . ساخت و آموزش مدل :

۲ . ۱. روش KNN :

در این روش همان طور که می دانیم باید بر اساس n همسایه نزدیک ، داده را کلاس بندی کنیم. پیاده سازی و تست این روش با استفاده از کتابخانه sklearn کار سختی نیست پس به بیان نتایج پیاده سازی این روش با استفاده از هر یک از نرمال سازی هایی که در قسمت قبل گفته شد (به ترتیب) می پردازیم.

۲ . ۱. ۱. استفاده از متد scale :

با انتخاب مقدار ۷ برای تعداد همسایه ها دقت ۷۸ درصدی خواهیم داشت.

۲ . ۱. ۲. استفاده از متد min-max scale :

با انتخاب مقدار ۷ و همچنین ۱۲ برای تعداد همسایه ها دقت ۸۰ درصدی خواهیم داشت.

۲ . ۱. ۳. استفاده از متد robust scale:

با انتخاب مقدار ۱۲ برای تعداد همسایه ها دقت ۸۲ درصدی خواهیم داشت.

۲ . ۱. ۴. استفاده از متد normalizer:

با انتخاب مقدار ۸ برای تعداد همسایه ها دقت ۶۷ درصدی خواهیم داشت.

با بررسی روش های انجام شده در این قسمت می توان گفت که بهترین روش شماره ۳ سپس ۲ سپس ۱ و در انتها نیز روش ۴ است. که نتیجه در روش های ۱ و ۲و ۳ به یکدیگر نزدیک بوده ولی نتیجه در ۴ بسیار ضعیف تر است.

۲ . ۲. روش naïve bayes :

در این روش همان طور که می دانیم داده ها را بر اساس تابع نرمال گاوسی کلاس بندی کنیم. پیاده سازی و تست این روش با استفاده از کتابخانه sklearn کار سختی نیست پس به بیان نتایج پیاده سازی این روش ، با استفاده از هر یک از نرمال سازی هایی که در قسمت قبل گفته شد (به ترتیب) می پردازیم.

۲ . ۲. ۱. استفاده از متد scale :

به صورت کلی در این قسمت معیاری تغییر دهنده دیگری مانند knn نداریم. دقت در این حالت ۷۷ درصد می باشد.

۲ . ۲. ۲. استفاده از متد min-max scale :

دقت در این حالت ۷۵ درصد می باشد.

۲ . ۲. ۳. استفاده از متد robust scale:

دقت در این حالت ۷۷ درصد می باشد.

۲ . ۲. ۴. استفاده از متد normalizer:

دقت در این حالت ۷۲ درصد می باشد.

با بررسی روش های انجام شده در این قسمت می توان گفت که بهترین روش شماره ۳ و ۱ سپس ۲ و در انتها نیز روش ۴ است. که نتیجه در هر ۴ روش به یکدیگر نزدیک است.

به صورت کلی ما برای هر دوی این الگوریتم ها روش نرمال سازی robust scale را انتخاب می کنیم که به ما در هر دو بهترین جواب را می دهد.