{A, B}	B → A	5 / 6 = 0.83	**معتبر**
{E, B}	E → B	4 / 8 = 0.5	**معتبر**
{E, B}	B → E	4 / 6 = 0.67	**معتبر**
{A, E, B}	EB → A	3 / 4 = 0.75	**معتبر**
{A, E, B}	AB → E	3 / 5 = 0.6	**معتبر**
{A, E, B}	AE → B	3 / 5 = 0.6	**معتبر**
{A, E, B}	A → EB	3 / 7 = 0.43	**نا معتبر**
{A, E, B}	E → AB	3 / 8 = 0.37	**نا معتبر**
{A, E, B}	B → AE	3 / 6 = 0.5	**معتبر**

قواعد معتبر قابل استخراج در جدول بالا مشخص شده اند.

پیاده سازی :

سوال ۸)

در حل این سوال با توجه به اینه هدف مقایسه جنگل تصادفی با درخت تصمیم می باشد، از روش پیش پردازشی که در حل تمرین قبلی استفاده شد، استفاده می شود.

در تمرین قبلی دقت ها با استفاده از سایت kaggle محاسبه شدند اما با توجه به اینکه این سایت محدودیت آپلود در هر روز دارد، در این تمرین از تقسیم داده ها به train و test استفاده شده است.

برای انجام مقایسه بین حالت قبلی(درخت تصمیم) و حالت فعلی(جنگل تصادفی)، کد مربوط به پیاده سازی درخت تصمیم نیز در کد پیوست قرار دارد و با آن روش نیز می توانیم classifier مان را بسازیم. در بالای کد متغیری به نام tree_forest وجود دارد که اگر مقدار آن را ۰ بگذاریم، از درخت تصمیم استفاده می شود و اگر مقدار آن را ۱ بگذاریم از جنگل تصادفی استفاده می شود.

به صورت کلی دقت های بدست آمده در این حالت با استفاده از همان درخت تصمیم تمرین قبل بسیار بیشتر از دقت هایی است که در روش قبلی بدست آمدند.

مقایسه ها

۱ . به طور کلی دقت های مربوط به دو مدل در چهار حالت انتخابی در دو جدول زیر آمده است.

درخت تصمیم

دقت	معیار تقسیم	تعداد برگ های درخت(عمق)
۰,۹۰۱	gini	9
۰,۹۰۱	entropy	9
۰,۹۲	gini	12
۰,۹۰۱	entropy	12

جنگل تصادفی

دقت	معیار تقسیم	تعداد برگ های درخت(عمق)
۰,۹۲۶	gini	9
۰,۸۹۱	entropy	9
۰,۹	gini	12
۰,۹۳۰	entropy	12

همان طور که دیده می شود دقت در هر دو حالت بسیار بالا می باشد. در درخت تصمیم سه تا از دقت ها ۹۰ درصد و یکی از آنها ۹۲ درصد می باشد. و در درخت تصمیم دقت های متغیر از ۸۹ تا ۹۳ درصد داریم.

بنابراین بالاترین دقت بدست آمده توسط مدل جنگل تصادفی بیشتر از بالاترین دقت بدست آمده توسط درخت تصمیم می باشد.

۲ . انتظار می رود که هم زمان یادگیری و هم زمان تست برای جنگل تصادفی بسیار بیشتر از یک درخت تصمیم باشد.

با محاسبه زمان های تمرین و تست بر حسب نانو ثانیه می بینیم که زمان تمرین برای یک درخت تصمیم در حدود ۱۵,۶۰۰,۰۰۰ نانو ثانیه و برای جنگل تصادفی در حدود ۱۸,۶۰۰,۰۰۰ نانو ثانیه می باشد. که اختلاف کوچک تری نسبت به انتظار قبلی وجود دارد.

زمان تست در هر دو حالت بر حسب نانو ثانیه بسیار کوتاه و در حد ۰ می باشد. بنابراین در این حالت مقایسه ای نمی توانیم انجام دهیم و می توان گفت هر دو سرعت حدودا یکسان دارند.

سوال ۹)

۱ . پیش پردازش انجام شده در این قسمت نیز دقیقا مانند پیش پردازش انجام شده در تمرین قبل می باشد که توضیحات آن به طور کامل در تمرین قبلی آمده است. برای ارزیابی نیز از شیوه تقسیم داده ها به دو بخش استفاده می شود.

۲ . دقت مدل آموزشی SVM با هسته خطی، ۰,۷۶۸ می باشد که حدود ۷۷ درصد است.

۳ . همان طور که در سوال قبل دیدیم دقت مدل های درخت تصمیم و جنگل تصادفی هر دو در حدود ۹۰ درصد بود، در حالی که با SVM خطی دقت بسیار پایین تر ۷۷ درصدی داریم.

علت پایین تر بودن دقت در این حالت این است که SVM نمی تواند خط(ابر صفحه) بهتری پیدا کند که بتواند دقت را افزایش دهد. یعنی می توان گفت توانایی آن در حل این مسئله با کرنل خطی محدود است و از حد خاصی فراتر نخواهد رفت چون پراکندگی داده ها این اجازه را نمی دهد. اما درخت تصمیم و جنگل تصادفی هر دو شان می توانند تا حد مشخصی داده ها را از یکدیگر جداسازی کرده و در دسته های جدا قرار دهند در حقیقت محدودیت خاصی در عمق آنها وجود ندارد. اما برای عملکرد مناسب شان برایشان محدودیت عمق تعیین می شود. بنابراین محدودیت عمق اعمال شده درخت و جنگل برای بهتر شدن عملکرد آنهاست و ربطی به محدودیت مدل ندارد در نتیجه این دو مدل عملکرد بهتری دارند.

۴ . در کد امکان اعمال دو تابع rbf و poly به کرنل فراهم است. دقت در این دو این مدل بسیار به هم نزدیک است. دقت کرنل rbf ، ۰,۸۵ می باشد. و دقت کرنل polynomial ، ۰,۸۴۱ می باشد.

۵ . مشخص است که دقت با کرنل غیر خطی بهتر از کرنل خطی می باشد. علت آن نیز محدودیت های کرنل خطی است. همان طور که در قسمت قبل نیز بیان شده کرنل خطی تنها می تواند بین داده های در فضا یک ابر صفحه رسم کرده و با استفاده از آن به جداسازی داده ها بپردازد. اما اگر به عنوان مثال داده های دو کلاس ما با یک دایره از هم دیگر تفکیک پذیر باشند. این مدل عملکرد بسیار ضعیفی خواهد داشت. در این سوال با توجه به اختلاف ۷ ۸ درصدی بین کرنل خطی و غیر خطی اختلاف آنقدر فاحش نبوده است. اما به صورت کلی با کرنل غیر خطی می توان داده ها را به فضای بالاتر برد و تفکیک پذیری بهتری بین شان ایجاد کرد که در اینجا نیز همین اتفاق رخ داده است.