به نام خدا

ارائه پروژه های مربوط به svm و decision tree

علیرضا اسلامی خواه

99521064

استاد عبدی

ترم 1401-1402

در ابتدا:

پروژه مربوط به درخت تصمیم

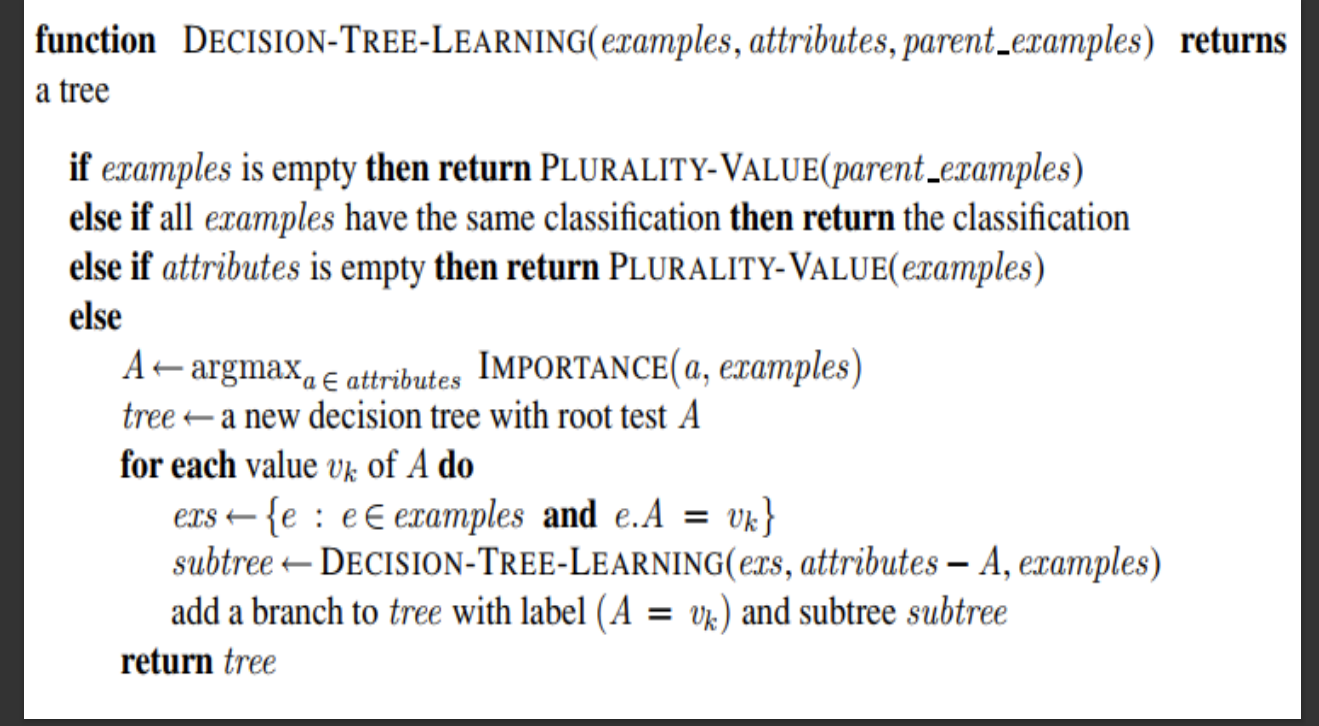
اول : درخت مربوط به داده های رستوران

در ابتدا ما یک فایل اکسل برای ذخیره داده ها برای استفاده از آنها درست کردیم که مانند جزوه میبود.



سپس شروع به درست کردن درخت تصمیم گرفتیم.

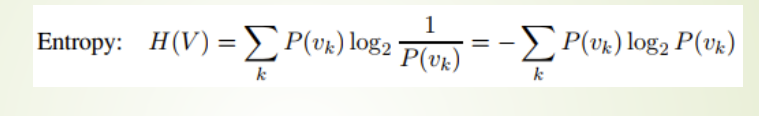
اصلی ترین تابع درخت تصمیم همانطور که در شکل زیر هم نشان داده میشود تابع مربوط به درخت تصمیم است.



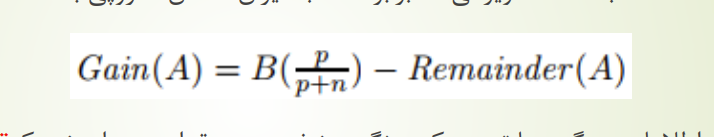
که ما این تابع را به همین اسم پیاده سازی کردیم.

!! توجه جزئیات تمام کدها به صورت کامنت در کد نوشته شده است و اینجا موضوعات کلی گفته میشود.!!

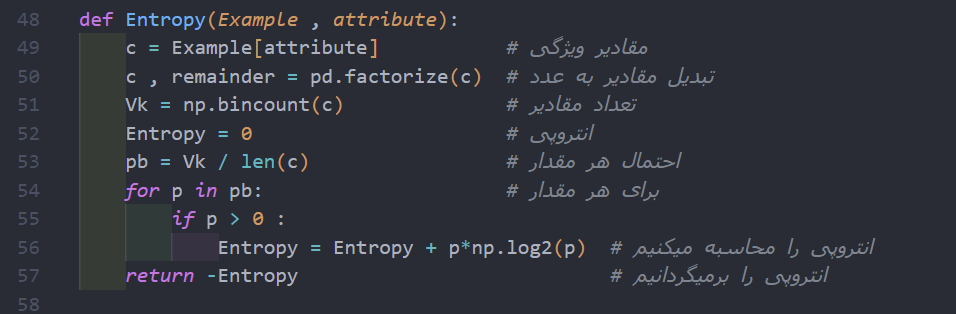
تابع بعدی که بسیار مهم بود تابعی به اسم entropy بود که مانند شکل زیر پیاده سازی شد.



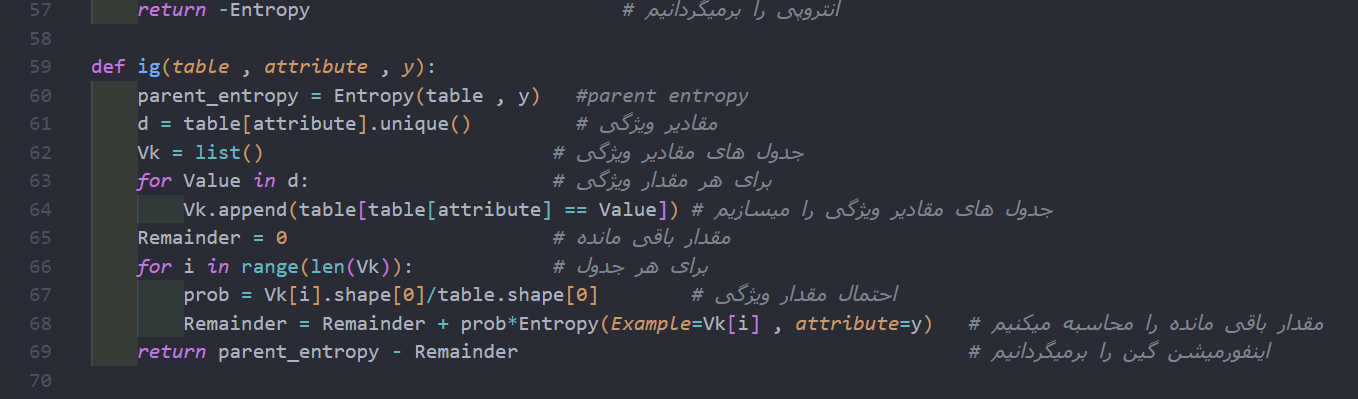
که بعدا هم در بدست آوردن gain یا همان ارزش هر ویژگی مورد استفاده قرار گرفت.



کد آن مانند شکل زیر پیاده سازی شده است.

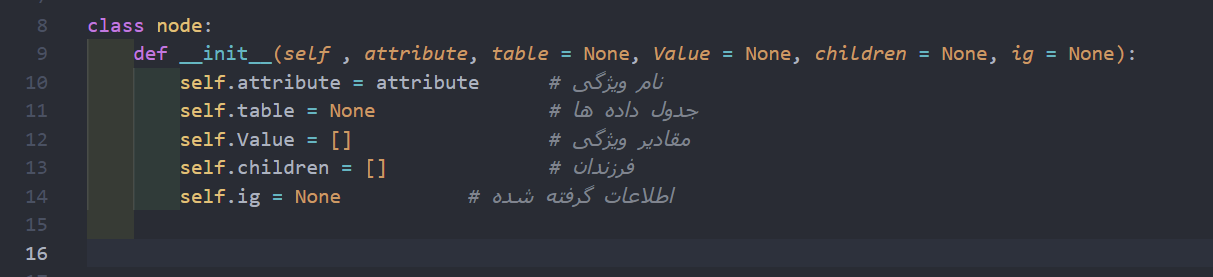


سپس به پیاده سازی gain یا همان معیار تشخیص ویژگی بعدی پرداختیم.



در کنار همه این توابع ما نیاز داشتیم گره ها را با استفاده از کلاسی تعریف کنیم که یک ساختار کلی داشته باشد.

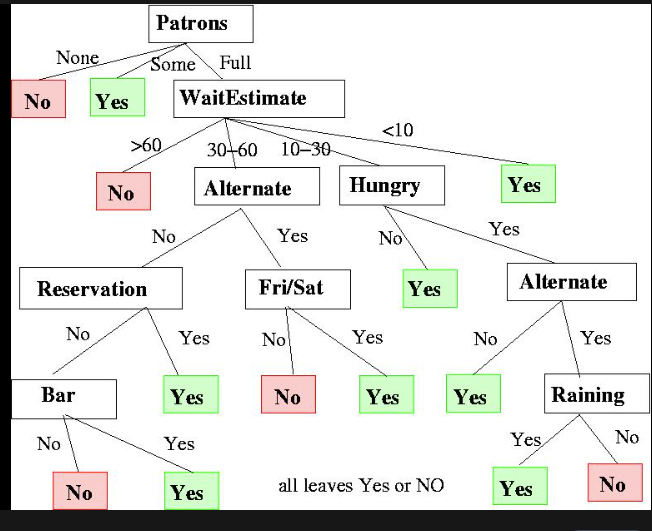
این ساختار شامل بچه ها ، ویژگی ، ارزش ها و .. میباشد.



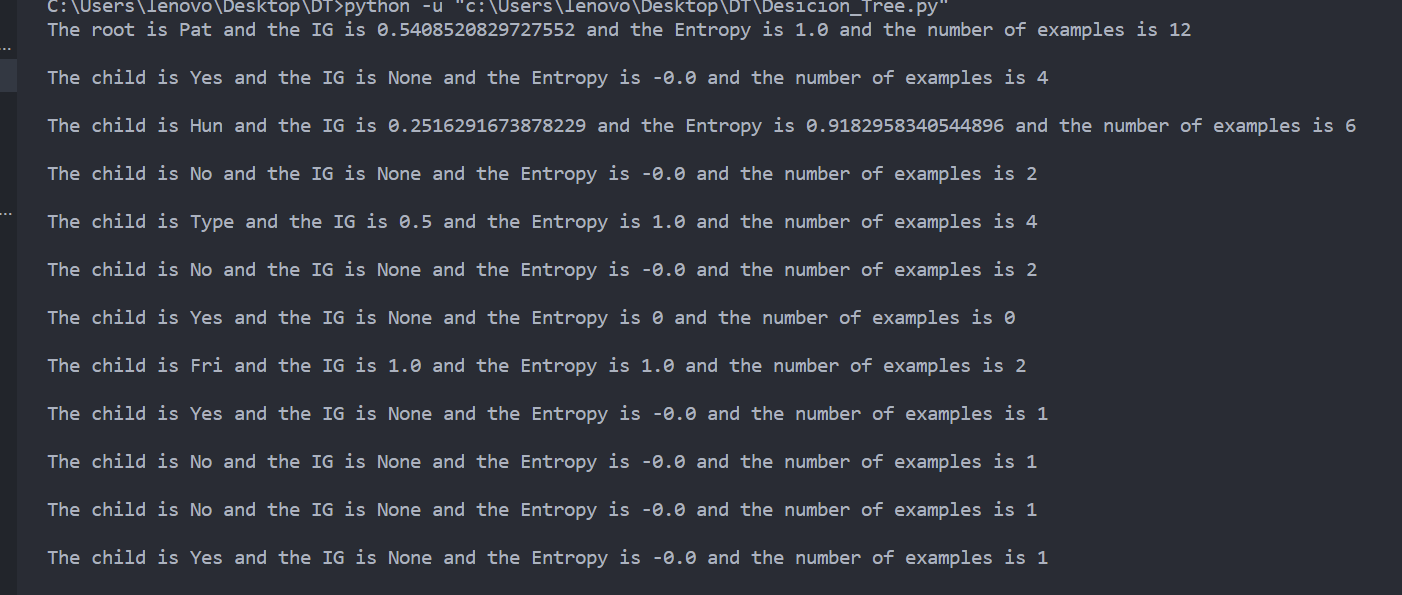
توابع classification و plurarity هم مانند آنچه در جزوه گفته شد پیاده سازی شدند.



در انتها هم به ران کردن کد و رفع اشکال های آن پرداختیم که همچین خروجی را به ما داد.



که به صورت دقیق تر ترمینال خروجی هر یک از گره ها چاپ میشود.



که اطلاعاتی را شامل خود ویژگی گره و gain آن و آنتروپی آن به ما میدهد. در انتها هم تعداد نمونه ها یا همان examples را میدهد.

درخت تصمیم برای نمونه دیابت:

تمامی بخش ها شبیه بخش قبل بود با چند تفاوت:

اول اینکه ما نیاز داشتیم داده ها را گسسته سازی کنیم. برای این کار روش زیر را انجام دادیم :

در این تابع ابتدا بعد از گرفتن ستون دلخواه ابتدا کوچکترین و بزرگترین را

    در آن مشخص کرده و سپس با استفاده از این دو متغیر مقدار گام را محاسبه میکنیم

    دانستن مقدار گام به ما کمک میکند که بفهمیم چه مقادیری در چه بازه ای قرار دارند

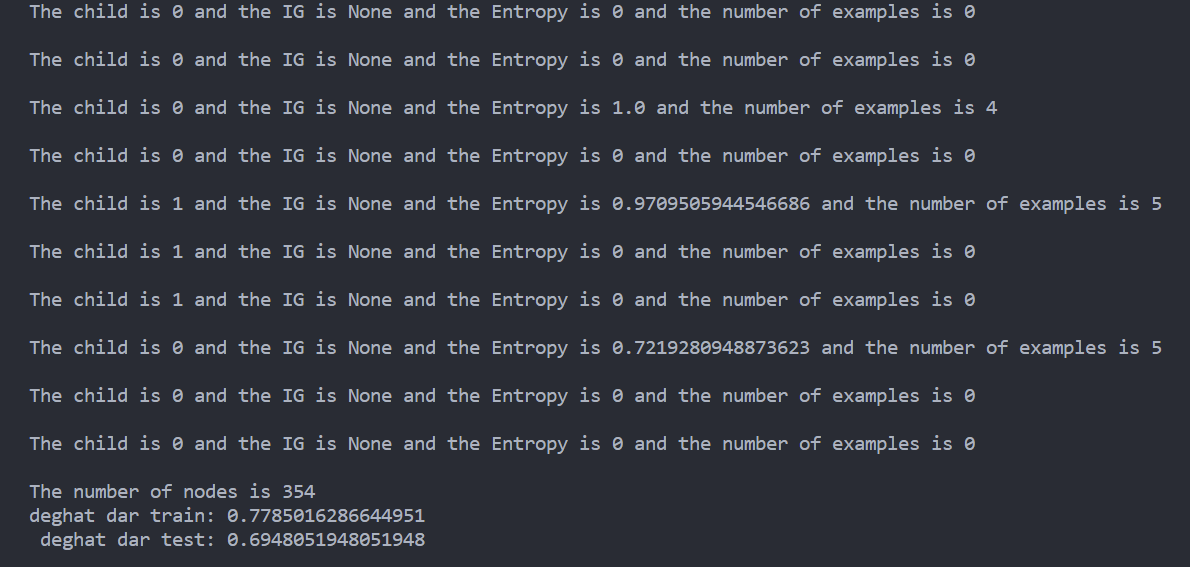
    برای گذاشتن مقادیر در بازه های مختلف از حلقه های دوتایی استفاده کردیم

    و همانگونه که در کد پایین هم قبل مشاهده است ما مقادیر را در بازه های مناسب قرار میدهیم

منظور از گسسته سازی اینست که ما مثلا برای داده ها یک بازه تعیین کردیم و با استفاده از 🡪 گفتیم که عدد مشخص شده در سطر در این بازه قرار دارد.

گسسته سازی باعث شد ما بتوانیم به نوعی مقادیر را صفر و یکی کنیم.

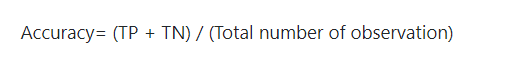
نمونه خروجی کد برای دیابت:



که در دو خط آخر با استفاده از تابع fit که مشخص کردیم میزان دقت درخت های ساخته شده را سنجیدیم.

توجه شود که در تابع main تعریف شده ما 80 درصد را برای آموزش و 20 درصد را برای آزمایش تعریف کردیم.

میزان دقت درخت ها هم با استفاده از همین فرمول بدست آوردیم.

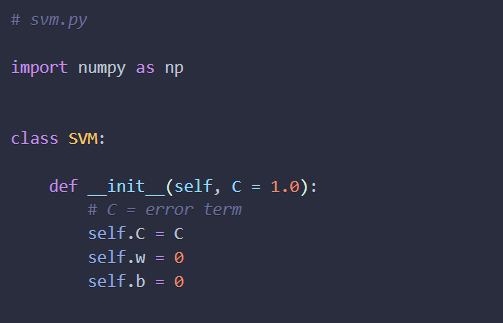


پروژه مربوط به svm

هدف اصلی SVM به حداکثر رساندن حاشیه بین دو کلاس مختلف است. این بدان معناست که شما می‌خواهید مطمئن شوید که به تعداد نقاط یک کلاس در یک طرف مرز تصمیم‌گیری و به همان تعداد نقطه در کلاس دیگر در طرف دیگر قرار دارند.

هنگامی که این اتفاق می افتد، تمام نقاط با درجه جدایی بالاتر به درستی طبقه بندی می شوند در حالی که تمام نقاط با درجه جدایی پایین تر به اشتباه طبقه بندی می شوند.

ابتدا یک کلاس SVM ایجاد کردیم و مقداری را مقداردهی اولیه کردیم. عبارت C به عنوان عبارت خطا شناخته می شود که برای بهینه سازی تابع باید اضافه کنیم



سپس به بررسی تابع fit میپردازیم:

آنچه در این روش برازش اتفاق می افتد واقعاً ساده است، ما در تلاش هستیم تا تلفات را در تکرارهای متوالی کاهش دهیم و بهترین پارامترهای w و b را پیدا کنیم. توجه داشته باشید که در اینجا از Batch Gradient Descent استفاده می کنیم. وزن‌ها (w) و سوگیری (b) در هر تکرار با استفاده از گرادیان‌ها و نرخ یادگیری به‌روزرسانی می‌شوند که منجر به به حداقل رساندن ضرر می‌شود. هنگامی که پارامترهای بهینه یافت می شوند، روش به سادگی آن را همراه با تلفات برمی گرداند.

سپس با استفاده از کتابخانه mtplotlib شروع به به تصویر کشیدن داده ها و محدوده آن ها کردیم.

نمونه خروجی نشان داده شده:

