به نام خدا

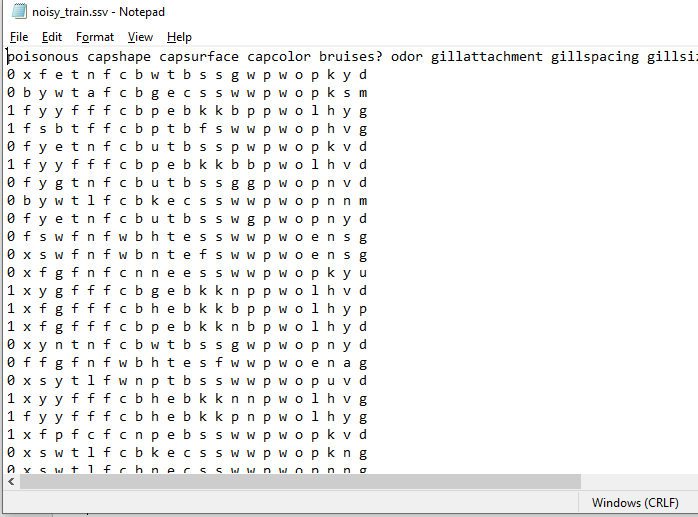
حسین غلامی تمرین سری دو یادگیری ماشین – درخت تصمیم

توضحیات

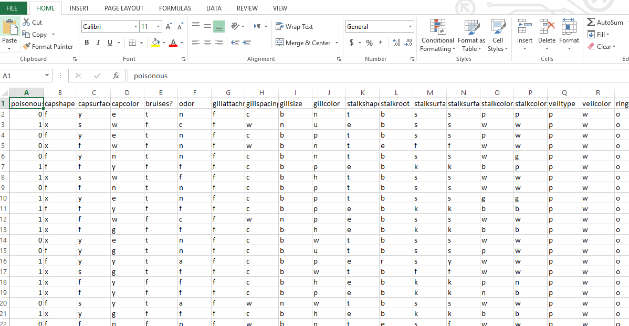
در انجام این تمرین از آن جایی که فرمت اطلاعات به صورت ssv بودند به جهت ساده سازی و استفاده از متد های pandas

اطلاعات را به فرمت csv در اوردم

نحوه تبدیل:

 ابتدا فایل را با notpad باز کرده ، و برای اینکه به فرمت dataframe، pandas در بیاید ، فایل های اضافی را حذف نموده

سپس فاصله را در این فرمت با کاما (,) جایگزین نموده و فایل را یا فرمت csv ذخیره میکنیم

خروجی به شکل زیر حاصل شد از آن جایی که فایل csv را در اکسل نیز میتوان مشاهده کرد ، تصویر آن نیز قرار گرفت

پیاده سازی الگوریتم :

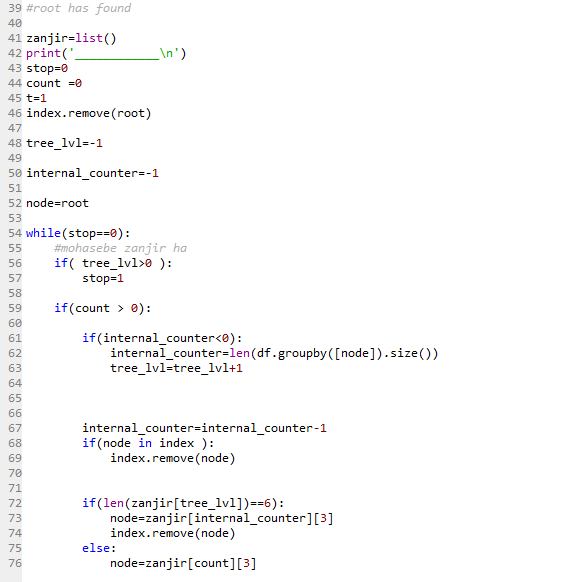
در سه مرحله انجام شد:

1. پیاده سازی اولیه
2. بررسی کتابخانه ها و عملکردشان و نوع کارکردن با آن ها
3. پیاده سازی نهایی

پیاده سازی اولیه

به این صورت که در ابتدا ریشه درخت مشخص میشود و در ادامه شاخه ها و نود ها پس از هم محاسبه میشوند. ولی این که پس از تمام شدن درخت چگونه از روی شاخه ها بازگشت ، قابل پیاده سازی نبود ، چرا که نام ویژگی های یکسانی وجود داشت و پیاده سازی امکان پذیر نبود







که به خروجی زیر رسیدم که شاخه های درخت را تشکیل میداد ولی برای محاسبه درخت مشکل داشت:

.....

2-1)در مرحله بعد برای بررسی دقیق سایرین و فرایندی که درآینده استفاده خواهد شد ، به بررسی کتابخانه های موجود پرداختم که با جست و جویی ساده به دو کتابخانه Sklean.tree و Id3 رسیدم ، که هر کدام را توضیح و مزایا و معایب آن ها را از دیدگاه خودم بررسی میکنم

Sklearn.tree کتابخانه ای کامل که علاوه بر روش entropy میتوان از روش gini نیز استفاده نمود. توانایی pruning

وجود ندارد. همجنین یکی از مشکلات اصلی بحث Passing categorical data to Sklearn Decision Tree است که

نمیتوان به این تابع مقدار کمی مانند ‘a’ نسبت داد و باید همه اطلاعات عدد باشند. راه حل آن این است که از

from sklearn import preprocessing

le = preprocessing.LabelEncoder()

le.fit(["paris", "paris", "tokyo", "amsterdam"])

le.transform(["tokyo", "tokyo", "paris"])

استفاده کرد و اطلاعات را به عدد تبدیل نمود و سپس از طریق

list(le.inverse\_transform([2, 2, 1]))

به حالت اولیه برگرداند.

یک نمونه در ضمیمه قرار گرفت استفاده در ضمیمه قرار گرفت.

2-2 ) id3 کتابخانه و کلاس بعدی است که میتوان با نام decision-tree-id3 آن را نصب نمود. که بعد از اجرا خروجی .dot

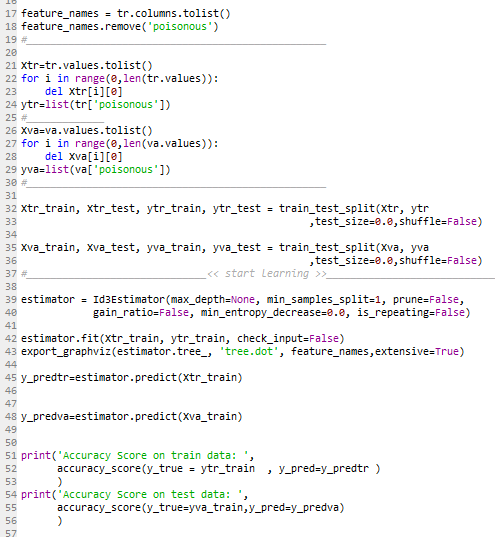
میدهد ، همچنین میتوان خروجی .dot را با graphviz به فرمت png برده و نمایش داد.

از لینک زیر برای نصب graphviz می­توان استفاده کرد :

<https://bobswift.atlassian.net/wiki/spaces/GVIZ/pages/20971549/How+to+install+Graphviz+software>

باتوجه به نوع اطلات موجود استفاده از id3 مناسب تر می باشد چرا که علاوه پذیرفتن و قبول کردن اطلاعات کتگورکال میتواند pruning را انجام دهد.

حال به بررسی اطلاعات و یافتن بهترین سطح هرس میپردازیم



بدون prune نتیجه به روی داده validation به شرح زیر است ،



حال prune را فعال نموده ونتیجه :



حال purne را مجدد غیرفعال نموده و محدودیت را روی تعداد عمق درخت میبریم ، عمق درخت ما 11 است. پس به کم کردن آن میپردازیم تا جایی که میزان دقت اطلاعات validation حداکثر شود. اگر عمق درخت را 2 قرار دهیم به  میرسیم که بیشترین میزان دقت در داده تست است.

حالت های مختلفی بررسی شد ، که بهرین نتیجه را بر روی داده validation از طریق کم کردن شاخه ها دریافت شد.

حال به رسم نتایج میپردازیم

فایل .dot را با کمک graphiz که پیشتر نصبش توضیح داده شد ، توسط command

dot file.dot -Tpng -o image.pngt در cmd به فایل png تبدیل میکنیم

درخت شماره 1



درخت شماره2

درخت شماره3

\*\* تصاویر در ضمیمه قرار گرفت ، اینجا نمیشد بزرگ کرد.

حال اطلاعات train شده را برای بهترین حالت یادگیری (تغییر عمق شاخه بررسی می­کنیم) که نتیجه زیر حاصل شد:



روش سوم

با توجه به مرحله قبل دریافتیم که بهترین پاسخ را میتوان از تعیین عمق شاخه بدست آورد.

پس به پیاده سازی کامل بدون استفاده از کتابخانه id3 پرداخته شد . به صورتی که تابعی با ورودی های 1- دیتا فریم 2-لیست ویژگی ها 3- عمق نفوذ ، نوشته شد، برای محاسبه بهترین عمق نفوذ ، ابتدا نتیجه را روی داده validation بررسی کرده و درجایی که بهترین نتیجه را بگیریم ، روی داده تست اجرا میکنیم:

برای بررسی بهترین عمق نفوذ چند مثال را بررسی میکنیم:

مثال 1:

tree = id3(df\_shroom, attribute\_names,depth\_of\_tree=None)

که خروجی آن:

Accuracy of train is 1.0

Accuracy of validation is 0.7672770230360307

مثال 2:

tree = id3(df\_shroom, attribute\_names,depth\_of\_tree=10)

که نتایج زیر حاصل شد:

Accuracy of train is 0.9091714665485158

Accuracy of validation is 0.8470171293561725

مثال 3:

tree = id3(df\_shroom, attribute\_names,depth\_of\_tree=5)

که نتایج زیر حاصل شد:

Accuracy of train is 0.9020824102791316

Accuracy of validation is 0.8865918487891317

مثال 4:

tree = id3(df\_shroom, attribute\_names,depth\_of\_tree=3)

که نتایج زیر حاصل شد:

Accuracy of train is 0.898980948161276

Accuracy of validation is 0.8907265209686946

مثال 5:

tree = id3(df\_shroom, attribute\_names,depth\_of\_tree=2)

که نتایج زیر حاصل شد:

Accuracy of train is 0.898980948161276

Accuracy of validation is 0.8907265209686946

مثال 5:

tree = id3(df\_shroom, attribute\_names,depth\_of\_tree=1)

که نتایج زیر حاصل شد:

Accuracy of train is 0.8932210899424015

Accuracy of validation is 0.8753691671588896

همانگونه که مشاهده میشود ، برای درختی دو یا سه سطحی ، بهترین نتیجه را داریم ،در زیر درخت دو سطحی نمایش داده میشود :

{'odor': {'a': {'sporeprintcolor': {'k': 0, 'n': 0, 'u': 0}},

'c': {'sporeprintcolor': {'k': 1, 'n': 1}},

'f': {'sporeprintcolor': {'h': 1}},

'l': {'sporeprintcolor': {'k': 0, 'n': 0, 'u': 0}},

'm': {'sporeprintcolor': {'w': 1}},

'n': {'sporeprintcolor': {'k': 0, 'n': 0, 'r': 1, 'w': 0}},

'p': {'sporeprintcolor': {'k': 1, 'n': 1}}}}

حال به بررسی اطلاعات مربوط به دادگان تست میپردازیم :

که نتیجه زیر حاصل شد :

Accuracy of train is 0.898980948161276

Accuracy of validation is 0.8907265209686946

Accuracy of test is 0.9037780401416765

که درصد ملاحظه شده ای نسبت به درخت بدون هرس ، یا عمق نفوذ بهبود یافته

همچنین کد بخش آخر در ادامه آمده است :

