نام و نام خانوادگی دانشجو: محمدامین احمدی رشته :کامپیوتر شماره دانشجویی: 9912358001

نام استاد : دکتر منصوری زاده

موضوع :تحلیل داده های چاقی


بسم الله الرحمن الرحيم

فهرست مطالب

مقدمه 4

شرح تمرین 5

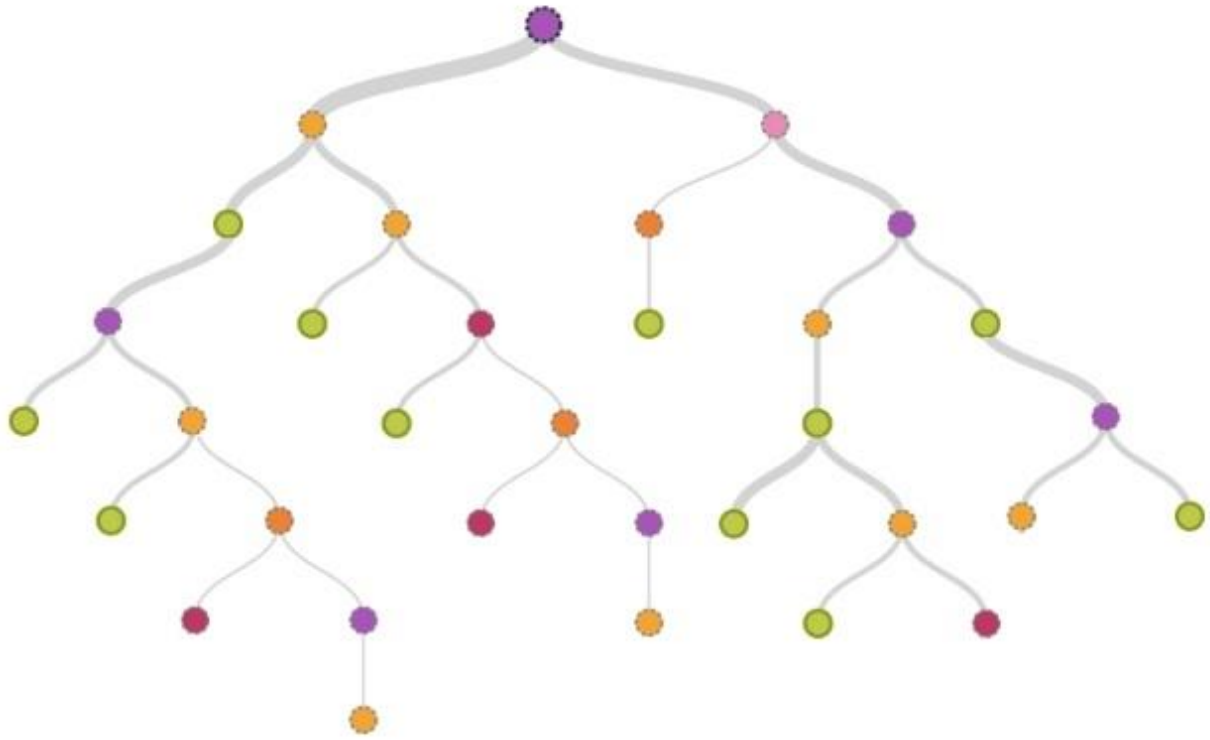
معیار ها 6

مقایسه ی میزان دقت هر مدل 7

منابع 8

مقدمه

یک درخت تصمیم، مدلی است که برای حل وظایف «دسته‌بندی (Classification)» و «رگرسیون (Regression)» مورد استفاده قرار می‌گیرد. مدل، امکان تولید خروجی‌های گوناگون را فراهم کرده و امکان انجام تصمیم‌گیری با داده‌ها را فراهم می‌کند. در تمرین بیان شده در این گزارشکار، تاثیر «متغیرهای توصیفی (explanatory variables)»(از جمله میزان مصرف کلسیم، زمان تلویزیون دیدن، فعالیت بدنی در هفته، استفاده از سیگار، مصرف آب، مصرف غذا در وعده‌های غذایی،تعداد وعده‌های غذایی در روز،مصرف سبزیجات در روز و مصرف میوه در روز) در تشخیص نوع چاقی فرد تحلیل خواهد شد. شایان توجه است که این تمرین، از فایل ObesityDataSet.csv برای دیتاست خود استفاده می کند.



پیش پردازش داده ها با پایتون:

هسته اصلی یادگیری ماشین پردازش داده ها است. قبل از شروع کار با الگوریتم های یادگیری ماشین داده ها باید آماده شوند تا دقت و خروجی کار بالاتر رود.

- گام اول: وارد کردن کتابخانه های مورد نیاز:
  - زمانی که می خواهیم پیش پردازش داده ها انجام دهیم معمولا با کتابخانه pandas پایتون کار می کنیم. این کتابخانه برای وارد کردن داده ها و مدیریت آن ها بسیار پرکاربرد است.
- گام دوم: برسی داده ها و هندل کردن مقادیر NULL:
  - استفاده از میانگین برای داده های عددی.
  - استفاده از مُد برای داده های کیفی.
- گام سوم نرمال سازی ستون ها:
  - از تنظیم کردن مقیاس مقادیر ستون های مختلف.
- گام چهارم: تبدیل داده های کیفی(categorical):
  - داده های کیفی برای محاسبات راحت تر بهتر است که به صورت عددی دربیایند.
- گام پنجم: تقسیم بندی داده ها به دو قسمت آموزش و تست.
- گام ششم : ساخت مدل

گام اول

با دستور :

df = pd.read_csv('ObesityDataSet.csv')

فایل دیتا را می خوانیم.

گام دوم

داده هایی که به دست می آوریم بندرت همگن است. گاهی اوقات داده ها ممکن است دارای داده از دست رفته باشند و باید به آنها رسیدگی شود تا عملکرد مدل یادگیری ماشین ما را کاهش ندهد.

یکی از قسمت های مهم در پیش پردازش داده ها با پایتون مدیریت داده های گم شده می باشد.برای انجام این کار باید داده های از دست رفته را با میانگین یا میانگین کل ستون جایگزین کنیم. برای این منظور ما از کتابخانهsklearn.impute استفاده خواهیم کرد که شامل یک کلاس به نام Imputerاست که به ما در پر کردن داده های از دست رفته کمک می کند.

گام سوم

بیشتر الگوریتم های یادگیری ماشین از فاصله اقلیدسی برای محاسات خود استفاده می کنند. به همین دلیل اگر چند نمونه مقدار خیلی زیاد یا خیلی کمی داشته باشند دقت مدلسازی کاهش می یابد. برای حل این مشکل از مقیاس بندی داده ها استفاده می شود. یکی از معروف ترین این مقایس بندی ها تبدیل Z است. تبدیل Z با استفاده از کلاس "StandardScaler" که در کتابخانه "sklearn.preprocessing است انجام می شود.

گام چهارم

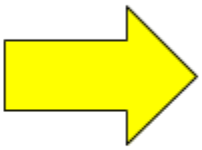
هر داده ای که عددی نباشد کیفی یا categorical است. برای مدلسازی حتما باید داده ها به صورت عددی باشند. برای مثال رنگ، رشته تحصیلی، وضعیت زندگی همگی categorical هستند.

برای این کار کلاس "LabelEncoder" را از کتابخانه "sklearn.preprocessing" وارد کرده و یک شی labelencoder_X از کلاس LabelEncoder ایجاد می کنیم. پس از آن ما از روش fit_transform برای تبدیل داده ها استفاده می کنیم.

پس از تبدلی مقادیر باید باز هم تغییراتی روی اعداد انجام دهیم. فرض کنید به رنگ قرمز عدد ۱ و به رنگ سبز عدد ۲ را نسبت دهیم. در این حالت الگوریتم های یادگیری ماشین عدد ۲ را برتر از عدد ۱ در نظر می گیرند. در صورتی که ما همچین نیتی نداشتیم. برای آن که از نظر الگوریتم اعداد برتری نسبت به هم نداشته باشند از **One-Hot Encoding**استفاده می کنیم.

One-Hot Encodingدر جایی که اعداد سلسله مراتبی نیستند کاربرد دارد. مثلا شماره تلفن یا کد پستی. این ها فقط اعدادی هستند که هیچ برتری بر دیگری ندارند. برای **One-Hot Encoding**از روش زیر استفاده می می کنیم.

Color
Red
Red
Yellow
Green
Yellow



Red	Yellow	Green
1	0	0
1	0	0
0	1	0
0	0	1

گام پنجم

در این مرحله از پیش پردازش با پایتون ما داده های خود را به دو مجموعه تقسیم می کنیم ، یکی برای آموزش مدل خود به نام مجموعه آموزشی و دیگری برای آزمایش عملکرد مدل خود. تقسیم به طور کلی 20/80 است. برای این کار ما "train_test_split" را از کتابخانه "sklearn.model_selection" را وارد می کنیم.

اکنون برای ساخت مجموعه های آموزشی و آزمایشی خود ، 4 مجموعه ایجاد خواهیم کرد:

- **X_train**:نمونه های قسمت آموزش
- **X_test**:نمونه های قسمت تست
- **Y_train**:برچسب های قسمت آموزش
- **Y_test**:برچسب های قست تست

تابع test_train_split نمونه ها و بر چسب ها به همراه نسبت آموزش به تست را در یافت می کند و آن را در ۴ متغییر می ریزد. ترتیب قرار داده متغییر ها باید به همین صورت باشد.

گام ششم

به سراغ ساخت مدل میرویم: یک الگوریتم درخت تصمیم (Decision Tree) و یک الگوریتم جنگل تصادفی (Random Forest) را برای مسائل دسته‌بندی استفاده می‌کنیم. این دو الگوریتم معمولاً برای مسائل دسته‌بندی و پیش‌بینی مورد استفاده قرار می‌گیرند. در این قسمت، مدل‌ها با داده‌های آموزشی آموزش داده شده‌اند، و سپس میزان دقت(accuracy) ، دقت مثبت (precision)، حساسیت (recall) و اسکور F1 (F1 score) بر روی داده‌های تست ارزیابی شده‌اند.

مقایسه ی میزان دقت هر مدل

مدل/دقت	Accuracy	Precision	Recall	F1 Score
Random Forest	0.9432624113475178	0.9446683508660677	0.9432624113475178	0.9436400477178193
Decision Tree	0.9361702127659575	0.936458271579944	0.9361702127659575	0.9361814257281418

منابع

- دیپ تیپ
- چت جی پی تی