



دانشگاه بوعلی سینا

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

نام استاد: دکتر منصوری زاده

موضوع: تحلیل داده کنگره ی آمریکا



فهرست مطالب

4	مقدمه
5	شرح تمرین
11	استخراج قوانین
14	مقایسه ی مدل ها
15	منابع

مقدمه

استفاده از درخت تصمیم به ما این امکان را می‌دهد که با بهبود فهم از توزیع داده‌ها و روابط میان ویژگی‌ها، تصمیمات بهتری بر اساس داده‌های موجود بگیریم. این روش قابل فهم، تفسیرپذیر، و قدرتمند است و از تصمیمات گام به گام در ارتباط با ویژگی‌ها و تأثیر آنها بر نتایج آگاهی می‌آورد. با استفاده از درخت تصمیم، ما می‌توانیم الگوهای پیچیده در داده‌ها را شناسایی کرده و به دست آوردن توصیفات قابل فهم از تصمیمات ما را تسهیل نماییم.

از طرفی استفاده از قوانین وابستگی در تحلیل داده‌ها و استخراج الگوها، یک راهکار قدرتمند برای بهبود فهم ما از روابط میان متغیرهاست. این قوانین اطلاعات مخفی و الگوهای پنهان را در داده‌ها آشکار می‌سازند و این اطلاعات می‌توانند در تصمیم‌گیری‌ها و بهبود فرآیندهای تصمیمی ما، می‌توانیم به سادگی قوانین مهم را از *Apriori* تأثیر گذار باشند. با بهره‌گیری از الگوریتم‌هایی مانند داده‌ها استخراج کرده و درک عمیق‌تری از رفتارها و الگوهای موجود در داده‌ها به دست آوریم. این ابزار قدرتمند به ما کمک می‌کند تا به سرعت و با دقت به سوالات مهم پیرامون رابطه میان متغیرها پاسخ دهیم و تصمیمات بهتری اتخاذ کنیم.

شایان توجه است که این تمرین، از فایل `house-votes-84.csv` برای دیتاست خود استفاده می‌کند.

شرح پروژه

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 معمولاً با کتابخانه pandas

پایتون کار می کنیم. این کتابخانه برای وارد کردن داده ها و مدیریت آن ها بسیار پرکاربرد است.

- گام دوم: بررسی 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 میانه کل ستون جایگزین کنیم. در اینجا چون تمام داده های ما کتگوریکال است ما از میانه استفاده می کنیم. برای این منظور کتابخانه `sklearn.impute` را استفاده خواهیم کرد که شامل یک کلاس به نام `Imputer` است که به ما در پر کردن داده های از دست رفته که با "?" نمایش داده شده اند، کمک می کند.

گام سوم

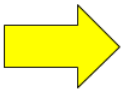
هر داده ای که عددی نباشد کیفی یا `categorical` است. برای مدلسازی حتما باید داده ها به صورت باینری باشند. برای مثال رنگ، رشته تحصیلی، وضعیت زندگی همگی `categorical` هستند.

پس از تبدیل 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 همچنین نیتی نداشتیم. برای آن که از نظر الگوریتم اعداد برتری نسبت به هم نداشته باشند از **One-Hot Encoding** استفاده می کنیم.

One-Hot Encoding در 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	Yellow	Green
Red	1	0	0
Red	1	0	0
Yellow	0	1	0
Green	0	0	1
Yellow	0	0	1



Red	Yellow	Green
True	False	False
True	False	False
False	True	False
False	False	True

گام پنجم (درخت)

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 برچسب ها به همراه نسبت آموزش به تست را در یافت می کند و آن را در ۴ متغیر می ریزد. ترتیب قرار داده متغیر ها باید به همین صورت باشد. لیبل نهایی داده ی ناشنخته ی ما قرار است یکی از انواع احزاب باشد پس ما این ستون را وارد تست نمیکنیم و در آموزش براساس حزب دموکرات تصمیم گیری ها را پیش می بریم.

گام پنجم (قوانین وابستگی)

پیاده سازی الگوریتم **Apriory** خود شامل چندین قدم است:

1. استفاده از **apriori** و **association_rules** از **mlxtend.frequent_patterns:**

- این دستورات از کتابخانه **mlxtend** برای اجرای الگوریتم **Apriori** و استخراج قوانین استفاده می کنند.

2. تبدیل دیتافریم به نوع داده: **bool**

- این دستور تبدیل داده های دیتافریم را به نوع داده **bool** انجام می دهد. احتمالاً این تبدیل برای اجرای الگوریتم **Apriori** و استفاده از متدلوژی ضروری است.

3. تعیین حداقل: **Support**

- **min_support** تعیین می کند که چه مقدار حداقل پشتیبانی (**support**) باید داشته باشند قوانین تا به عنوان فراوان ترین مجموعه ها شناخته شوند.

4. اجرای الگوریتم: **Apriori**

- `frequent_itemsets = apriori(df_encoded,`
`min_support=min_support, use_colnames=True`
این دستور الگوریتم

Apriori را روی دیتافریم اجرا می کند و مجموعه های فراوان تر (frequent itemsets) را بر اساس حداقل پشتیبانی استخراج می کند.

5. تعیین حداقل Confidence:

- **min_confidence** تعیین می کند که چه مقدار حداقل اطمینان (confidence) باید داشته باشند قوانین تا به عنوان قوانین قوی شناخته شوند.

6. استخراج قوانین انجمن با حداقل Confidence:

- `rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=min_confidence)` این دستور با استفاده از مجموعه های فراوان تر حاصل از Apriori، قوانین انجمن (association rules) را با حداقل اطمینان استخراج می کند.

7. نمایش قوانین:

- این بخش از کد، قوانین حاصل از استخراج را نمایش می دهد. اینجا یک شرط افزوده شده است تا تنها قوانینی که دارای 'Class Name_democrat' در **consequents** هستند نمایش داده شوند.
- `print(f"{antecedents} ----> {consequents}")` این دستور هر قانون را در قالب یک خط شامل مقدم و تالی چاپ می کند.

گام ششم (و آخر درخت)

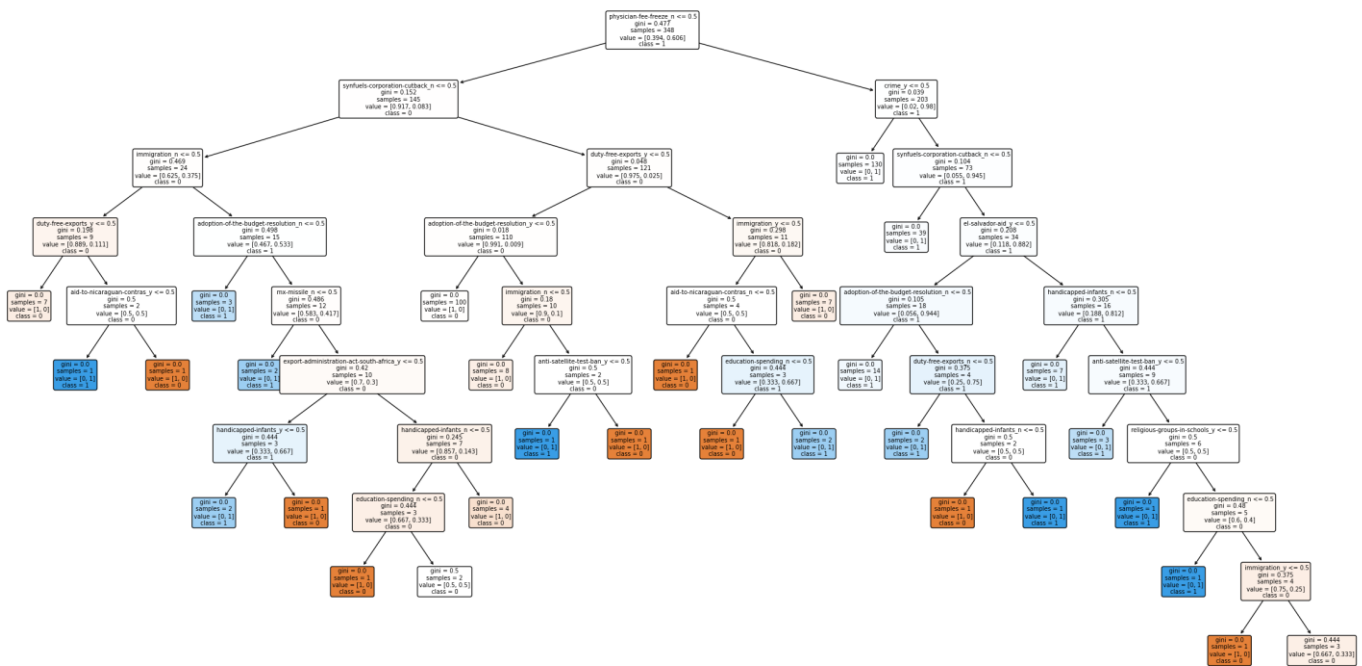
به سراغ ساخت مدل میرویم: یک الگوریتم درخت را برای مسائل دسته بندی استفاده می کنیم. این الگوریتم معمولاً برای مسائل دسته بندی و پیش بینی مورد استفاده قرار می گیرد و در داده های

ناشناخته به دنبال پیدا کردن لیبیل (دموکرات) است 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) و Support روی داده‌های تست ارزیابی شده‌اند.

Classification Report (Class Name_democrat):

- Accuracy on training set (Class Name_democrat): 0.9942528735632183
- Accuracy on test set (Class Name_democrat): 0.954022988505747

Democrat	precision	recall	f1-score	support
True	0.94	0.94	0.94	31
False	0.96	0.96	0.96	56



برخی قوانین استخراج شده از درخت:

Rule 1: If {'handicapped-infants_n': True, 'handicapped-infants_y': False, 'water-project-cost-sharing_n': False, 'water-project-cost-sharing_y': True, 'adoption-of-the-budget-resolution_n': True, 'adoption-of-the-budget-resolution_y': False, 'physician-fee-freeze_n': False, 'physician-fee-freeze_y': True, 'el-salvador-aid_n': False, 'el-salvador-aid_y': True, 'religious-groups-in-schools_n': False, 'religious-groups-in-schools_y': True, 'anti-satellite-test-ban_n': False, 'anti-satellite-test-ban_y': True, 'aid-to-nicaraguan-contras_n': True, 'aid-to-nicaraguan-contras_y': False, 'mx-missile_n': True, 'mx-missile_y': False, 'immigration_n': True, 'immigration_y': False, 'synfuels-corporation-cutback_n': False, 'synfuels-corporation-cutback_y': True, 'education-spending_n': False, 'education-spending_y': True, 'superfund-right-to-sue_n': False, 'superfund-right-to-sue_y': True, 'crime_n': False, 'crime_y': True, 'duty-free-exports_n': True, 'duty-free-exports_y': False, 'export-administration-act-south-africa_y': True} then Class Name_democrat

Rule 2: If {'handicapped-infants_n': True, 'handicapped-infants_y': False, 'water-project-cost-sharing_n': True, 'water-project-cost-sharing_y': False, 'adoption-of-the-budget-resolution_n': False, 'adoption-of-the-budget-resolution_y': True, 'physician-fee-freeze_n': False, 'physician-fee-freeze_y': True, 'el-salvador-aid_n': False, 'el-salvador-aid_y': True, 'religious-groups-in-schools_n': False, 'religious-groups-in-schools_y': True, 'anti-satellite-test-ban_n': False, 'anti-satellite-test-ban_y': True, 'aid-to-nicaraguan-contras_n': False, 'aid-to-nicaraguan-contras_y': True, 'mx-missile_n': True, 'mx-missile_y': False, 'immigration_n': False, 'immigration_y': True, 'synfuels-corporation-cutback_n': True, 'synfuels-corporation-cutback_y': False, '}

```
education-spending_n': True, ' education-spending_y': False, ' superfund-right-to-sue_n':
True, ' superfund-right-to-sue_y': False, ' crime_n': False, ' crime_y': True, ' duty-
free-exports_n': True, ' duty-free-exports_y': False, ' export-administration-act-south-
africa_n': False, ' export-administration-act-south-africa_y': True} then Class
Name_democrat
```

Rule 3: If {' handicapped-infants_n': True, ' handicapped-infants_y': False, ' water-
project-cost-sharing_n': True, ' water-project-cost-sharing_y': False, ' adoption-of-the-
budget-resolution_n': False, ' adoption-of-the-budget-resolution_y': True, ' physician-
fee-freeze_n': True, ' physician-fee-freeze_y': False, ' el-salvador-aid_n': True, ' el-
salvador-aid_y': False, ' religious-groups-in-schools_n': False, ' religious-groups-in-
schools_y': True, ' anti-satellite-test-ban_n': False, ' anti-satellite-test-ban_y': True,
' aid-to-nicaraguan-contras_n': False, ' aid-to-nicaraguan-contras_y': True, ' mx-
missile_n': False, ' mx-missile_y': True, ' immigration_n': False, ' immigration_y': True,
' synfuels-corporation-cutback_n': True, ' synfuels-corporation-cutback_y': False, '
education-spending_n': False, ' education-spending_y': True, ' superfund-right-to-sue_n':
True, ' superfund-right-to-sue_y': False, ' crime_n': False, ' crime_y': True, ' duty-
free-exports_n': False, ' duty-free-exports_y': True, ' export-administration-act-south-
africa_n': False, ' export-administration-act-south-africa_y': True} then Class
Name_democrat

Rule 4: If {' handicapped-infants_n': False, ' handicapped-infants_y': True, ' water-
project-cost-sharing_n': False, ' water-project-cost-sharing_y': True, ' adoption-of-the-
budget-resolution_n': False, ' adoption-of-the-budget-resolution_y': True, ' physician-
fee-freeze_n': True, ' physician-fee-freeze_y': False, ' el-salvador-aid_n': True, ' el-
salvador-aid_y': False, ' religious-groups-in-schools_n': True, ' religious-groups-in-
schools_y': False, ' anti-satellite-test-ban_n': False, ' anti-satellite-test-ban_y':
True, ' aid-to-nicaraguan-contras_n': False, ' aid-to-nicaraguan-contras_y': True, ' mx-
missile_n': False, ' mx-missile_y': True, ' immigration_n': True, ' immigration_y': False,
' synfuels-corporation-cutback_n': False, ' synfuels-corporation-cutback_y': True, '
education-spending_n': True, ' education-spending_y': False, ' superfund-right-to-sue_n':
True, ' superfund-right-to-sue_y': False, ' crime_n': True, ' crime_y': False, ' duty-
free-exports_n': False, ' duty-free-exports_y': True, ' export-administration-act-south-
africa_n': False, ' export-administration-act-south-africa_y': True} then Class
Name_democrat

برخی قوانین استخراج شده از طریق الگوریتم (S:0.5 , C:0.9)Apriori :

```
[' adoption-of-the-budget-resolution_y'] ----> ['Class Name_democrat']
```

```
[' physician-fee-freeze_n'] ----> ['Class Name_democrat']
```

```
[' adoption-of-the-budget-resolution_y', ' physician-fee-freeze_n'] ----> ['Class
Name_democrat']
```

```
[' aid-to-nicaraguan-contras_y', ' physician-fee-freeze_n'] ----> ['Class Name_democrat']
```

```
[' education-spending_n', ' physician-fee-freeze_n'] ----> ['Class Name_democrat']
```

مقایسه ی دو روش

با توجه به خروجی های درخت تصمیم و قوانین وابستگی، می توانیم مقایسه ای انجام دهیم:

1. درخت تصمیم:

دقت (Accuracy): دقت بسیار بالاست (تقریباً 99.4٪ در داده های آموزش و 95.4٪ در داده های تست).

ماتریس ماتریس سردرگمی (Confusion Matrix): بیانگر این است که تعداد زیادی از نمونه ها به درستی دسته بندی شده اند، با توجه به تعداد بسیار کم False Positive و False Negative.

Confusion Matrix (Class Name_democrat):

29	2
2	54

2. قوانین وابستگی (الگوریتم Apriori):

الگوریتم Apriori قوانینی با حداقل support 0.5 و confidence 0.9 استخراج کرده است. قوانین نشان دهنده ارتباط میان رای ها و حزب Democrat را نشان می دهند.

3. مقایسه:

• دقت:

درخت تصمیم دقت بالاتری دارد.

• پوشش:

الگوریتم Apriori تعداد کمتری از روابط را پوشش می دهد، اما ممکن است در مواردی با ساپورت و کانفیدنس بالا قوانین جم و جور تری نسبت به روش درخت به ما بدهد و پردازش را راحت تر کند.

• تفسیرپذیری:

درخت تصمیم قابل فهمتر و قابل تفسیرتر است، زیرا هر گره و شاخه از یک ویژگی معناپذیر تبعیه می کند. قوانین وابستگی ممکن است در موارد پیچیده تر، تفسیر آنها سخت تر باشد.

در نهایت، انتخاب بین این دو روش به اهداف خاص و نیازهای پروژه ما بستگی دارد. اگر دقت بیشتر و تفسیرپذیری مهم هستند، ممکن است درخت تصمیم را به کار بیاید. اگر می خواهیم الگوهای انجمنی و اتحادیه های نهاده ها را مطالعه کنید، الگوریتم Apriori مفیدتر باشد.

منابع

- دپ ٽپ
- فرادرس
- چت جي پي ٽي