HomeWork

AI

Numpy & Pandas

1. فایل تمرین را در پنل خود آپلود کنید.
2. title فایل تمرین به صورت (نام تمرین+نام و نام خانوادگی) به انگلیسی باشد.
3. از فرمت ipynb. برای اجرای بخش های مختلف در سلول های جداگانه استفاده نمایید.
4. در صورتی که سوال و یا ابهامی دارید در گروه چت تلگرامی بپرسید.

۱. مبانی NumPy – ساختار داده‌های پایه

هدف: هدف این تمرین، اطمینان از تسلط شما بر مفاهیم پایه‌ای ساخت، دستکاری و بررسی آرایه‌های NumPy است. این عملیات، اولین قدم در کار با هر دیتاست در یادگیری ماشین است.

ساخت دیتاست:

- یک آرایه یک‌بعدی NumPy به نام feature_vector بسازید که شامل ۱۲ عدد صحیح متوالی از عدد ۵ به بعد باشد.

- یک آرایه دوبعدی NumPy به نام feature_matrix با ابعاد (3, 4) بسازید که با اعداد صحیح شبه‌تصادفی (pseudo-random) بین ۰ و ۹ پر شده باشد. برای تکرارپذیری نتایج، از np.random.seed(42) استفاده کنید.
- مشخصات shape, ndim, size و dtype هر دو آرایه feature_vector و feature_matrix را چاپ کنید.

ایندکسینگ و برش‌دهی - دسترسی به داده‌ها:

- از feature_matrix، عنصر واقع در سطر دوم و ستون سوم را انتخاب و چاپ کنید.
- کل سطر اول feature_matrix را انتخاب و چاپ کنید.
- کل ستون آخر feature_matrix را انتخاب و چاپ کنید.
- یک زیرآرایه 2x2 از گوشه بالا سمت راست feature_matrix انتخاب کنید.

تغییر شکل داده:

- آرایه feature_vector را به یک آرایه دوبعدی با ۴ سطر و ۳ ستون تغییر شکل دهید (reshape) و shape جدید آن را چاپ کنید.
- آرایه feature_matrix را به یک آرایه یک‌بعدی تبدیل کنید (آن را flatten کنید) و نتیجه را چاپ کنید.

ترکیب دیتاست‌ها:

- یک ماتریس جدید 3x4 بسازید که تمام عناصر آن عدد ۱ باشد.
- این ماتریس جدید را به صورت عمودی (vertically stack) با feature_matrix اصلی خود ترکیب کنید.
- یک بردار ستونی 3x1 با مقادیر [10, 20, 30] بسازید.
- این بردار ستونی را به صورت افقی (horizontally stack) به feature_matrix خود اضافه کنید.

۲. عملیات عددی و پیش‌پردازش داده

هدف: این تمرین شما را با قدرت عملیات برداری، توابع تجمعی و Broadcasting آشنا می‌کند. این‌ها مفاهیم بنیادی برای پیش‌پردازش داده در یادگیری ماشین، مانند نرمال‌سازی هستند.

ساخت دیتاست:

- یک آرایه دوبعدی NumPy به نام scores با ابعاد (15, 3) بسازید که شامل اعداد صحیح شبه‌تصادفی بین ۵۰ و ۱۰۰ باشد. برای تکرارپذیری از seed شماره ۱۰۱ استفاده کنید.

تحلیل داده با توابع تجمعی:

- میانگین نمرات هر آزمون را محاسبه کنید (یعنی میانگین هر ستون).
- انحراف معیار (standard deviation) نمرات هر آزمون را محاسبه کنید.
- بالاترین نمره کسب شده در آزمون سوم را پیدا کنید.
- میانگین نمرات هر دانشجو را محاسبه کنید (یعنی میانگین هر سطر).

Broadcasting:

- فرض کنید که استاد تصمیم گرفته نمرات آزمون اول را با افزودن ۵ نمره به همه دانشجویان بهبود دهد. یک آرایه جدید به نام curved_scores بسازید که در آن فقط به ستون اول آرایه scores عدد ۵ اضافه شده است. از broadcasting برای این کار استفاده کنید

نرمال‌سازی داده:

یک مرحله بسیار رایج در پیش‌پردازش داده‌هایML ، نرمال‌سازی ویژگی‌ها است. این کار با کم کردن میانگین و تقسیم بر انحراف معیار برای هر ویژگی انجام می‌شود.

- میانگین و انحراف معیار هر آزمون را از آرایه اصلی scores محاسبه کنید.
- یک آرایه جدید به نام normalized_scores با استفاده از فرمول Z-score بسازید:

(scores - mean) / std

شما باید از broadcasting به درستی استفاده کنید تا آرایه یک‌بعدی mean و std را از ماتریس دوبعدی scores کم و بر آن تقسیم کنید.

- برای تأیید کار خود، میانگین و انحراف معیار هر ستون از normalized_scores را محاسبه کنید. میانگین باید بسیار نزدیک به ۰ و انحراف معیار بسیار نزدیک به ۱ باشد.

۳. فیلتر کردن داده و مهندسی ویژگی

هدف: این تمرین بر استفاده از ماسک‌های بولین (Boolean masking) برای فیلتر کردن داده و ساخت ویژگی‌های جدید تمرکز دارد.

در این تمرین نیز از همان دیتاست scores تمرین قبل استفاده خواهیم کرد.

شناسایی دانشجویان برتر:

- یک ماسک بولین بسازید تا تمام نمراتی که در آرایه scores بزرگتر از ۹۵ هستند را مشخص کند.
- با استفاده از این ماسک، تعداد نمرات بالای ۹۵ را بشمارید.
- یک آرایه جدید به نام high_performers_scores بسازید که فقط شامل نمرات بالاتر از ۹۵ باشد.

تحلیل شرطی:

- میانگین نمرات آزمون دوم را فقط برای دانشجویانی محاسبه کنید که در آزمون اول نمره‌ای برابر یا کمتر از ۷۰ گرفته‌اند.

مهندسی ویژگی - ساخت ویژگی قبول/مردود:

- یک آرایه یک‌بعدی جدید به نام pass_fail بسازید. یک دانشجو در صورتی قبول (True) محسوب می‌شود که میانگین نمراتش (در هر سه آزمون) بیشتر از ۶۵ باشد. در غیر این صورت مردود (False) است.

فیلترینگ پیچیده – دانشجویان ممتاز :

دانشجوی ممتاز کسی است که در هر سه آزمون نمره‌ای برابر یا بالاتر از ۹۰ کسب کرده باشد.

- یک ماسک بولین برای شناسایی این دانشجویان بسازید.
- از این ماسک برای نمایش نمرات تمام دانشجویان ممتاز استفاده کنید. اگر دانشجوی ممتازی وجود نداشت، پیامی مبنی بر این موضوع چاپ کنید.

۴. تحلیل اکتشافی داده یک دیتاست کلاسیک ML

هدف: این تمرین شما را با جریان کاری بنیادین هر پروژه یادگیری ماشین آشنا می‌کند: بارگذاری و بررسی اولیه دیتاست.

دیتاست :دیتاست تایتانیک. برای راحتی، از URL مستقیم زیر استفاده کنید:

```
https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv
```

بارگذاری داده:

- با استفاده از pd.read_csv() دیتاست تایتانیک را از URL بالا در یک DataFrame به نام df بارگذاری کنید.
- ۵ سطر اول DataFrame را با استفاده از متد head() نمایش دهید.

بررسی اولیه:

- از متد info() استفاده کنید. این یک گام حیاتی برای درک نوع داده‌های ستون‌ها و شناسایی مقادیر گمشده است.

- از متد describe() برای به دست آوردن آمار توصیفی (میانگین، انحراف معیار و...) برای ستون‌های عددی استفاده کنید.

انتخاب داده :

- در یادگیری ماشین، ما با ویژگی‌ها (ورودی‌ها) و هدف(خروجی قابل پیش‌بینی) کار می‌کنیم.
- ستون 'Survived' را انتخاب و نمایش دهید. این یک Pandas Series است. نوع داده آن را پرینت کنید.
- ستون‌های 'Pclass', 'Sex' و 'Age' را انتخاب و نمایش دهید.
- با استفاده از loc.، داده‌های مسافر با ایندکس ۳ را نمایش دهید.
- با استفاده از iloc.، داده‌های ۵ مسافر اول و ۳ ستون اول را نمایش دهید.

فیلتر کردن داده:

با استفاده از boolean masking به سؤالات زیر پاسخ دهید:

- چند مسافر زنده ماندند؟ (راهنمایی: می‌توانید روی یک Series بولین از sum() استفاده کنید)
- یک DataFrame جدید به نام survivors بسازید که فقط شامل داده‌های مسافران زنده‌مانده باشد.
- از میان بازماندگان، چند نفر مرد بودند؟
- یک DataFrame بسازید که فقط شامل مسافران درجه ۱ (Pclass == 1) با سن بالای ۵۰ سال باشد.

۵. پاک‌سازی و پیش‌پردازش داده

هدف: مدل‌های یادگیری ماشین نمی‌توانند با داده‌های گمشده کار کنند. این تمرین بر مهارت حیاتی پاک‌سازی داده و جایگزینی مقادیر گمشده تمرکز دارد.

شناسایی داده‌های گمشده:

- با استفاده از دستور isnull().sum()، تعداد مقادیر گمشده در هر ستون از DataFrame تایتانیک را بشمارید. کدام ستون‌ها بیشترین داده گمشده را دارند؟

مدیریت مقادیر گمشده - حذف کردن:

- ستون 'Cabin' بیش از حد مقدار گمشده دارد. با استفاده از متد drop()، این ستون را از df حذف کرده و نتیجه را در یک DataFrame جدید به نام df_cleaned ذخیره کنید.

مدیریت مقادیر گمشده - جایگزینی :

- ستون 'Age' دارای مقادیر گمشده است. یک استراتژی رایج، پر کردن این مقادیر با میانه (median) سن مسافران است. میانه سن را محاسبه کنید.
- با استفاده از fillna()، مقادیر گمشده ستون 'Age' در df_cleaned را با میانه سنی که محاسبه کردید، پر کنید.
- ستون 'Embarked' نیز چند مقدار گمشده دارد. برای داده‌های دسته‌بندی شده، استراتژی خوب، پر کردن با مُد (mode) است. مُد این ستون را پیدا کرده و با fillna(). مقادیر خالی را جایگزین کنید.
- با اجرای مجدد isnull().sum()، روی df_cleaned، مطمئن شوید که دیگر هیچ مقدار گمشده‌ای در ستون‌های 'Age' و 'Embarked' وجود ندارد.

تکمیل فرآیند:

- ستون 'Pclass' یک دسته را نشان می‌دهد اما به صورت عدد صحیح ذخیره شده است. با استفاده از متد astype()، نوع داده این ستون را به string یا category تغییر دهید.

۶. مهندسی ویژگی و تحلیل گروهی

هدف :کیفیت یک مدل یادگیری ماشین به شدت به کیفیت ویژگی‌های آن بستگی دارد. این تمرین به شما یاد می‌دهد که چگونه از ویژگی‌های موجود، ویژگی‌های جدید و آموزنده‌تری بسازید.

ساخت ویژگی جدید از ویژگی‌های موجود:

- در df_cleaned، یک ستون جدید به نام 'FamilySize' بسازید که حاصل جمع دو ستون 'SibSp' (تعداد خواهر/برادر/همسر) و 'Parch' (تعداد والدین/فرزندان) باشد.

استخراج اطلاعات برای ساخت ویژگی:

- ستون 'Name' شامل عناوینی مانند .Mr، .Mrs و .Miss است که می‌تواند یک ویژگی بسیار مهم باشد.
- یک ستون جدید به نام 'Title' بسازید و این عناوین را از ستون 'Name' استخراج کنید. (راهنمایی: می‌توانید از str.extract() با یک عبارت منظم استفاده کنید.
- مقادیر منحصربه‌فرد (unique) این ستون جدید را نمایش دهید.

تحلیل گروهی برای کسب بینش:

- متد groupby(). برای درک رابطه بین ویژگی‌های مختلف و متغیر هدف ('Survived') ضروری است.
- دیتافریم را بر اساس 'Sex' گروه‌بندی کرده و میانگین ستون 'Survived' را برای هر گروه محاسبه کنید تا نرخ بقای زنان و مردان را ببینید.
- دیتافریم را بر اساس 'Pclass' گروه‌بندی کرده و میانگین نرخ بقا را محاسبه کنید. کدام کلاس بیشترین شانس بقا را داشته است؟
- دیتافریم را بر اساس هر دو ستون 'Pclass'و 'Sex' گروه‌بندی کرده و میانگین نرخ بقا را محاسبه کنید.

دسته‌بندی داده‌های عددی:

- گاهی اوقات تبدیل یک ویژگی عددی پیوسته مانند 'Age' به یک ویژگی دسته‌بندی شده مفید است.
- یک ستون جدید به نام 'AgeGroup' بسازید. اگر سن مسافر زیر ۱۸ سال بود، مقدار آن 'Child'، اگر بین ۱۸ و ۶۵ بود 'Adult'، و اگر بالاتر از ۶۵ بود 'Senior' باشد. (راهنمایی: می‌توانید یک تابع بنویسید و از متد apply(). روی ستون 'Age'استفاده کنید)