

در ابتدا کتابخانه‌های مورد نظر را در برنامه خود وارد می‌کنیم.

فایل دیتاست را از گوگل دریافت کرده و در کنار فایل jupyter notebook خود قرار می‌دهیم و با دستور `pd.read_csv("./bank.csv", delimiter=";")` آن را می‌خوانیم (قرارگیری در RAM). پارامتر `delimiter` نشان دهنده نوع جداساز ما است که اغلب موارد کاما (,) است اما در اینجا (;) است.

قبل از انجام کاری، لازم است به سایت گفته رجوع کنیم و اطلاعات بیشتری راجب دیتاست کسب کنیم مانند تعداد سطرها/نمونه‌ها و تعداد ستون‌ها/ویژگی‌ها/بعدها و همچنین لازم است بدانیم ستون‌ها از چه نوع ساختاری هستند و مقادیر آنان چیست.

با استفاده از توابع داخلی (builtin function) کتابخانه Pandas می‌توانیم کارهای آماری روی دیتاست خود انجام دهیم نظیر تعداد نمونه‌ها، واریانس داده‌های عددی، کوچک‌ترین و بزرگترین مقدار داده‌ی عددی و ... تا بیشتر راجب داده‌های خود اطلاعات داشته باشیم.

یکی از موارد مهم در مرحله پیش‌پردازش داده‌ها (preprocessing)، حذف داده‌های null است چون الگوریتم یادگیری ماشین قادر به حل این مشکل نیست.

با دستور `df.isnull().sum` بررسی می‌کنیم که هر ستون چه مقدار داده‌ی null دارد که در این دیتاست، داده‌ی null نداریم.

بعد از این کار لازم است داده‌های طبقه‌بندی (categorical) را به داده‌های عددی (numeric) تبدیل کنیم. چون الگوریتم‌های یادگیری ماشین بر اساس داده‌های عددی یادگیری انجام می‌دهند. اما خوب است بدانیم لازم نیست داده‌های ستونی را که مقادیر آن True یا False است را به عددی تبدیل کنیم چرا که در پایتون `True=1` و `False=0` تعریف می‌شود.

یکی دیگر از روش‌های پیش‌پردازش داده‌های که باعث فهم بهتر الگوریتم می‌شود، `Z-score`، `min-max scaler` و ... است که در اینجا انجام داده نشده است.

بعد از این مراحل باید دیتاست خود را به بردار ویژگی `X` و بردار هدف `y` قسمت‌بندی کنیم و بعد از آن هم دو مجموعه آموزش (train) و آزمایش (test) درست کنیم تا با مجموعه آموزش مدل را آموزش دهیم و با مجموعه آزمایش، مدل را تست و ارزیابی کنیم.

بعد از این مراحل پیش‌پردازش وارد فاز جدید می‌شویم: آموزش مدل

یک نمونه از آن الگوریتمی که قرار است یاد بگیرد، ایجاد کرده و در صورت لزوم ابرپارامترها (hyperparameters) را به آن می‌دهیم. و بعد از آن با دادن مجموعه آموزش به الگوریتم، مدل آن را یاد می‌گیرد.

در اینجا از درخت تصمیم (Decision Tree) استفاده می‌کنیم که زیر مجموعه‌ای از مسائل طبقه‌بندی (classification) است.

بعد از یادگیری مدل، باید مدل را سنجید به همین دلیل ما مجموعه بردارهای تست X را به مدل می‌دهیم تا به ما بردار هدف y را بدهد که ما آن را y_pred می‌نامیم.

اکنون باید مقایسه‌ای بین y_pred با مقادیر واقعی بردار هدف y_true انجام دهیم و عملکرد مدل را معیارهای به سنجیم. چندین معیار ارزیابی عملکرد مدل‌های طبقه‌بندی وجود دارد به نام‌های $accuracy$ و $recall$ و $precision$ و $f1\text{-score}$ و ... که پشت تمام این‌ها روابط ریاضی است که تا حد خوبی در کلاس بررسی و یادگرفته شده است و مهم‌تر از روابط لازم است درک کنیم که هر کدام چه چیزی را بیان می‌کند.

در کتابخانه Sklearn، جمعی از این معیارهای ارزیابی وجود دارد: `classification_report` که با دادن y_true و y_pred به عنوان آرگومان به تابع، برایمان معیارها را بدست می‌آورد.

هنگامی داریم نمونه‌ای از الگوریتم Decision Tree را ایجاد می‌کنیم یکی از ابرپارامترهای آن `max_depth` است که مشخص می‌کند عمق درخت تصمیم ما چقدر باشد و یکی از روش‌های جلوگیری از بیش‌برازش (overfit) است.

پیاده سازی الگوریتم ID3:

الگوریتم ID3 وظیفه پیدا کردن ویژگی‌هایی دارای اطلاعات زیاده‌تر (Gain بیشتر) را دارد و آن‌ها را در سطوح بالاتری از درخت قرار می‌دهد. هر بار که یک ویژگی در سطحی از درخت انتخاب شد، زیر درخت‌های آن نیز دقیقاً به همان صورت (ویژگی‌هایی با اطلاعات بالا) انتخاب می‌شوند و در سطوح و گره‌های بعدی قرار می‌گیرند. البته وقتی یک گره از درخت انتخاب شد، برای ساخت زیر درخت‌های دیگر، مجموعه داده‌ها بر اساس مقدار گرهی انتخاب شده در شاخه‌های بالاتر، کوچکتر می‌شوند و هر چه در درخت پایین‌تر می‌رویم (به برگ‌ها نزدیک‌تر می‌شویم)، مجموعه داده‌ها برای محاسبه‌ی مقدار اطلاعات کمتر می‌شوند.

ساخت درخت تصمیم با استفاده از الگوریتم ID3 بر اساس معیار اطلاعات (Information Gain) و انتروپی (Entropy) استفاده می‌شود. Entropy در واقع نشان دهنده‌ی کم بودن اطلاعات است. یعنی در مجموعه‌ی داده‌ی شما، بر اساس یک ویژگی (یک بُعد) چقدر می‌توانید تشخیص دهید که کلاس نهایی چیست. Gain که در واقع همان Information Gain است، از Entropy در هر مقدار از ویژگی‌ها کمک گرفته و به میزان اطلاعاتی که می‌توان از یک ویژگی (بُعد) به دست آورد، گفته می‌شود. یعنی یک ویژگی خاص چقدر می‌تواند اطلاعات بیشتری به ما بدهد.

کد نوشته شده یک تابع برای محاسبه انتروپی، یک تابع برای محاسبه اطلاعات، و یک تابع برای ساخت درخت تصمیم با استفاده از الگوریتم ID3 دارد.

با استفاده از دیتاست فراهم شده و توابع تعریف شده، یک درخت تصمیم را بر اساس معیار اطلاعات و انتروپی می‌سازد و آن را به صورت گرافیکی نمایش می‌دهد. این درخت تصمیم نشان می‌دهد که چگونه می‌توان با پرسیدن سوالات یا شرایط مختلف از ویژگی‌های متقاضیان، تصمیم‌گیری در مورد اعطای وام به آن‌ها را انجام داد.