باسمه تعالی

تمرین درس هوش مصنوعی: نوآوری و جامعه

شرح تمرین

XonLearn یک استارت‌آپ مستقر در مرکز کارآفرینی شریف است که دوره‌های آموزشی آنلاین در زمینه فناوری‌های پیشرفته به دانش‌جویان و متخصصان ارائه می‌کند تا به ارتقای دانش و مهارت آن‌ها کمک کند. یکی از دوستان شما در دانشکده به مدیران این مجموعه پیشنهاد کرده است تا جلسات ابتدایی دوره‌های خود را به صورت رایگان در اختیار برخی از مشتریان بالقوه بگذارد تا آن‌ها ترغیب به خرید کل جلسات دوره شوند.

با این حال، مدیران مجموعه مایل نیستند تا محتوای جلسات را برای عموم علاقه‌مندان به صورت رایگان در دسترس قرار دهند. آن‌ها سراغ شما آمده‌اند و برخی از اطلاعات مشتریان خود را در اختیار شما قرار داده‌اند تا با استفاده از ابزارهای هوش مصنوعی به آن‌ها در شناسایی آن دسته از مشتریانی که به احتمال بیشتر به این اقدام، پاسخ قطعی مثبت می‌دهند، کمک کنید. در این صورت، آن‌ها می‌توانند پیشنهاد دسترسی رایگان به جلسات ابتدایی را تنها در اختیار گروه محدودی از مشتریان قرار دهند که به احتمال بیشتری نیز کل دوره را خریداری خواهند کرد.

داده‌های مشتریان در یک فایل اکسل به نام XonLearn به پیوست تمرین قرار دارد.

صورت سوالات تمرین نیز در یک فایل با فرمت ipynb با نام HW3_AI Innovation and Society ضمیمه شده است. پیشنهاد می‌کنیم برای باز کردن این نوت‌بوک از ژوپیتر نوت‌بوک (Jupyter Notebook) استفاده کنید. یکی از راه‌های ساده‌ی دسترسی به ژوپیتر نوت‌بوک و بهره‌مندی از کتابخانه‌های زبان پایتون استفاده از Anaconda است. لطفا به منظور نصب به لینک زیر مراجعه کنید.

https://www.codecademy.com/article/setting-up-jupyter-notebooks

دقت کنید که کدها و جواب سوالات را باید در نواحی تعین شده در نوت‌بوک ضمیمه وارد کنید و سپس در CW بارگذاری کنید. تکالیفی که در فرمت‌های دیگر بارگذاری شوند، تصحیح نخواهند شد.

برای انجام تمرین، ابتدا قسمت **توضیحات مقدماتی** را در پایان این فایل مطالعه کنید. سپس کارگاه زیر را مشاهده کنید:

https://drive.google.com/file/d/1C2Bg_ni1qHM5yt-mSaOr7OiFWvGDJMEQ/view?usp=sharing

برای دسترسی به نوت‌بوک تهیه شده در کارگاه می‌توانید به لینک زیر مراجعه کنید:

https://drive.google.com/file/d/1hDS2Dy8O3Z7O3F6VMcSxiv3cYDusPB2l/view?usp=sharing

به منظور آشنایی با مقدمات کتابخانه‌های Numpy، Pandas و Matplotlib می‌توانید به لینک زیر مراجعه کنید:

https://www.kaggle.com/code/chats351/introduction-to-numpy-pandas-and-matplotlib

هم‌چنین در صورت علاقه به یادگیری بیشتر می‌توانید در دوره‌ی آموزشی یادگیری ماشین استنفورد Andrew NG نیز شرکت کنید (برای این تمرین نیازی نیست.):

https://www.youtube.com/watch?v=J8Eh7RqggsU&list=PLoROMvodv4rO1NB9TD4iUZ3qghGEGtqNX

اهداف تمرین

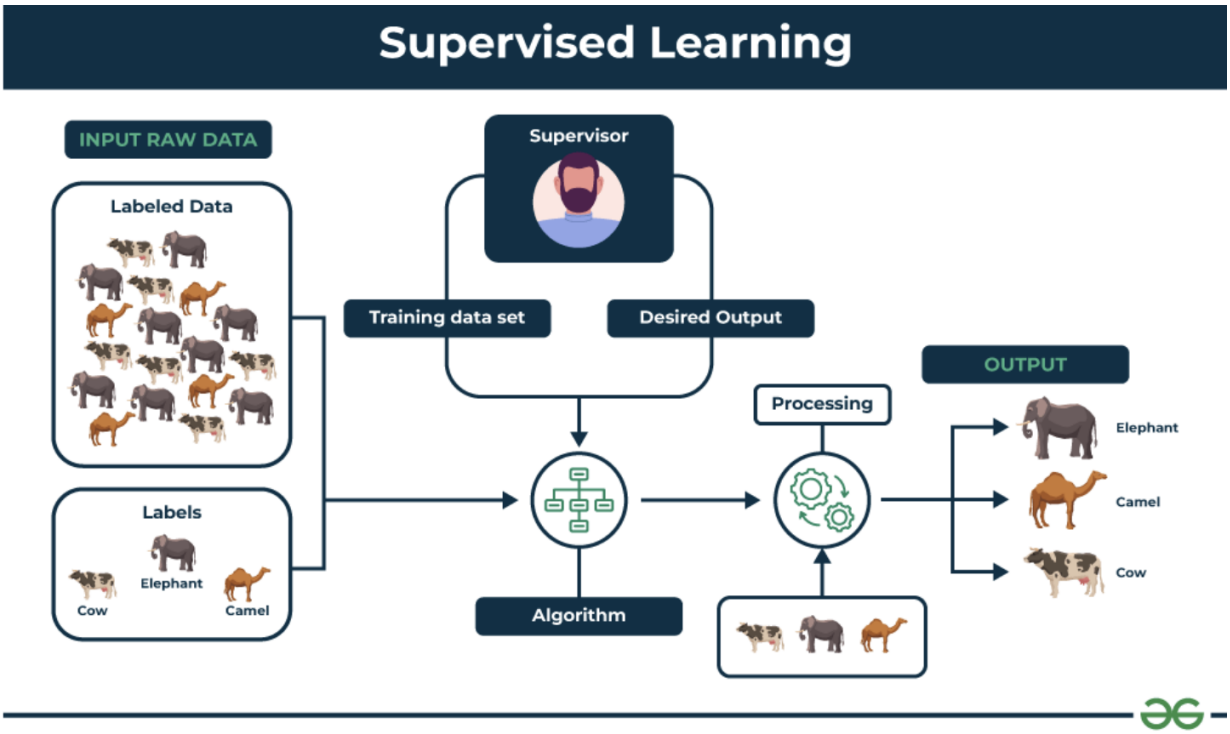
- آشنایی با فضای تعاملی ژوپیتر نوت‌بوک (یا Google Colab)
- آشنایی مقدماتی با زبان برنامه‌نویسی پایتون برای آنالیز داده و برنامه‌نویسی الگوریتم‌های یادگیری ماشین
- آشنایی با دو الگوریتم پرکابرد یادگیری ماشین (درخت تصمیم و جنگل تصادفی) و نقاط قوت و ضعف آن‌ها
- آشنایی با مراحل متداول آموزش الگوریتم‌های یادگیری ماشین و ایرادات احتمالی و چگونگی اصلاح آن‌ها
- دستیابی به درک عملی از برخی کاربردهای هوش مصنوعی در کسب و کارها

توضیحات مقدماتی

الگوریتم‌های یادگیری ماشین متنوع هستند و عموما بر اساس نحوه‌ی آموزش به دو دسته اصلی تقسیم می‌شوند:

۱. یادگیری تحت نظارت (Supervised Learning)
۲. یادگیری بدون نظارت (Unsupervised Learning)

در یادگیری تحت نظارت، الگوریتم با داده‌های برچسب‌دار آموزش می‌بیند و یاد می‌گیرد رابطه‌ی میان داده‌ها (تصاویر حیوانات) و برچسب آن‌ها (نام حیوانات) را تشخیص دهد. سپس این الگوریتم‌ها می‌تواند برچسب داده‌های بدون برچسب-گذاری را پیش‌بینی کند. برای درک بهتر به تصویر زیر نگاه کنید.



الگوریتم‌های دسته‌ی یادگیری تحت نظارت به دو زیرشاخه‌ی اصلی تقسیم می‌شوند:

۱. رگرسیون (Regression)
۲. طبقه‌بندی (Classification)

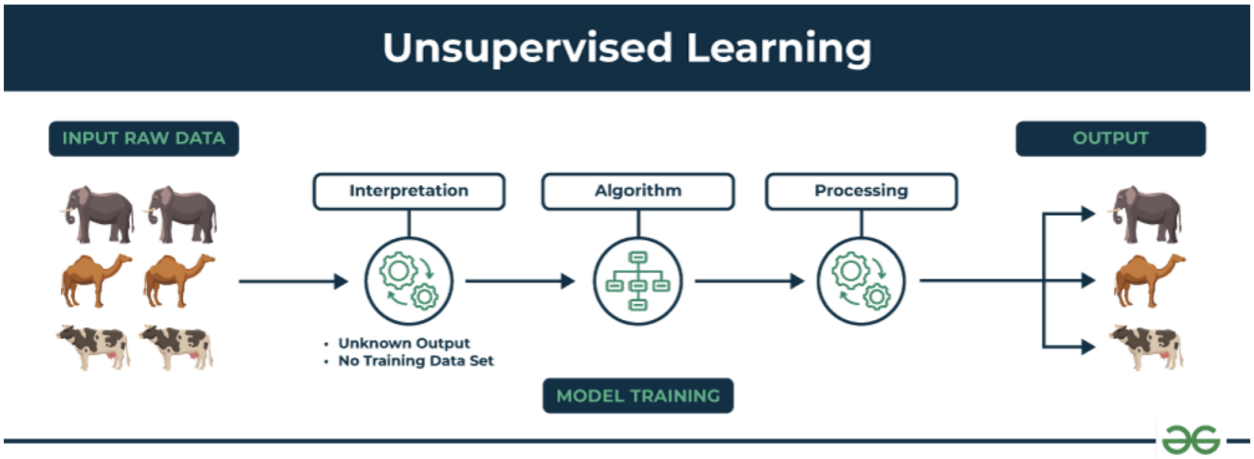
در زیرشاخه‌ی رگرسیون، هدف پیش‌بینی متغیرهای کمی پیوسته است؛ مانند قیمت خانه. برخی از الگوریتم‌های پرکاربرد در زیرشاخه‌ی رگرسیون عبارتند از:

- Linear Regression
- Polynomial Regression
- Support Vector Machine Regression
- Decision Tree Regression
- Random Forest Regression

در زیرشاخه‌ی طبقه‌بندی، هدف پیش‌بینی متغیرهای گسسته است؛ مانند رنگ. برخی از الگوریتم‌های پرکاربرد در زیرشاخه‌ی طبقه‌بندی عبارتند از:

- Logistic Regression
- Support Vector Machines
- Decision Trees
- Random Forests
- Naive Baye

از سمت دیگر، یادگیری بدون نظارت شامل آموزش الگوریتم با داده‌هایی است که برچسب‌گذاری نشده‌اند. هدف این یادگیری تشخیص الگوها و روابط میان داده‌ها بدون استفاده از دستورالعملی مشخص است. به تصویر نگاه کنید.



الگوریتم‌های دسته‌ی یادگیری بدون نظارت نیز به دو زیرشاخه‌ی اصلی تقسیم می‌شوند:

۱. خوشه‌بندی (Clustering)
۲. کاهش ابعاد (Dimension Reduction)

در زیرشاخه‌ی خوشه‌بندی، هدف گروه‌بندی داده‌های مشابه با یکدیگر است. انواع روش‌های خوشه‌بندی عبارتند از:

- Hierarchical clustering
- K-means clustering
- Principal Component Analysis
- Singular Value Decomposition
- Independent Component Analysis
- Gaussian Mixture Models (GMMs)
- Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

در زیرشاخه‌ی کاهش ابعاد، هدف تشخیص الگوی داده‌هاست. برخی از الگوریتم‌های پرکاربرد این روش عبارتند از:

- Apriori Algorithm
- Eclat Algorithm
- FP-Growth Algorithm

در این تمرین از شما انتظار داریم تا به کمک دو الگوریتم درخت تصمیم و جنگل تصادفی که از الگوریتم‌های پرکاربرد یادگیری تحت نظارت در هر دو زیرشاخه‌ی رگرسیون و طبقه‌بندی هستند، داده‌های مشتریان را طبقه‌بندی کنید. از این رو در ادامه به شرح بیشتر این دو الگوریتم و منطق پشت آن‌ها می‌پردازیم.

درخت تصمیم

درخت تصمیم (Decision Tree) یک الگوریتم یادگیری ماشین قدرتمند و در عین حال بصری است که همان طور که پیش‌تر توضیح داده شد، در زیرشاخه‌های رگرسیون و طبقه‌بندی مورد استفاده قرار می‌گیرد. محبوبیت این الگوریتم ناشی از سادگی، تفسیرپذیری و تطبیق‌پذیری آن است.

منطق الگوریتم درخت تصمیم، مدل‌سازی فرآیند تصمیم‌گیری با پرسیدن سؤالات متوالی است؛ به طوری که داده‌ها بر اساس ویژگی‌هایی که برای پیش‌بینی آموزنده‌تر هستند، به چندین زیرمجموعه تقسیم شوند. برای درک بهتر این روش، به بازی ۲۰ سوالی فکر کنید.

درخت تصمیم از گره‌ها، شاخه‌ها و برگ‌ها تشکیل شده است. بالاترین گره، گره ریشه (Root Node) نامیده می‌شود که کل مجموعه داده را نشان می‌دهد. هر گره در درخت نشان‌دهنده‌ی یک سوال است و شاخه‌های زیر یک گره پاسخ‌های احتمالی به آن سوال را نشان می‌دهد که منجر به سوالات بعدی (گره‌های داخلی) یا تصمیم نهایی (گره‌های برگ) می‌شوند. هنگامی که یک گره‌ی داخلی به گره‌های دیگری تقسیم می‌شود، به آن گره‌ی تصمیم (Decision Node) می‌گویند. گره‌هایی که بیشتر از هم جدا نمی‌شوند و به شاخه‌های جدیدی نمی‌انجامند، گره برگ نام دارند که خروجی نهایی الگوریتم هستند. به فرآیند تقسیم یک گره به دو یا چند زیرگره جداسازی (Splitting) گفته می‌شود. گاهی برای کاهش پیچیدگی مدل و یا جلوگیری از بیش‌برازش (Overfitting) آن لازم است برخی از گره‌های داخلی یک گره‌ی تصمیم حذف شود. به این فرآیند هرس (Pruning) می‌گویند.

مراحل آموزش و استفاده از یک درخت تصمیم عبارتند از:

۱. **شروع از ریشه**: با مجموعه داده در گره‌ی ریشه شروع کنید.

۲. **انتخاب ویژگی**: در هر گره، یک ویژگی را انتخاب کنید که به بهترین شکل داده‌ها را به چند دسته تقسیم می‌کند. این انتخاب بر اساس معیارهایی مانند ضریب ناخالصی جینی (Gini Impurity)، آنتروپی و یا کاهش واریانس (برای رگرسیون) انجام می‌شود که در ادامه بیشتر توضیح می‌دهیم.

۳. **جداسازی**: برای هر مقدار ممکن از ویژگی انتخاب شده، شاخه‌هایی ایجاد کنید و مجموعه داده را بر اساس آن تقسیم کنید.

۴. **تکرار مراحل ۲ و ۳ برای گره‌های داخلی**: مراحل ۲ و ۳ را برای هر گره جدید ایجاد شده تکرار کنید تا زمانی که یکی از شرایط توقف برآورده شود. این شرایط عبارتند از: ۱. تمام نقاط داده‌ی حاضر در یک گره، متعلق به یک کلاس هستند و هیچ ویژگی‌ای باقی نمانده است که به تقسیم‌بندی آن کلاس به زیرکلاس‌های جدید بیانجامد. ۲. درخت به یک عمق از پیش تعریف شده رسیده است.

۵. **پیش‌بینی**: برای پیش‌بینی یک نمونه جدید، از ریشه‌ی درخت ساخته شده شروع کنید و با توجه به ویژگی‌های نمونه از میان گره‌ها حرکت کنید تا به یک گره برگ برسید.

همان طور که گفته شد، دو معیار رایج برای انتخاب ویژگی وجود دارد: ۱. ناخالصی جینی و ۲. آنتروپی.

- معیار ناخالصی جینی فرکانس برچسب‌گذاری نادرست هر عنصر مجموعه داده را در صورتی که به طور تصادفی برچسب‌گذاری شده باشد، اندازه‌گیری می‌کند. برای مجموعه‌ای با کلاس‌های J،

$$Gini = 1 - \sum_{j=1}^{J} p_j^2$$

که در آن p_j نسبت نمونه‌های کلاس j در بین نمونه‌های آموزشی در مجموعه داده است.

- آنتروپی نیز سطح عدم قطعیت (یا ناخالصی) را در گروهی از نقاط داده اندازه‌گیری می‌کند. آنتروپی زمانی که یک مجموعه حاوی ترکیبی از کلاس‌ها باشد، بیشتر و زمانی که مجموعه‌ای خالص است، کمتر است. آنتروپی برای یک مجموعه به صورت زیر تعریف می‌شود:

$$Entropy = -\sum_{j=1}^{J} p_j \log_2 p_j$$

به تفاوت آنتروپی قبل و بعد از تقسیم روی یک ویژگی information gain گفته می‌شود.

درختان تصمیم به دلیل سادگی و قابل تفسیر بودن کاربردهای فراوانی در دنیای روزمره دارند. آن‌ها تصمیم‌گیری انسانی را بیشتر از سایر الگوریتم‌ها تقلید می‌کنند و منطق آنها شفاف و قابل درک است. با این حال، آن‌ها می‌توانند مستعد بیش‌برازش باشند؛ به ویژه درختان عمیق. روش‌هایی مانند هرس کردن، تعیین حداکثر عمق و روش‌های مجموعه‌ای (مانند جنگل‌های تصادفی) می‌توانند به کاهش بیش‌برازش کمک کنند.

جنگل تصادفی

جنگل تصادفی (Random Forest) یک روش یادگیری مجموعه‌ای (Ensemble Learning) است که با ساختن تعداد زیادی درخت تصمیم در زمان آموزش و اعلام کلاسی که درختان بیشترین پیش‌بینی را برای آن دارند، عمل می‌کند.

منطق پشت جنگل‌های تصادفی این است که در حالی که یک درخت تصمیم ممکن است مستعد بیش‌برازش باشد و یا به نویز در داده‌های آموزشی بسیار حساس باشد، ترکیب پیش‌بینی‌های چندین درخت واریانس را کاهش می‌دهد و به یک مدل قوی‌تر و دقیق‌تر منجر می‌شود. در حقیقت، این روش از خرد جمعی استفاده می‌کند که در آن خطاهای پیش‌بینی تک تک درختان می‌توانند ضمن میانگین‌گیری از بین بروند و منجر به تعمیم بهتر داده‌های دیده نشده شوند.

یک جنگل تصادفی معمولاً با تجمیع دو تکنیک پرکاربرد ساخته می‌شود: ۱. بسته‌بندی بوت استرپ (Bootstrap Aggregating) یا به اختصار Bagging و ۲. تصادفی بودن ویژگی‌ها.

- **بوت استرپ**: در این تکنیک، برای هر درخت یک نمونه تصادفی از داده‌ها با جایگزینی انتخاب می‌شود تا زیرمجموعه‌های مختلفی از داده‌ها برای آموزش درختان مختلف ایجاد شود.
- **تصادفی بودن ویژگی**: هنگام تقسیم گره‌ها در طول ساخت یک درخت، به جای همه ویژگی‌ها، یک زیرمجموعه تصادفی از ویژگی‌ها برای تقسیم در گره‌ها درنظرگرفته می‌شود. این تکنیک تضمین می‌کند که درختان متنوع هستند و همبستگی بین آنها را کاهش می‌دهد.

مراحل آموزش و استفاده از یک درخت تصمیم عبارتند از:

۱. **ایجاد یک جنگل**: با ایجاد چندین درخت تصمیم شروع کنید. تعداد درختان (n) یک فراپارامتر (Hyperparameter) است. فراپارامترها تنظیماتی هستند که به کمک آن‌ها می‌توان ساختار جنگل را تغیر داد.

۲. **نمونه برداری تصادفی**: برای هر درخت، به طور تصادفی از مجموعه‌ی داده با جایگزینی نمونه‌برداری کنید تا یک مجموعه داده‌ی بوت استرپ ایجاد کنید. این نمونه همان چیزی است که درخت روی آن آموزش داده خواهد شد.

۳. **تقسیم گره‌ها با تصادفی بودن ویژگی**: در طول ساخت هر درخت، در هر گره، به طور تصادفی زیرمجموعه‌ای از ویژگی‌ها را انتخاب کنید و بهترین تقسیم را از این زیرمجموعه (به جای کل مجموعه ویژگی) تعیین کنید.

۴. **ساختن درختان**: هر درخت را بدون هرس به حداکثر میزان رشد دهید.

۵. **پیش‌بینی**: برای پیش‌بینی کلاس یک نمونه‌ی جدید، اجازه دهید هر درخت در جنگل پیش‌بینی کند و از رای اکثریت (برای طبقه‌بندی) یا میانگین (برای رگرسیون) به عنوان پیش‌بینی نهایی استفاده کنید.

جنگل‌های تصادفی با ایجاد مجموعه‌ای که می‌تواند الگوهای پیچیده در داده‌ها را بدون قرار گرفتن در دام بیش برازش ثبت کند، درخت‌های تصمیم را بهبود می‌بخشد. با ادغام پیش‌بینی‌ها در چندین درخت، جنگل‌های تصادفی اغلب به دقت بالایی در بسیاری از وظایف دست می‌یابند و در برابر نویز مقاوم هستند. جنگل‌های تصادفی به طور گسترده‌ای برای طبقه‌بندی و رگرسیون مورد استفاده قرار می‌گیرند. علیرغم نقاط قوت آن‌ها، جنگل‌های تصادفی به دلیل ماهیت مجموعه‌ای که دارند، می‌توانند از نظر محاسباتی سنگین باشند و طبیعتا از تفسیرپذیری کمتری برخوردارند.