به نام کیمیاگر عالم

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

یادگیری ماشین کاربردی

عنوان

تمرین دوم (HW02)

مدرس

دکتر احسان ناظر فرد

دانشجو

امیرحسین بابائیان

۴۰۱۱۳۱۰۰۲

ترم بهار ۰۲-۰۱

دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)

فهرست

فهرست تصاویر

سوال اول

بخش a

مجموعه داده‌های نامتوازن چالش‌هایی برای الگوریتم‌های یادگیری ماشین ایجاد می‌کنند. این چالش‌ها شامل عدم توازن بین تعداد نمونه‌های مثبت و منفی در داده‌های طبقه‌بندی، تفاوت بین مقادیر حاوی داده‌های آموزش و تست و نویز داده‌ها می‌شوند. عدم توازن در تعداد نمونه‌های مثبت و منفی می‌تواند باعث شود که الگوریتم‌های طبقه‌بندی به طور ناخواسته به دنبال دسته‌ی بیشتری از داده‌ها بگردند و عملکرد آن‌ها را بدرستی تحلیل کنند.

برای حل این چالش‌ها، تکنیک‌های پیش‌پردازش مانند Oversampling و Undersampling می‌توانند به کار گرفته شوند . Oversamplingبه معنی تکثیر نمونه‌های داده‌ی کمتر است و Undersampling به معنی حذف نمونه‌های اضافی است. این تکنیک‌ها می‌توانند تعادل بین تعداد نمونه‌های مثبت و منفی را در مجموعه داده برقرار کنند. تکنیک‌های دیگری مانندSMOTE ، ترکیب Oversampling و Undersampling و دسته‌بندی نمونه‌های داده‌ای با استفاده از خوشه‌بندی وجود دارد که می‌تواند در مقابله با مسئله‌ی نامتوازنی مجموعه داده کمک کنند.

بخش b

نرمال‌سازی داده‌ها به طرز قابل توجهی بر عملکرد الگوریتم‌های یادگیری ماشین تأثیر می‌گذارد. در واقع، با نرمال‌سازی داده‌ها، مقیاس آن‌ها را به یک رنج مشخص و یا صفر و یک تبدیل می‌کنیم و باعث می‌شویم که الگوریتم‌های یادگیری ماشین به درستی بر روی داده‌ها عمل کنند.

چندین تکنیک معمول برای نرمال سازی داده‌ها وجود دارد، از جمله Min-Max scaling و .Z-score normalization در Min-Max scaling، مقادیر داده‌ها را در یک بازه مشخص (معمولاً بین ۰ تا ۱) قرار می‌دهیم. درZ-score normalization ، مقادیر داده‌ها را به میانگین داده‌ها تقسیم کرده و به انحراف معیار تقسیم می‌شوند.

انتخاب تکنیک مناسب برای نرمال سازی داده‌ها، تأثیر مستقیمی بر عملکرد الگوریتم‌های یادگیری ماشین دارد. به همین دلیل، باید ملاحظاتی را در نظر گرفت. برخی از این ملاحظات شامل اندازه مجموعه داده، توزیع داده، نوع الگوریتم یادگیری ماشین و هدف مورد نظر از آن‌ها است. به طور کلی، نرمال سازی داده‌ها می‌تواند بهبود قابل توجهی در عملکرد الگوریتم‌های یادگیری ماشین داشته باشد.

بخش c

تبدیل متغیرهای طبقه‌ای به ویژگی‌های عددی با استفاده از تکنیک‌های پیش‌پردازش، از جمله روش‌های مختلف رمزگذاری، امکان‌پذیر است. با این کار، می‌توانیم متغیرهای طبقه‌ای را به فضای عددی تبدیل کنیم و با استفاده از آن‌ها در الگوریتم‌های یادگیری ماشین استفاده کنیم.

روش‌های مختلف رمزگذاری شامل روشOne-hot encoding ، Label encodingو Binary encoding است. هر کدام از این روش‌ها ابعاد مختلفی از داده‌ها را به دنبال دارند که ممکن است تأثیر مستقیمی بر دقت الگوریتم‌های یادگیری ماشین داشته باشد.

انتخاب روش مناسب رمزگذاری بر عملکرد الگوریتم‌های یادگیری ماشین، پیچیدگی محاسباتی و تفسیرپذیری الگوریتم تأثیرگذار است. برخی از ملاحظاتی که باید در انتخاب روش مناسب برای رمزگذاری در نظر گرفت عبارت‌اند از: نوع داده‌های ورودی، توزیع داده‌ها، ابعاد داده‌ها، و توانایی تفسیر پذیری الگوریتم و...

در کل، استفاده از تکنیک‌های پیش‌پردازش جهت تبدیل متغیرهای طبقه‌ای به ویژگی‌های عددی، می‌تواند در بهبود دقت الگوریتم‌های یادگیری ماشین مؤثر باشد.

بخش d

در حوزه یادگیری ماشین، مدل‌های پارامتریک و غیرپارامتریک به دو دسته اصلی تقسیم می‌شوند. در مدل‌های پارامتریک، تعداد پارامترهای ثابت و محدودی برای مدل وجود دارد که باید توسط داده‌های آموزشی تعیین شود. در حالی که در مدل‌های غیرپارامتریک، تعداد پارامترها محدود نیست و تعداد بیشتری از پارامترها می‌توانند توسط داده‌های آموزشی تعیین شوند.

مدل درخت تصمیم یک مدل غیرپارامتریک است. این به این معناست که تعداد پارامترهای آن محدود نیست و توسط داده‌های آموزشی تعیین نمی‌شوند. درخت تصمیم با استفاده از یک سری از سوالات بله/خیر، داده‌ها را به دو دسته جدا می‌کند و هیچ پارامتر ثابتی برای تعیین این سوالات وجود ندارد. به همین دلیل، درخت تصمیم به عنوان یک مدل غیرپارامتریک شناخته می‌شود.

بخش e

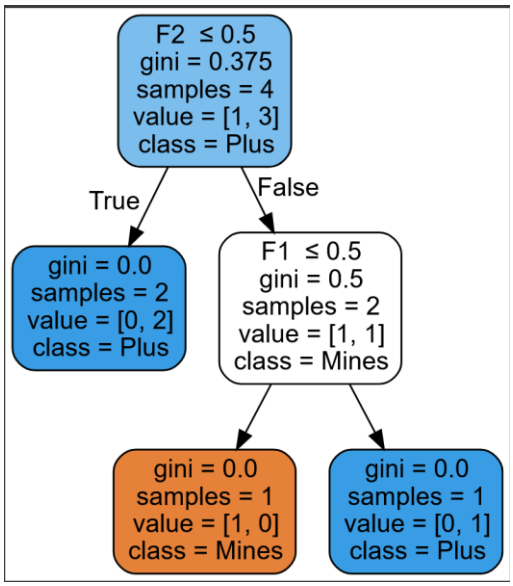
درخت تصمیم یک مدل یادگیری ماشین است که با استفاده از سوالات بله/خیر، داده‌های ورودی را به دسته‌بندی‌های مختلف تقسیم می‌کند. برای ایجاد درخت تصمیم، باید به یک معیار تقسیم (splitting criteria) مراجعه کنیم که بتواند بهترین تقسیم را برای داده‌ها پیدا کند. برخی از معیارهای تقسیم عبارتند از:

- جینی :(Gini impurity) این معیار برای تقسیم داده‌ها به دسته‌هایی با سطح انحراف کمتر استفاده می‌شود.
- انباشت سازی اطلاعات :(Information gain) این معیار برای تقسیم داده‌ها به دسته‌هایی با اطلاعات بیشتر استفاده می‌شود.
- نرخ خطا :(Classification error rate) این معیار برای تقسیم داده‌ها به دسته‌هایی با نرخ خطای کمتر استفاده می‌شود.

استفاده از هر یک از این معیارها بسته به ویژگی‌های داده‌ها و مسئله مورد نظر ممکن است بهتر باشد. برای انتخاب معیار مناسب برای مسئله‌ی خود، بهتر است به ماهیت مسئله، نوع داده‌های ورودی و تعداد ویژگی‌های داده‌ها توجه کنید. همچنین، می‌توانید با آزمایش معیارهای مختلف بر روی داده‌های آموزشی، معیاری را انتخاب کنید که بهترین عملکرد را در مدل شما داشته باشد.

بخش f

برای این بخش نیز کد زده شد، چرا که حل سوال به صورت دستی بسیار طولانی و خسته کننده بود ☹



تصویر ۱ - خروجی قطعه کد برای سوال تئوری

بخشی از محسابه :

Information Gain (IG) = Entropy(S) - [Sum over all values of the feature F] (|Sv| / |S|) * Entropy(Sv)

Entropy(S) = -p+ * log2(p+) - p- * log2(p-)

F1:

$S+ = \{(۴,۱)\}, S- = \{(۳,۰), (۲,۱)\}, Sv+ = \{(۴,۱)\}, Sv- = \{(۳,۰)\}, S۰+ = \{\}, S۰- = \{(۲,۱)\}$

$\text{Entropy}(S+) = ۰, \text{Entropy}(S-) = -(۲/۳) * \log۲(۲/۳) - (۱/۳) * \log۲(۱/۳) = ۰.۹۱۸$

$\text{Entropy}(Sv+) = ۰, \text{Entropy}(Sv-) = ۰ \text{ Entropy}(S۰+) = ۰, \text{Entropy}(S۰-) = ۰$

$IG(F۱) = ۰.۸۱۱ - [(۲/۴) * ۰.۹۱۸ + (۱/۴) * ۰ + (۱/۴) * ۰] = ۰.۳۱۱$

برای باقی موارد نیز به همین صورت محسابه انجام میشود که مقدار IG به ترتیب برای F2، F3، F4و F5 برابراست با ۰,۳۱، ۰,۳۱، ۰,۱۹و ۰,۳۱.

براساس موارد بدست آمده می توانیم مورد با بالاتر IG را استفاده کنیم.

بخش g

درخت تصمیم ممکن است به گونه‌ای بزرگ و پیچیده شود که به داده‌های آموزشی خود بسیار خوب بخورد اما برای داده‌های جدید به خوبی عمل نکند. به این مشکل، بیش‌برازش یا overfitting گفته می‌شود که به معنای این است که مدل به داده‌های آموزشی بسیار عالق شده است و به جای یادگیری الگوهای کلی، به تفاوت‌های جزئی و نویز داده‌ها حساس شده است.

برای جلوگیری از این مشکل، می‌توانیم به جای ساخت یک درخت بسیار بزرگ و پیچیده، درختی با ساختار ساده‌تر و کمترین تعداد گره‌ها و شاخه‌ها بسازیم. این کار را می‌توان با حذف برخی از گره‌ها و شاخه‌هایی که دارای کمترین ارزش و اهمیت هستند، انجام داد. به این کار، "pruning"می‌گویند.

بسته به ساختار و خصوصیات داده‌ها و مسئله مورد نظر، اعمال pruning ممکن است بهبود قابل توجهی در نتایج بیاورد. در مواردی که درخت تصمیم بسیار بزرگ و پیچیده شده است، pruningباعث کاهش overfittingمی‌شود و مدل قابل استفاده‌تر و دقیق‌تر می‌شود. همچنین، pruningباعث کاهش پیچیدگی محاسباتی مدل می‌شود و سرعت یادگیری و پیش‌بینی را بهبود می‌بخشد.

اما در برخی موارد، اعمال pruning ممکن است باعث کاهش دقت و عملکرد مدل شود. به‌عنوان مثال، در صورتی که درخت تصمیم به اندازه کافی ساده باشد، pruningممکن است باعث حذف اطلاعات مهم و الگوهای قابل تشخیص در داده‌ها شود و به‌این‌ترتیب دقت پیش‌بینی را کاهش دهد.

بنابراین، برای استفاده از pruning در درخت تصمیم، باید به دقت و با توجه به خصوصیات مسئله، میزان pruning مناسب را انتخاب کرد. انتخاب این میزان به‌عنوان یک پارامتر مهم در فرایند یادگیری ماشین به شمار می‌آید که نیازمند تجربه و آزمون با مجموعه‌داده‌های مختلف است.

بخش h

در حالت کلی، الگوریتم درخت تصمیم برای حل مسائل دسته‌بندی دو کلاسه (binary classification) طراحی شده است، به‌عنوان مثال، مسئله پیش‌بینی احتمال ابتلا به یک بیماری بر اساس شاخص‌های مختلف. اما اگر مسئله به شکل یک مسئله دسته‌بندی چند کلاسه (multi-class classification) مطرح شود، درخت تصمیم به‌طور پیش‌فرض می‌تواند با استفاده از تعدادی از مفاهیم مانند one-vs-all (OVA) و یا one-vs-one (OVO) به مسئله تطبیق داده شود.

در الگوریتمOVA ، برای حل یک مسئله دسته‌بندی چند کلاسه، برای هر کلاس، یک درخت تصمیم طراحی می‌شود که به‌صورت مستقل از سایر درخت‌ها، مسئله دسته‌بندی بین آن کلاس و سایر کلاس‌ها را حل می‌کند. در الگوریتمOVO ، برای هر دو کلاس ممکن، یک درخت تصمیم طراحی می‌شود که مسئله دسته‌بندی بین دو کلاس را حل می‌کند.

به‌طور کلی، استفاده از درخت تصمیم برای مسائل دسته‌بندی چند کلاسه به نوع مسئله و تعداد کلاس‌ها وابسته است و برای طراحی یک درخت تصمیم مناسب برای هر مسئله، نیاز به تجربه و تحلیل دقیق داریم.

بخش i

دو معیار Information Gain و Gain Ratio در الگوریتم درخت تصمیم به منظور انتخاب بهترین ویژگی برای تقسیم داده‌ها استفاده می‌شوند.

معیار Information Gain از تغییرات اطلاعاتی قبل و بعد تقسیم داده‌ها براساس یک ویژگی خاص برای محاسبه استفاده می‌کند. برای محاسبهInformation Gain ، باید مقدار entroy (آنتروپی) را برای هر گره و سپس برای تمام گره‌های فرزندان حساب کنید و سپس تفاوت بین مقدار entroy قبل و بعد از تقسیم داده‌ها را برای ویژگی مورد نظر محاسبه کنید.

از سوی دیگر، Gain Ratioمقدار Information Gain را با توجه به میزان اطلاعاتی که ویژگی در مورد کلاس‌های مختلف داده‌ها به ما ارائه می‌دهد تعدیل می‌کند. با داشتن این معیار، می‌توان از تمایز اندازه‌گیری‌های Information Gain بین ویژگی‌های با مقادیر بازده یکسان جلوگیری کرد.

به‌طور خلاصه، اگرچه هدف هر دوی این معیارها انتخاب بهترین ویژگی برای تقسیم داده‌ها در الگوریتم درخت تصمیم است، اما از دیدگاه محاسباتی و تحلیلی، این دو معیار با یکدیگر تفاوت دارند.

بخش j

در مسائل رگرسیون چند خروجی، درخت تصمیم می‌تواند برای پیش‌بینی چندین متغیر هدف به صورت همزمان استفاده شود. برای این کار، معمولاً از روش‌های ترکیبی مانند رگرسیون درخت تصمیم (Regression Trees) یا رگرسیون مبتنی بر درخت (Tree-based Regression) استفاده می‌شود.

در این روش‌ها، هر گره از درخت یک مدل رگرسیونی جداگانه را برای پیش‌بینی همه متغیرهای هدف ایجاد می‌کند. سپس به ازای هر نمونه ورودی، متغیرهای هدف با استفاده از مدل‌های رگرسیونی هر گره به طور مستقل پیش‌بینی می‌شوند. در نهایت، پیش‌بینی‌های هر مدل رگرسیونی با هم ترکیب شده و به عنوان خروجی نهایی ارائه می‌شود.

به‌طور خلاصه، درخت تصمیم می‌تواند برای مسائل رگرسیون چند خروجی با استفاده از روش‌های ترکیبی مانند رگرسیون درخت تصمیم یا رگرسیون مبتنی بر درخت استفاده شود. هر گره از درخت یک مدل رگرسیونی جداگانه برای پیش‌بینی همه متغیرهای هدف ایجاد می‌کند و پیش‌بینی‌های هر مدل رگرسیونی با هم ترکیب شده و به عنوان خروجی نهایی ارائه می‌شود.

بخش k

الگوریتم‌های درخت تصمیم CART و C۴.۵ دو الگوریتم محبوب برای ساخت درخت تصمیم هستند. این دو الگوریتم در تفکیک داده‌ها و ساختن گره‌های درخت تصمیم تفاوت‌هایی دارند.

در الگوریتم CART ، برای جداسازی داده‌ها از یک معیار جداسازی به نام Gini impurity استفاده می‌شود، که مقدار آن بین صفر تا یک است. هدف این الگوریتم بهینه کردن Gini impurity برای جداسازی داده‌ها و به دست آوردن بهترین شاخه‌ها برای ساخت درخت تصمیم است.

اما در الگوریتم C۴.۵ ، از معیار Entropy برای جداسازی داده‌ها استفاده می‌شود Entropy .میزان ترکیب و تنوع داده‌ها را نشان می‌دهد و هدف الگوریتم C۴.۵ بهینه کردن این معیار و ساختن گره‌های درخت تصمیم با بهترین معیار جداسازی است.

بنابراین، درخت تصمیم CART و C۴.۵ از دو معیار مختلف برای جداسازی داده‌ها استفاده می‌کنند CART .از Gini impurity استفاده می‌کند و C۴.۵ از Entropy. همچنین، در الگوریتم C۴.۵ برای بالا بردن دقت و بهبود عملکرد درخت تصمیم، از معیار gain ratio برای انتخاب بهترین ویژگی برای ساخت گره‌های درخت استفاده می‌شود gain ratio .علاوه بر در نظر گرفتن تنوع داده‌ها، اندازه نمونه‌ها را نیز در نظر می‌گیرد. در صورتی که برای یک ویژگی، تعداد نمونه‌ها کم باشد، این معیار از آن ویژگی استفاده نمی‌کند.

اما در الگوریتم CART ، از معیار Information gain استفاده می‌شود که تنها توانایی در نظر گرفتن تنوع داده‌ها را دارد و اندازه نمونه‌ها را در نظر نمی‌گیرد.

بنابراین، الگوریتم C۴.۵ از gain ratio برای بهترین کردن درخت تصمیم استفاده می کند، در حالی که الگوریتم CART از Information gainبرای ساخت درخت تصمیم استفاده می کند.

بخش l

در سیستم‌های قائم بر قوانین، قوانین به صورت جداگانه تعریف می‌شوند و سپس اجرا می‌شوند. به عبارت دیگر، این سیستم‌ها به دنبال پیدا کردن یک قانون اولیه هستند و سپس برای هر حالت ممکن، یک قانون خاص تعریف می‌شود.

در حالی که در درخت تصمیم، ساختار درخت توسط داده‌های آموزشی و تعدادی از ویژگی‌های آن‌ها تعیین می‌شود. سپس در هنگام پیش‌بینی برای یک نمونه جدید، از روی مسیر مربوط به آن نمونه در درخت تصمیم، تصمیم گیری می‌شود.

بنابراین، اصلی‌ترین تفاوت بین سیستم‌های قائم بر قوانین و درخت تصمیم، در روش تعیین شرایط تصمیم گیری است. در سیستم‌های قائم بر قوانین، قوانین به صورت جداگانه تعریف می‌شوند و پس از تعریف، در صورتی که برای یک حالت مشخص نتیجه مشخص شده باشد، دیگر قوانین اجرا نمی‌شوند. اما در درخت تصمیم، برای هر حالت، تصمیم گیری به صورت پی در پی انجام می‌شود و هر گره می‌تواند چندین شاخه داشته باشد.

بخش m

عمق درخت تصمیم می‌تواند تأثیر زیادی بر روی تعادل بین انحراف و واریانس داشته باشد. درخت‌های با عمق بیشتر، معمولاً پیچیدگی بیشتری دارند و به همین دلیل ممکن است با مشاهده داده‌های جدید، به طور قابل توجهی دچار بیش‌برازش شوند که باعث افزایش واریانس و کاهش انحراف می‌شود.

به عبارت دیگر، درخت‌های با عمق زیاد تمایل دارند که به صورت دقیق به داده‌های آموزشی بپردازند ولی به نتایج نامطلوبی در پیش‌بینی داده‌های جدید منجر شوند. به همین دلیل، ممکن است نتایج آن‌ها به عنوان یک مدل پیش‌بینی، مطلوب نباشد.

بنابراین، با افزایش عمق درخت، واریانس مدل افزایش می‌یابد و انحراف آن کاهش می‌یابد. به همین دلیل، باید تعادلی بین عمق درخت و تعادل بین انحراف و واریانس در نظر گرفته شود.

سوال دوم

فایل کد در پوشه ی Source Codes با عنوان P2.ipynb قرارداده شده است.

بخش a

(385, 12)

	user_id	weight	height	salads_per_week	veggies_fruits_per_day	healthy_diet	aerobic_per_week	sports_per_week	current_smoking	survey.month	prob_cancer	cancer_category
0	55	140.0	69.0	0.0	NaN	Below average	2.0	0.0	Never	2008.09	0.066120	(0.004486346, 0.2]
1	36	150.0	67.0	2.0	1.0	Below average	3.0	3.0	Never	2008.09	0.366939	(0.2, 0.4]
2	39	105.0	66.0	0.0	2.0	Average	1.0	0.0	Never	2008.09	0.805540	(0.8, 0.998722287]
3	37	220.0	77.0	2.0	5.0	Very healthy	5.0	5.0	Never	2008.09	0.537907	(0.4, 0.6]
4	72	135.0	62.0	0.0	1.0	Unhealthy	0.0	0.0	Never	2008.09	0.098464	(0.004486346, 0.2]

تصویر ۲ – خواندن داده ها

بخش b

(385, 12)

	user_id	weight	height	salads_per_week	veggies_fruits_per_day	healthy_diet	aerobic_per_week	sports_per_week	current_smoking	survey.month	prob_cancer	cancer_category
0	55	140.0	69.0	0.0	NaN	Below average	2.0	0.0	Never	2008.09	0.066120	(0.004486346, 0.2]
1	36	150.0	67.0	2.0	1.0	Below average	3.0	3.0	Never	2008.09	0.366939	(0.2, 0.4]
2	39	105.0	66.0	0.0	2.0	Average	1.0	0.0	Never	2008.09	0.805540	(0.8, 0.998722287]
3	37	220.0	77.0	2.0	5.0	Very healthy	5.0	5.0	Never	2008.09	0.537907	(0.4, 0.6]
4	72	135.0	62.0	0.0	1.0	Unhealthy	0.0	0.0	Never	2008.09	0.098464	(0.004486346, 0.2]

تصویر ۳ – جایگزین کردن داده های خالی قد با میانگین

بخش c

(377, 12)

	user_id	weight	height	salads_per_week	veggies_fruits_per_day	healthy_diet	aerobic_per_week	sports_per_week	current_smoking	survey.month	prob_cancer	cancer_category
0	55	140.0	69.0	0.0	NaN	Below average	2.0	0.0	Never	2008.09	0.066120	(0.004486346, 0.2]
1	36	150.0	67.0	2.0	1.0	Below average	3.0	3.0	Never	2008.09	0.366939	(0.2, 0.4]
2	39	105.0	66.0	0.0	2.0	Average	1.0	0.0	Never	2008.09	0.805540	(0.8, 0.998722287]
3	37	220.0	77.0	2.0	5.0	Very healthy	5.0	5.0	Never	2008.09	0.537907	(0.4, 0.6]
4	72	135.0	62.0	0.0	1.0	Unhealthy	0.0	0.0	Never	2008.09	0.098464	(0.004486346, 0.2]

تصویر ۴ – حذف مواردی که ستون Current_smoking خالی بود

بخش d

(377, 12)

```
<ipython-input-4-afc174e654ff>:3: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise
  data.fillna(data.mean(), inplace=True)
```

	user_id	weight	height	salads_per_week	veggies_fruits_per_day	healthy_diet	aerobic_per_week	sports_per_week	current_smoking	survey.month	prob_cancer	cancer_category
0	55	140.0	69.0	0.0	2.140921	Below average	2.0	0.0	Never	2008.09	0.066120	(0.004486346, 0.2]
1	36	150.0	67.0	2.0	1.000000	Below average	3.0	3.0	Never	2008.09	0.366939	(0.2, 0.4]
2	39	105.0	66.0	0.0	2.000000	Average	1.0	0.0	Never	2008.09	0.805540	(0.8, 0.998722287]
3	37	220.0	77.0	2.0	5.000000	Very healthy	5.0	5.0	Never	2008.09	0.537907	(0.4, 0.6]
4	72	135.0	62.0	0.0	1.000000	Unhealthy	0.0	0.0	Never	2008.09	0.098464	(0.004486346, 0.2]

تصویر ۵ – اقدامات مازاد خواسته شده برای پیش پردازش

بخش e

حذف ستون های اضافه

(377, 10)

	weight	height	salads_per_week	veggies_fruits_per_day	healthy_diet	aerobic_per_week	sports_per_week	current_smoking	survey.month	cancer_category
0	140.0	69.0	0.0	2.140921	Below average	2.0	0.0	Never	2008.09	(0.004486346, 0.2]
1	150.0	67.0	2.0	1.000000	Below average	3.0	3.0	Never	2008.09	(0.2, 0.4]
2	105.0	66.0	0.0	2.000000	Average	1.0	0.0	Never	2008.09	(0.8, 0.998722287]
3	220.0	77.0	2.0	5.000000	Very healthy	5.0	5.0	Never	2008.09	(0.4, 0.6]
4	135.0	62.0	0.0	1.000000	Unhealthy	0.0	0.0	Never	2008.09	(0.004486346, 0.2]

تصویر ۶ – حذف دو ستون اضافه

لیبل انکودینگ

	weight	height	salads_per_week	veggies_fruits_per_day	healthy_diet	aerobic_per_week	sports_per_week	current_smoking	survey.month	cancer_category
0	140.0	69.0	0.0	2.140921	1	2.0	0.0	3	2008.09	0
1	150.0	67.0	2.0	1.000000	1	3.0	3.0	3	2008.09	1
2	105.0	66.0	0.0	2.000000	0	1.0	0.0	3	2008.09	4
3	220.0	77.0	2.0	5.000000	4	5.0	5.0	3	2008.09	2
4	135.0	62.0	0.0	1.000000	3	0.0	0.0	3	2008.09	0

تصویر ۷ – تبدیل داده های غیرعددی به عددی با لیبل انکودینگ

بخش f

داده ها طبق گفته ی صورت سوال به دو بخش آموزش و تست تقسیم شدند، همچنین همه ی داده ها استاندارد سازی شده اند.

بخش g

اقدامات مربوط به آموزش انجام شد و در نهایت نیز کار پیشبینی نیز انجام شده است.

بخش h

```
Confusion Matrix:
[[ 7  4  5  4  4  0]
 [ 9  6  7  4  0  0]
 [ 4  7  5  2  6  0]
 [ 8  4  2  1  0  0]
 [12  5  3  2  1  0]
 [ 0  1  0  0  1  0]]
R2 Score: -1.2833256252445837
Accuracy: 0.17543859649122806
```

تصویر ۸ – خروجی نهایی Confusion Matrix

بخش i

زیربخش اول

صرفا اقدام کردیم به حذف همه ی سطرهایی که اطلاعات یکی از ستون هایش خالی باشد.

```
Confusion Matrix:
[[0 2 2 0 0]
 [1 1 2 3 1]
 [4 1 1 0 2]
 [2 3 2 0 4]
 [0 1 0 1 1]]
R2 Score: -1.2354414527238577
Accuracy: 0.08823529411764706
```

تصویر ۹ – خروجی بدست آمده Confusion Matrix پس از تغییرات اولیه

زیربخش دوم

اطلاعات صرفا داده های ستون قد را با میانگین پر کردیم و بعد از آن آمدیم و داده های اضافه را حذف کردیم

```
Confusion Matrix:
[[ 4  8  3  4  3  1]
 [ 5  6  9  2  2  0]
 [ 8  4  5  0  1  0]
 [ 5  2  3  1  2  0]
 [11  4  8  3  3  0]
 [ 0  1  0  0  0  0]]
R2 Score: -1.0878319229646922
Accuracy: 0.17592592592592593
```

تصویر ۱۰ – خروجی نهایی Confusion Matrix پس از تغییرات ثانویه

سوال سوم

فایل کد در پوشه ی Source Codes با عنوان P3.ipynb قرارداده شده است.

بخش a

The KDDCUP99 dataset is a widely used dataset in the field of network security research. It was created as part of the 1999 KDD Cup data mining competition, and contains network traffic data that has been pre-processed and anonymized.

The dataset consists of a large number of network connections, each of which is labeled as either normal or one of several different types of attack (such as DoS, Probe, or R2L). Each connection is described by 41 different features, including things like the duration of the connection, the typ e of protocol used, and the number of bytes transmitted.

One of the key characteristics of the KDDCUP99 dataset is its imbalanced nature - that is, the nu mber of normal connections greatly outnumber the number of connections with attacks. This can make it challenging to build accurate models for detecting attacks.

بخش b

```
   duration protocol_type service flag  src_bytes  dst_bytes  land  \
0         0           tcp    http   SF        181       5450     0
1         0           tcp    http   SF        239        486     0
2         0           tcp    http   SF        235       1337     0
3         0           tcp    http   SF        219       1337     0
4         0           tcp    http   SF        217       2032     0
5         0           tcp    http   SF        217       2032     0
6         0           tcp    http   SF        212       1940     0

   wrong_fragment  urgent  hot  ...  dst_host_srv_count  \
0               0       0    0  ...                   9
1               0       0    0  ...                  19
2               0       0    0  ...                  29
3               0       0    0  ...                  39
4               0       0    0  ...                  49
5               0       0    0  ...                  59
6               0       0    0  ...                  69

   dst_host_same_srv_rate  dst_host_diff_srv_rate  \
0                     1.0                     0.0
1                     1.0                     0.0
2                     1.0                     0.0
3                     1.0                     0.0
4                     1.0                     0.0
5                     1.0                     0.0
6                     1.0                     0.0

   dst_host_same_src_port_rate  dst_host_srv_diff_host_rate  \
0                         0.11                         0.00
1                         0.05                         0.00
2                         0.03                         0.00
3                         0.03                         0.00
4                         0.02                         0.00
5                         0.02                         0.00
6                         1.00                         0.04

   dst_host_serror_rate  dst_host_srv_serror_rate  dst_host_rerror_rate  \
0                   0.0                       0.0                   0.0
1                   0.0                       0.0                   0.0
2                   0.0                       0.0                   0.0
3                   0.0                       0.0                   0.0
4                   0.0                       0.0                   0.0
5                   0.0                       0.0                   0.0
6                   0.0                       0.0                   0.0

   dst_host_srv_rerror_rate   label
0                       0.0  normal
1                       0.0  normal
2                       0.0  normal
3                       0.0  normal
4                       0.0  normal
5                       0.0  normal
6                       0.0  normal

[7 rows x 42 columns]
```

تصویر ۱۱ – نمایش ۷ مورد اولیه مجموعه داده ها

بخش c

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_srv_count	dst_host_same_srv_rate	dst_host_diff_srv_rate	dst_host_same_src_port_rate
0	0	1	22	9	181	5450	0	0	0	0	...	9	1.0	0.0	0.11
1	0	1	22	9	239	486	0	0	0	0	...	19	1.0	0.0	0.05
2	0	1	22	9	235	1337	0	0	0	0	...	29	1.0	0.0	0.03
3	0	1	22	9	219	1337	0	0	0	0	...	39	1.0	0.0	0.03
4	0	1	22	9	217	2032	0	0	0	0	...	49	1.0	0.0	0.02
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
494015	0	1	22	9	310	1881	0	0	0	0	...	255	1.0	0.0	0.01
494016	0	1	22	9	282	2286	0	0	0	0	...	255	1.0	0.0	0.17
494017	0	1	22	9	203	1200	0	0	0	0	...	255	1.0	0.0	0.06
494018	0	1	22	9	291	1200	0	0	0	0	...	255	1.0	0.0	0.04
494019	0	1	22	9	219	1234	0	0	0	0	...	255	1.0	0.0	0.17

494020 rows × 42 columns

تصویر ۱۲ – تصویر پس از اعمال پیش پردازش های لازم

بخش d

در صورت سوال خواسته شده بود با یک مدل این کار انجام شود اما ما با دو مدل این کار را کردیم و آموزش مدل KNN و DT انجام شد و در بخش بعدی خروجی میزان Accuracy نمایش داده خواهد شد.

بخش e

```
Decision Tree Accuracy: 0.9994332213270718
KNN Accuracy: 0.9984312376017165
```

تصویر ۱۳ – خروجی Accuracy مدل های DT و KNN

سوال چهارم

فایل کد در پوشه ی Source Codes با عنوان P4.ipynb قرارداده شده است.

بخش a1

	f1	f2	f3	f4	f5	f6	target
0	M01	A	20	N	N	Y	+
1	M01	A	20	?	?	?	+
2	M01	A	30	Y	Y	Y	+
3	M01	A	50	N	Y	Y	+
4	M01	A	55	Y	Y	N	+

تصویر ۱۴ – نمایش ۵ مورد ابتدایی مجموعه داده ها

بخش a2

اقدامات مربوط به پیش پردازش در این بخش انجام شد.

	f3	target	f1_M01	f1_M02	f2_A	f2_B	f2_C	f2_D	f4_?	f4_N	f4_Y	f5_?	f5_N	f5_Y	f6_?	f6_N	f6_Y
0	20	+	1	0	1	0	0	0	0	1	0	0	1	0	0	0	1
1	20	+	1	0	1	0	0	0	1	0	0	1	0	0	1	0	0
2	30	+	1	0	1	0	0	0	0	0	1	0	0	1	0	0	1
3	50	+	1	0	1	0	0	0	0	1	0	0	0	1	0	0	1
4	55	+	1	0	1	0	0	0	0	0	1	0	0	1	0	1	0

تصویر ۱۵ – پیش پردازش های انجام شده

بخش b

```
from sklearn.model_selection import train_test_split

X = df.drop('target', axis=1)
y = df['target']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, stratify=y, random_state=42)
```

تصویر ۱۶ – تقسیم داده های به تست و آموزش

بخش c

```
from sklearn.tree import DecisionTreeClassifier

clf = DecisionTreeClassifier()
```

تصویر ۱۷ – ساخت مدل ساده

بخش d

```
clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)
```

تصویر ۱۸ – مدل سازی مدل DT

بخش e

```
Accuracy: 0.75
Precision: 0.8823529411764706
Recall: 0.8333333333333334
F1 score: 0.8571428571428571
```

تصویر ۱۹ – خروجی مربوط به معیار های خواسته شده

بخش f

تصویر ۲۰ – خروجی درخت تصمیم خواسته شده

بخش g

```
Best parameters: {'max_depth': 2, 'min_samples_split': 2}
Best score: 0.8222222222222222
```

تصویر ۲۱ – بدست آوردن پارامترها با GridSearch

بخش h

```
Accuracy: 0.75
Precision: 0.8823529411764706
Recall: 0.8333333333333334
F1 score: 0.8571428571428571
```

تصویر ۲۲ – محاسبه معیار های خواسته شده

بخش i

برای ارزیابی عملکرد بهترین درخت تصمیم بر روی داده های آزمون، می توانیم از معیارهای ارزیابی مشابه قبلی استفاده کنیم. بله، درخت تصمیم می تواند وضعیت عبور یا شکست را بر اساس ویژگی های داده شده تعیین کند.

سوال پنجم

فایل کد در پوشه ی Source Codes با عنوان P5.ipynb قرارداده شده است.

بخش a

در این بخش ابتدا داده ها را لود میکنیم و ۳ کار روی داده ها انجام میدهیم، حذف ستون های بدون استفاده، پرکردن داده های از دست رفته و انجام لیبل انکودینگ، پس از انجام این مراحل خروجی دیتاست به شکل مقابل است:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked_C	Embarked_Q	Embarked_S
0	0	3	1	22.0	1	0	7.2500	0	0	1
1	1	1	0	38.0	1	0	71.2833	1	0	0
2	1	3	0	26.0	0	0	7.9250	0	0	1
3	1	1	0	35.0	1	0	53.1000	0	0	1
4	0	3	1	35.0	0	0	8.0500	0	0	1

تصویر ۲۳ – خروجی پس از اعمال موارد خواسته شده از جمله Label Encoding

بخش b

دیتاست را اول به دو بخش X و y تقسیم میکنیم که y مقدار ستون Survived می باشد و در ادامه نیز داده ها را طبق خواسته صورت سوال به دو بخش تست و آموزش با اندازه تست ۲۰ درصدی تقسیم میکنیم.

بخش c

```
Best hyperparameters: {'criterion': 'gini', 'max_depth': 7, 'min_samples_leaf': 2, 'min_samples_split': 10}
```

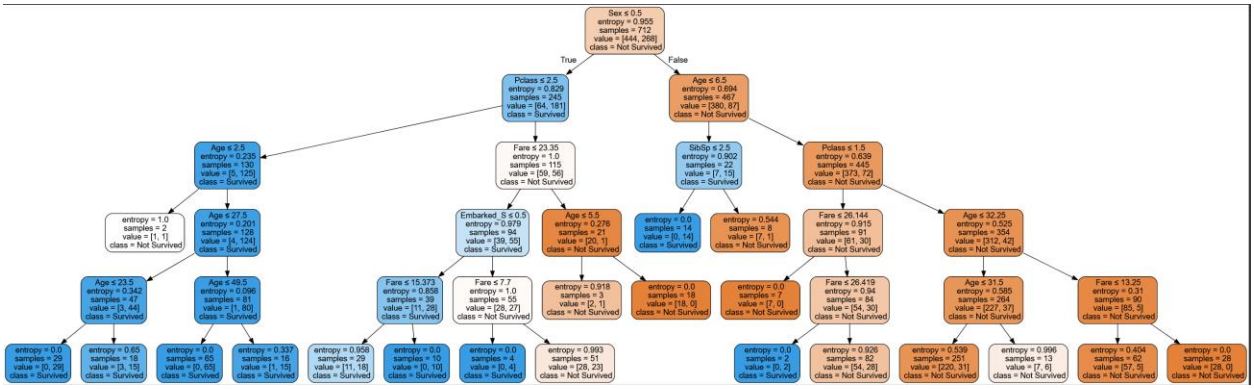
تصویر ۲۴ – محاسبه پارامتر های برتر

بخش d

```
Accuracy: 0.8044692737430168
Precision: 0.8305084745762712
Recall: 0.6621621621621622
F1 Score: 0.7368421052631579
```

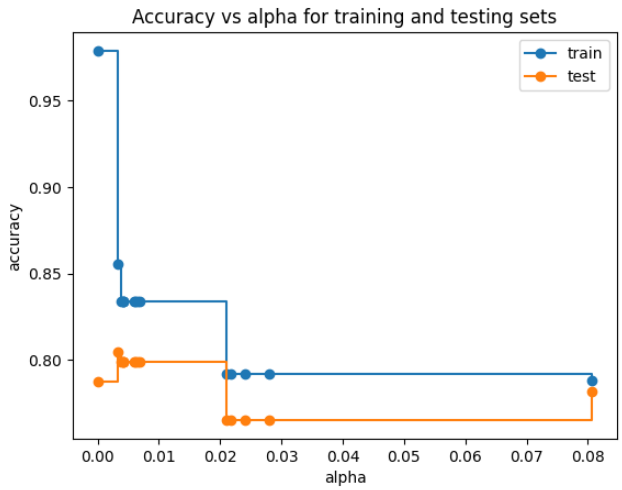
تصویر ۲۵ – محاسبه معیار های خواسته شده

بخش e



تصویر ۲۶ – درخت تصمیم بدست آمده در حالت اول

بخش f

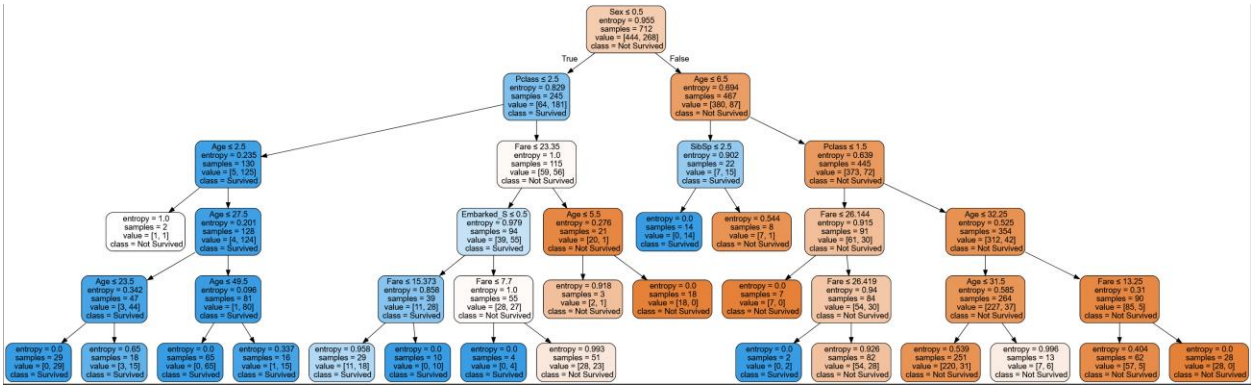


تصویر۲۷ – نمودار Accuracy vs Alpha در تست و آموزش

```
Pruned Decision Tree Performance:
Accuracy: 0.8044692737430168
Precision: 0.8421052631578947
Recall: 0.6486486486486487
F1 Score: 0.7328244274809161
```

تصویر ۲۸- محاسبه معیارها در حالت دوم

بخش g



تصویر ۲۹ - درخت تصمیم در حالت دوم

```
Unpruned Decision Tree Performance:
Accuracy: 0.8044692737430168
Precision: 0.8305084745762712
Recall: 0.6621621621621622
F1 Score: 0.7368421052631579
```

تصویر ۳۰ - خروجی معیارها در حالت دوم