سوال 7

Bootstrapping به صورت خودکار دستورات را Load و اجرا می کند. در خودگردان سازی به جای آن که بر مبنای یک نمونه(نمونه اصلی) به تخمین خطای معیار دست بزنیم با انجام نمونه گیری های فرعی متعدد که اغلب بیشتر از 200 بار است به یک توزیع نمونه ای تجربی دست میابیم که مبنای محاسبه خطای معیار قرار داده می شود. به این توزیع نمونه ای تجربی ، توزیع نمونه ای خودگردان یا Bootstrap گفته می شود. مهمترین تفاوتی که باید گفته شود این است که در روش بدون  Bootstrap شرط نرمال بودن توزیع داده ها وجود دارد اما در استنباط از طریق نمونه گیری خودگردان یا Bootstrap چنین شرطی وجود ندارد. برای مثال برای ایجاد «فاصله‌های اطمینان» (Confidence Interval)، «مدل‌های رگرسیونی» (Regression Models) و همچنین در حوزه «یادگیری ماشین» (Machine Learning) برای تخمین خطای مدل، می‌توان از روش بوت‌استرپ استفاده کرد.

اما درcross validation ، تکیه بر داده‌هایی است که مشاهده شده‌اند ولی در هنگام ساختن مدل به کار گرفته نمی‌شوند. این داده‌ها به منظور بررسی و سنجش کارایی مدل برای پیش‌بینی داده‌‌های جدید به کار می‌روند. در هر مرحله از فرایند CV، مدل بدست آمده توسط داده‌های آزمایشی برای پیش‌بینی داده‌های CV به کار گرفته وخطا یا دقتحاصل از برازش مدل روی داده‌های CV محاسبه می‌شود. معمولا میانگین این خطاها به عنوان خطای کلی مدل در نظر گرفته می‌شود.

سوال 9

ID3

در یک درخت تصمیم، مهم است که کدام یک از ویژگی‌ها (یا همان ابعاد) را در سطوح بالاتری از درخت انتخاب کنیم تا به طبقه‌بندی کمک کند. **الگوریتم ID3 وظیفه پیدا کردنِ ویژگی‌هایی دارای اطلاعات زیادتر (Gain بیشتر) را دارد و آن‌ها را در سطوحِ بالاتری از درخت قرار می‌دهد.** هر بار که یک ویژگی در سطحی از درخت انتخاب شد، زیر درخت‌های آن نیز دقیقا 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 ساختِ زیر درخت‌های دیگر، مجموعه داده‌ها بر اساس مقدارِ 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. این الگوریتم، درختانِ تصمیمِ از بالا به پایین می‌سازد و با طرح این سوال که چه صفتی باید در ریشه‌ی درخت آزمایش شود آغاز می‌کند. برای پاسخ به این سوال، با استفاده از یکی از انواع آزمایش‌های آماری برای تعیین مناسب‌ترین صفت برای دسته‌بندی مثال‌های آموزشی، تصمیم براساس هر صفت نمونه را ارزیابی می‌کند. سپس بهترین صفت را انتخاب کرده و به عنوان تست در گره‌ی ریشه‌ی درخت استفاده می‌کند. برای هر مقدار ممکن صفت تست شده در ریشه، یک گره‌ی متناظر ایجاد شده و مثال‌های آموزشی براساس مقادیر صفت تست، بین این گره‌ها افراز می‌شوند.

CART

CART که خود مخفف Classification And Regression Tree است بر اساس درخت های دودویی(باینری) بنا نهاده شده است. در این درس میخواهیم بیشتر با نحوه ساخت درخت CART آشنا شویم. این درخت میتواند پایه ای برای الگوریتم های پیچیده تر مانند جنگل تصادفی(Random Forest) باشد. الگوریتم درخت تصمیم CART برای ساخت درخت تصمیم، داده ها را به قسمت های دو دویی تقسیم کرده و بر اساس آن ها درخت دو دویی(باینری) را می سازد. الگوریتم CART نوعی الگوریتم طبقه بندی است که برای ساخت درخت تصمیم بر اساس شاخص ناخالصی جینی مورد نیاز است. این یک الگوریتم اساسی یادگیری ماشین است و طیف گسترده ای از موارد استفاده را ارائه می دهد.

**سوال 13**

**هرس‌‌کردن(Pruning) :**

زمانی‌‌که زیرگره‌‌های یک گره تصمیم را حذف می‌‌کنیم، اصطلاحا درخت را هرس کرده‌‌ایم. هرس کردن به معنی حذف شاخه‌‌هایی از درخت است که از ویژگی‌‌هایی منشعب شده است که کم اهمیت هستند. با این روش پیچیدگی درخت را کاهش و قدرت پیش‌‌بینی مدل را افزایش می‌‌دهیم. هرس کردن می‌‌تواند از ریشه یا برگ آغاز شود.

هرس کردن به دو روش pre-pruning و post-pruning صورت می‌‌گیرد.

**هرس قبل- پیش‌‌هرس (Pre-pruning) :**

در این شیوه، هرس کردن قبل از ساخت کامل درخت می‌‌باشد. به این صورت که به درختی که در حال رشد است اجازه رشد بیش از حد داده نشود. مثلا به گره تصمیمی می‌‌رسیم که در آن‌‌جا 4 آیتم مثبت و 1 آیتم منفی داریم و با توجه به آن‌‌که تمامی شروط توقف رعایت نشده است و می‌‌توان هم‌‌چنان از این گره، رشد درخت را ادامه داد ولی در این مرحله این گره تصمیم را به برگ تبدیل می‌‌نماییم و مقدار آن را بر‌‌اساس آیتم با مقادیر بیشتر که در مثال بالا آیتم‌‌های مثبت است برچسب‌‌گزاری می‌‌کنیم.

**هرس بعد (Post-pruning):**

در این شیوه هرس کردن، درخت به صورت کامل ساخته می‌‌شود و سپس عملیات هرس کردن درخت آغاز می‌‌شود. به این‌‌صورت که از پایین درخت یا همان برگ‌‌ها به سمت ریشه حرکت می‌‌کنیم و یک‌‌سری گره‌‌های میانی را تبدیل به برگ می‌‌کنیم. این روش هرس کردن از شیوه‌‌ی قبل کمی کندتر است ولی دارای دقت بیشتری می‌‌باشد.

گزارش تمرین 1

ابتدا کتابخانه های مورد نیاز را مشخص شده اند و فراخوانی دیتاست از کگل در مرحله بعد انجام میشود.بطور رندوم 5 ردیف اول دیتا را بعنوان خروجی گرفتم و قابل مشاهده است.قسمت اینفو هم اطلاعات مربوط به دتاست را نشان میدهد.

فیچر قیمت را به دوکلاس ارزان و گران با صفر و یک تقسیم میکنیم.

درواقع مقادیر صفر و یک را صفر و مقادیر دو و سه را یک قرار میدهد.

کد 14 یک فیچر لیست خالی میسازد.که هر ستون چک شود و بهترین آن انتخاب شود.

کد 15 الگوریتم forward selection را طراحی و اجرا میکند.یک مقدار بعنوان best feature بیان شده که همان بهترین ویژگی را در نهایت نشان میدهد.

در ادامه نرمالسازی داده ها بین صفر و یک انجام شده با استفاده از روش z\_score

Z=(x-u)/s (s واریانس) و نمایش نرمالسازی را هم داریم.

کد 18 خروجی فوروارد سلکشن رفته شده و فیچر ها و 5 فیچر بهتربه همراه خطاهای آنها نمایش داده شده اند.

یک نمودار با دو مولفه feature rank ,score بدست آوردم(با توجه به خروجی و نتایج فوروارد سلکشن)

سپس مد لاجستیک پیاده سازی شده با انتخاب همان 6 تا از بهترین ویژگی ها.

و نتایج سه معیار خواسته شده در سوال2 گزارش شده اند.

دراین قسمت pca پیاده سازی شده با استفاده از5 کامپوننت.

درواقع ما 20 ستون دیتاست را به 5 ستون تغییرو کاهش دادیم.مجددا معیارهای خواسته شده را استفاده و نتایج دیده میشود.میتوان دید که میزان خطا دراین مرحله کمتر شده است.

روش باینینگ روی فیچر باتری پاور انجام شده با سه اندازه مختلف برای بین ها.

و نتیجه و درواقع ماتریس آن را میتوان روی نمودار مشاهده کرد.10 کتگوری در نهایت خواهیم داشت که تعداد هر کتگوری نیز بدست آمده.

یک فیچر جدید به نام area به دیتاست اضافه شده و دوباره مراحل قبل روی آن انجام میشود.

فیچرهای کتگوریکال انتخاب و مشخص شده و one hot encoding اعمال میشود.درنهایت نتایج روی سه row برای نمونه قابل مشاهده اند.

دو مدل درخت تصمیم با دو روش طراحی شده اند کد 43 و 45 .