پروژه دوم درس داده‌کاوی

دانشگاه شهید بهشتی

دانشکده علوم ریاضی

حمیدرضا فیروزه(400422147)

استاد: دکتر هادی فراهانی

دستیار استاد: مهندس علی شریفی

بهار 1401

فهرست مطالب

1- مقدمه

هدف این پروژه بررسی دو دیتاست ارائه شده در سایت Kaggle است. دیتاست اول اطلاعات واقعی در مورد ویژگی های موبایل ها و قیمت آنها است. دیتاست دوم هم اطلاعات مرتبط با ملک‌های ارائه شده جهت اجاره در آلمان است.

به منظور بررسی این داده‌ها از Google Colab استفاده شده است. برای هر دیتاست یک سری مراحل به ترتیب اجرا شده است.

2- دیتاست یک

این مجموعه داده‌ها شامل داده‌های واقعی از موبایل های به همراه ویژگی های موبایل و قیمت آنهاست. در این دیتاست تسک های زیر انجام گرفته است:

تسک 1

1- روش انتخاب ویژگی Forward Selectionرا پیاده سازی کنید . برای معیار انتخاب فیچر جدید در هر مرحله از AUCاستفاده کنید . استفاده از پکیج مجاز نمی باشد و باید این بخش را خودتان پیاده سازی کنید. برای سادگی این پیاده سازی موبایل ها را به دو گروه قیمت بالا (دو کلاس گران را ادغام کنید و یک کلاس در نظر بگیرید) و گروه با قیمت پایین تقسیم بندی کنید. در روش انتخاب پیشرو ما از یک مجموعه تهی شروع کرده و در هرگام سعی داریم فیچر را به مجموعه فیچرهای انتخابی اضافه کنیم که AUCرا افرایش دهد .

برای این بخش ابتدا تابع log_reg را تعریف کردیم که خروجی آن AUC اسکور مدل برای ورودی X و y خواهد بود. سپس تابع forward_selection تعریف شده است. این تابع سه وردی لیست بهترین فیچرهای انتخاب شده، لیست فیچر های باقی مانده و اسکور مدل برای لیست فیچر های فعلی را دریافت می کند.

این فانکشن بعد از دریافت لیست بهترین فیچر های انتخاب شده، از بین فیچرهای باقیمانده، فیچری را انتخاب می کند که اسکور مدل را بهبود میدهد. در پایان بترین فیچر و اسکور مربوطه را برمی گرداند.

در بلوک بعدی با نوشتن یک لوپ در هر مرحله یک فیچر انتخاب می شود تا مدل بهتری داشته باشیم و این کار تا جایی ادامه می یابد که اسکور مدل بهتر می شود.

اجرای این تابع 7 فیچر زیر انتخاب می شود:

```
ram 0.507
1
battery_power 0.912
2
px_height 0.9355
3
px_width 0.9674999999999999
4
mobile_wt 0.988
5
four_g 0.992
6
talk_time 0.9930000000000001
7
 0.9935
```

تسک 2

2- با استفاده از کد پیاده سازی شده در بخش قبل به انتخاب ویژگی ها از فیچر ها بپردازید و سپس مدل لجستیک (با استفاده از پکیج) را بر روی فیچرهای انتخاب شده اجرا کنید.معیار های f1-score، recall، precision را گزارش کنید.

اسکورهای مرتبط با مدل به صورت زیر خواهد بود.

```
f1_score = 0.9945027486256871, recall = 0.995, percision_score =
                                                0.994005994005994
```

تسک 3

3- با استفاده از الگوریتم PCAدر حالتی که تعداد Component ها با تعداد فیچرهای انتخابی حاصل روش انتخاب ویژگی پیشرو برابر باشد (یعنی اگر در سوال ۱شما با استفاده از انتخاب ویژگی پیشرو به طور مثال ۵فیچر را انتخاب کردید در الگوریتم PCA هم به عنوان آرگومان ورودی تعداد Component را ۵درج کنید) دیتاست را تغییر دهید.

```
0    float64
1    float64
2    float64
3    float64
4    float64
5    float64
6    float64
dtype: object
```

تسک 4

4- با استفاده از دیتاست تغییر یافته در سوال قبلی و به کمک پکیج ها یک رگرسیون لجستیک را پیاده سازی کنید و معیار های f1-score، recall، precisionرا گزارش کنید.

خروجی اسکور بر اساس خروجی PCA

```
f1_score = 0.9924962481240621, recall = 0.992, percision_score =
                                                0.992992992992993
```

تسک 5

5- مهندسی ویژگی یکی از بخش های مهم در فرایندهای یادگیری ماشین میباشد . بر روی دیتاست موارد زیر را اجرا کنید .

a. بر روی فیچر battery power از روش binning استفاده کنید .)حداقل سه اندازه مختلف برای بین ها در نظر بگیرید و حتی سایز بین ها را نامساوی در نظر بگیرید

برای این بخش ابتدا امار توصیفی این فیچر را محاسبه می کنیم.

```
count    2000.000000
mean     1238.518500
std       439.418206
min       501.000000
25%       851.750000
50%      1226.000000
75%      1615.250000
max      1998.000000
```

بر اساس این گزراش سه بین تعریف می کنیم که بین اول شامل چارک اول، بین دوم شامل چارک دوم و سوم و بین سوم شامل چارک چهارم خواهد بود.

b. بر فیچرهای کتگوریکال در دیتاست one hot encoding را اعمال کنید . چرا ما باید به صورت کلی از این کدگذاری بهره ببریم .

در این دیتاست تمامی داده ها یا عددی هستند یا بولین، وداده ای وجود ندارد که به صورت کتگوریکال در نظر گرفته شود. با این حال می توان داده تبدیل شده battery_power را به صورت کتگوریکال در نظر گرفت و با استفاده از onhotencoder کار ترنسفورم را انجام داد.

Onhotnecoder داده های گتگوریکال ما را مفیدتر و قابل توضیح می‌سازد با استفاده از مقادیر عددی، ما به راحتی احتمال ارزش‌های خود را تعیین می‌کنیم. به طور خاص، یک onhotencoder داده های توصیفی را به مقادیر عدید تبدیل می کنید. علاوه بر این با این روش نحوه تبدیل به این اعداد به گونه ای خواهد بود که داده ها نسبت به هم اولویت نداشته باشند. مثل داده رنگ. رنگ قرمز نسبت به رنگ سبز اولیتی ندارد ولی استفاده از ordinalencoder باعث ترتیب دادن به داده ها خواهد شد و مدل یادگیری را گمراه کند.

c. بررسی کنید آیا استفاده از تبدیل هایی از قبیل log transform و یا تبدیل نمایی در اینجا کاربرد دارد . به صورت کلی چرا از این دست تبدیلات بهره میبریم . (در این بخش شما مجاز هستید اگر تبدیل دیگری را مناسب میدانید اعمال کنید این بخش نمره امتیازی برای شما خواهد داشت . حتما دلیل استفاده از تبدیل استفاده شده را بیان کنید).

d. یک فیچر جدید به نام مساحت یا حجم گوشی بسازید.

تسک 6

6- برای هریک از حالت های سوال قبلی یک مدل رگرسیون لجستیک بسازید و بررسی کنید یکبار هم هر ۵حالت را باهم اعمال کنید و مدل رگرسیون لجستیک روی آنها اجرا کنید . حاصل این مدل ها را گزارش کنید .

حالت 1: 0.904

حالت 2: 0.5529999999999999

حالت 3: 0.5529999999999999

تمامی حالت ها: 0.8955

تسک 7

7- Bootstrapping چیست و چه تفاوتی با Cross Validationدارد؟ در کجا استفاده میشود؟

Bootstrapping هر آزمون یا معیاری است که بر نمونه‌گیری تصادفی با جایگزینی تکیه دارد و روشی است که در بسیاری از موقعیت‌ها مانند اعتبارسنجی عملکرد یک مدل پیش‌بینی‌کننده کمک می‌کند. ما می توانیم این روش را چندین بار تکرار کنیم و میانگین امتیاز را به عنوان تخمین عملکرد مدل خود محاسبه کنیم.

Cross Validation روشی برای اعتبارسنجی عملکرد یک مدل است و با تقسیم داده های آموزشی به k قسمت انجام می شود. ما فرض می کنیم که قسمت های k-1 مجموعه آموزشی است و قسمت دیگر مجموعه تست ما است. می‌توانیم هر بار با نگه داشتن بخش متفاوتی از داده‌ها، این k بار را به‌طور متفاوت تکرار کنیم. در نهایت، میانگین نمره k را به عنوان تخمین عملکرد خود در نظر می گیریم. با افزایش تعداد تقسیم ها، واریانس نیز افزایش می یابد و بایاس کاهش می یابد. از سوی دیگر، اگر تعداد تقسیم‌ها را کاهش دهیم، بایاس افزایش و واریانس کاهش می‌یابد.

تسک 8

8- 5X2 Cross Validation را در یک پاراگراف توضیح دهید سپس بیان کنید در چه جاهایی استفاده از این روش میتواند مفید باشد.

در این متد 5 بار Cross Validation در حالتی که فولد ها برابر با 2 است. این روش باعث کاهش خطای نوع یک در محاسبه پارامترها خواهد شد.

تسک 9

9- در خصوص الگوریتم های مختلف ساخت درخت تصمیم (همانند CART، ID3و...) تحقیق کنید . به صورت کلی تفاوت الگوریتم های مختلف ساخت درخت تصمیم در چیست؟

الگوریتم CART تنها دو درخت دودویی تولید می‌کند: گره‌های غیر برگ همیشه دو فرزند دارند. برعکس، الگوریتم‌های درخت دیگر مانند ID3 می‌توانند درختان تصمیم‌گیری را با گره‌هایی که بیش از دو فرزند دارند، تولید کنند. به طور کلی تفاوت این درخت ها، در نحوه انتخاب ویژگی برای تقسیم در هر مرحله و چگونگی تقسیم این فیچرها است.

تسک 10

10- به دلخواه با استفاده از پکیج ها بر روی دیتاست مطرح شده یک درخت تصمیم بسازید .

درخت تصمیم با استفاده از متد DecisionTreeClassifier از کتابخانه sklearn فیت شده. در ابتدا هیچ محدودیتی برای درخت ایجاد نمی کنیم. اما در تسک بعدی با تعیین محدودیت درخت را هرس می کنیم.

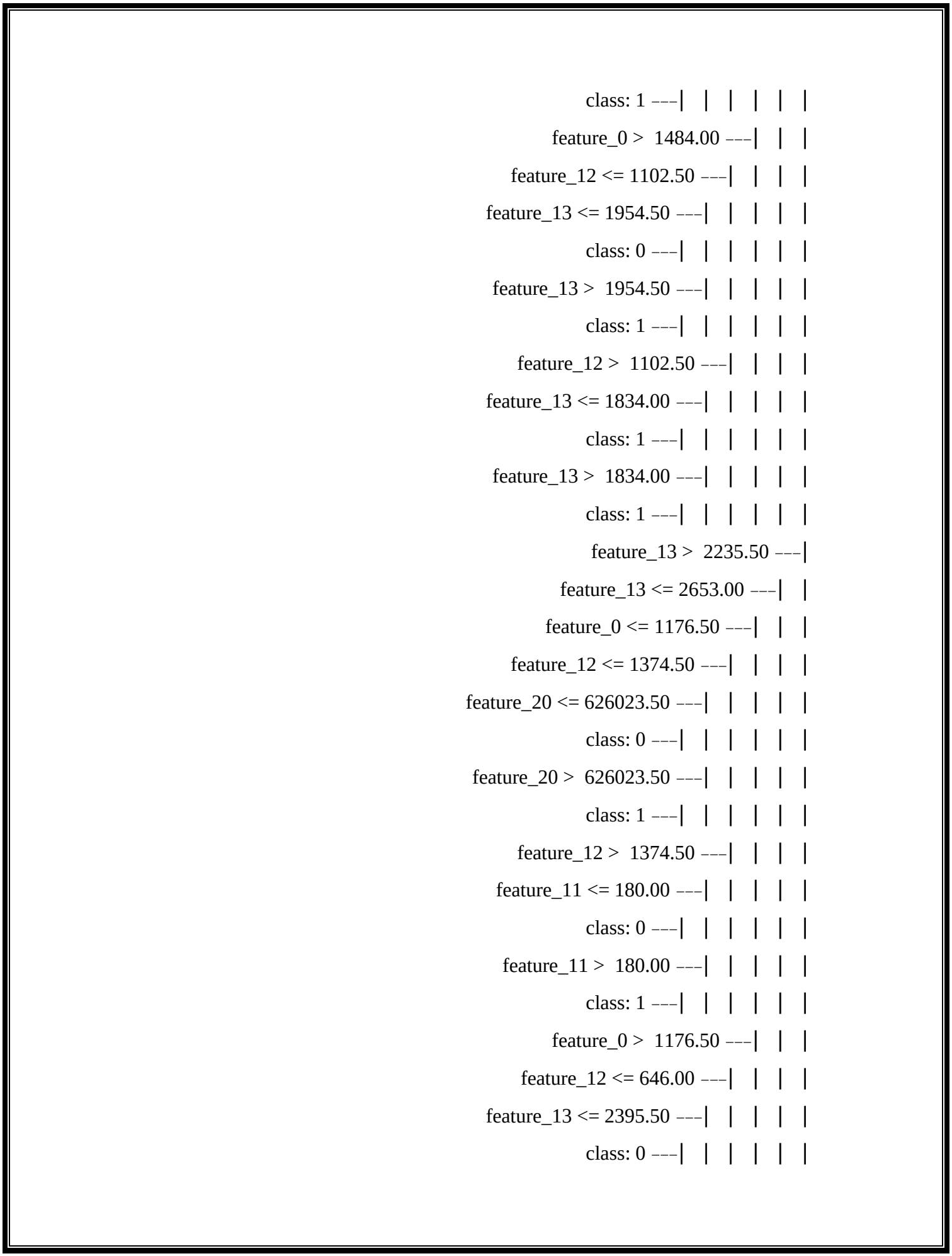
تسک 11

11- برای درخت تصمیم پارامتر های مختلف مورد ارزیابی قرار دهید . آیا عمق درخت و تعداد نمونه های موجود در هر هر گره تاثیری در عملکرد درخت تصمیم دارد؟

```
clf.get_depth()=5
clf.get_n_leaves=59
clf.get_params()
   {'ccp_alpha': 0.0,
    'class_weight': None,
    'criterion': 'gini',
    'max_depth': None,
    'max_features': None,
    'max_leaf_nodes': None,
    'min_impurity_decrease': 0.0,
    'min_samples_leaf': 1,
    'min_samples_split': 2,
    'min_weight_fraction_leaf': 0.0,
    'random_state': 0,
       'splitter': 'best'}
```

درخت زیر با در نظر گرفتن max-depth برابر با 5 در نظر گرفته شده است.

```
feature_13 <= 2235.50 –––|
feature_13 <= 1569.50 –––|   |
feature_20 <= 2369522.00 –––|   |   |
feature_13 <= 1481.00 –––|   |   |   |
class: 0 –––|   |   |   |   |
feature_13 >  1481.00 –––|   |   |   |
feature_20 <= 1711768.00 –––|   |   |   |   |
class: 0 –––|   |   |   |   |   |
feature_20 >  1711768.00 –––|   |   |   |   |
class: 1 –––|   |   |   |   |   |
feature_20 >  2369522.00 –––|   |   |
feature_13 <= 1180.00 –––|   |   |   |
class: 0 –––|   |   |   |   |
feature_13 >  1180.00 –––|   |   |   |
feature_0 <= 1434.00 –––|   |   |   |   |
class: 0 –––|   |   |   |   |   |
feature_0 >  1434.00 –––|   |   |   |   |
class: 1 –––|   |   |   |   |   |
feature_13 >  1569.50 –––|   |
feature_0 <= 1484.00 –––|   |   |
feature_20 <= 2020519.00 –––|   |   |   |
feature_20 <= 1423251.50 –––|   |   |   |   |
class: 0 –––|   |   |   |   |   |
feature_20 >  1423251.50 –––|   |   |   |   |
class: 0 –––|   |   |   |   |   |
feature_20 >  2020519.00 –––|   |   |   |
feature_0 <= 1078.00 –––|   |   |   |   |
class: 0 –––|   |   |   |   |   |
feature_0 >  1078.00 –––|   |   |   |   |
```


class: 1
feature_0 > 1484.00
feature_12 <= 1102.50
feature_13 <= 1954.50
class: 0
feature_13 > 1954.50
class: 1
feature_12 > 1102.50
feature_13 <= 1834.00
class: 1
feature_13 > 1834.00
class: 1
feature_13 > 2235.50
feature_13 <= 2653.00
feature_0 <= 1176.50
feature_12 <= 1374.50
feature_20 <= 626023.50
class: 0
feature_20 > 626023.50
class: 1
feature_12 > 1374.50
feature_11 <= 180.00
class: 0
feature_11 > 180.00
class: 1
feature_0 > 1176.50
feature_12 <= 646.00
feature_13 <= 2395.50
class: 0

```
|   |   |   |   |--- feature_13 >  2395.50
|   |   |   |   |--- class: 1
|   |   |   |--- feature_12 >  646.00
|   |   |   |   |--- feature_13 <= 2309.50
|   |   |   |   |--- class: 1
|   |   |   |   |--- feature_13 >  2309.50
|   |   |   |   |--- class: 1
|   |   |--- feature_13 >  2653.00
|   |   |   |--- feature_0 <= 569.50
|   |   |   |   |--- feature_12 <= 570.00
|   |   |   |   |--- class: 0
|   |   |   |   |--- feature_12 >  570.00
|   |   |   |   |--- feature_0 <= 566.00
|   |   |   |   |--- class: 1
|   |   |   |   |--- feature_0 >  566.00
|   |   |   |   |--- class: 0
|   |   |   |--- feature_0 >  569.50
|   |   |   |--- class: 1
```

تسک 12

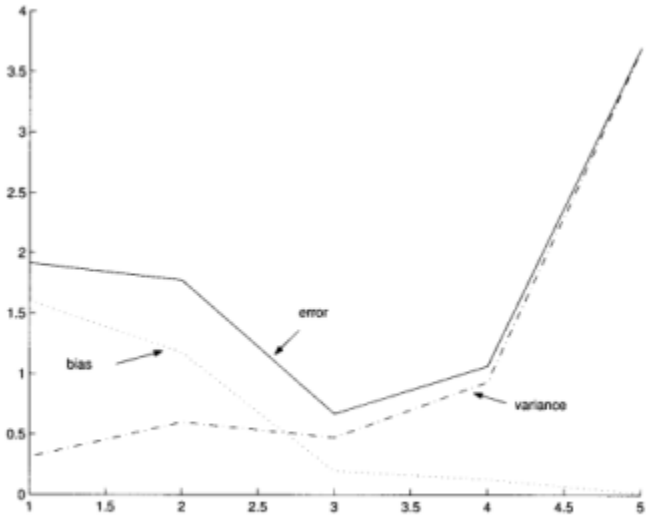
12- در خصوص هرس کردن Pruning درخت تصمیم تحقیق کنید . چرا ما به بحث هرس کردن درخت تصمیم نیاز دارد و چه کمکی به ما میکند

هرس یک درخت تصمیم‌گیری به جلوگیری از overfitting داده‌های آموزشی کمک می‌کند تا مدل ما به خوبی داده‌های تست را تعمیم دهد. هرس درخت تصمیم‌گیری به معنای حذف زیر درختی است که اضافی است و انشعاب مفیدی نیست و آن را با یک‌گره برگ جایگزین می‌کند.

برای این پروژه مانند تسک قبل مقدار عمق حداکثر درخت را برابر 5 در نظر گرفتیم. اسکور مدل روی داده های اموزشی برابر با 1 بود ولی بعد از هرس کردن 0.97 شد که نشان دهنده جلوگیری از overfitting است.

تسک 13

13- آیا میتوان با استفاده از روش Elbowبا استفاده نموداری مشابه نمودار زیر که نمایان گر بایاس ، واریانس و مرتبه مدل است . بهترین مرتبه مدل برای پیچیدگی مدل را یافت؟ به طور مثال با استفاده از روش elbowمیتوان در نظر گرفت که بر روی دیتاست ، مدلی از مرتبه ۳جواب خوبی به ما میدهد . آیا همواره در تمامی مسائل و نه صرفا بحث تحلیلی میتوان اینگونه قضاوت کرد و مرتبه مناسب را به دست آورد ؟(راهنمایی برای پاسخ به این سوال توجه به مفهوم بایاس میتواند کمک کننده باشد).



شکل ۱: نمودار بایاس و واریانس بنا بر مرتبه های مختلف مدل

به طور کلی وقتی تعداد فیچر های خیلی کم است، بایاس و واریانس بالاست. با افزایش تعداد فیچر ها، بایاس و واریانس کاهش می یابد. از جایی به بعد بایاس کاهش یافته ولی واریانس افزایش می یابد(overfitting اتفاق می‌افتد). نقطه ای که بهترین تریدآف بین بایاس و واریانس وجود دارد، تعداد فیچرهای بهینه مارا نشان میدهد. استفاده از نقطه elbow در نمودار جمع خطای بایاس و واریانش این نقطه بهینه را به ما نشان می دهد.