دانشگاه صنعتی اصفهان

دانشکده مهندسی برق و کامپیوتر

عنوان: تکلیف سوم درس مبانی داده کاوی

نام و نام خانوادگی: افروز شیخ الاسلامی

شماره دانشجویی: ۹۷۲۹۳۹۳

نیم‌سال تحصیلی: زمستان ۱۴۰۰

مدرّس: دکتر ناصر قدیری

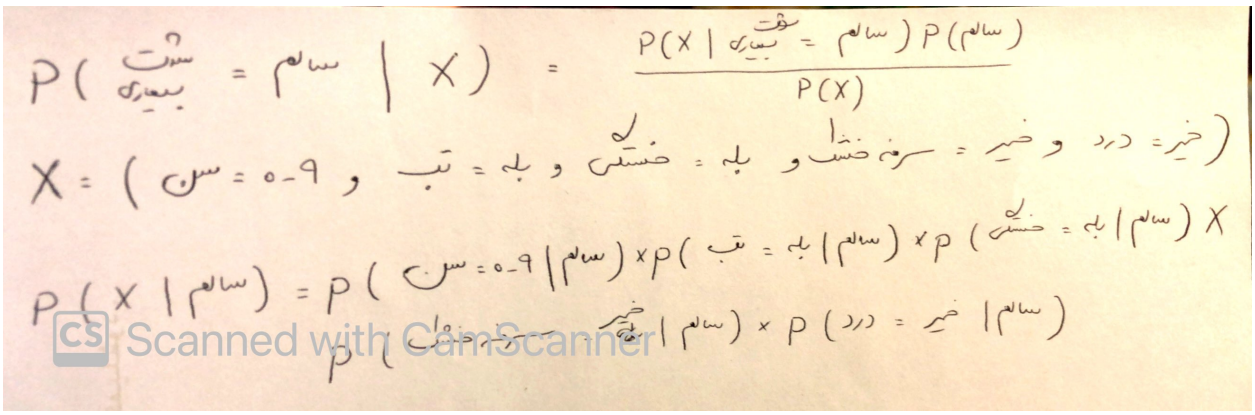
۱ شبکه عصبی

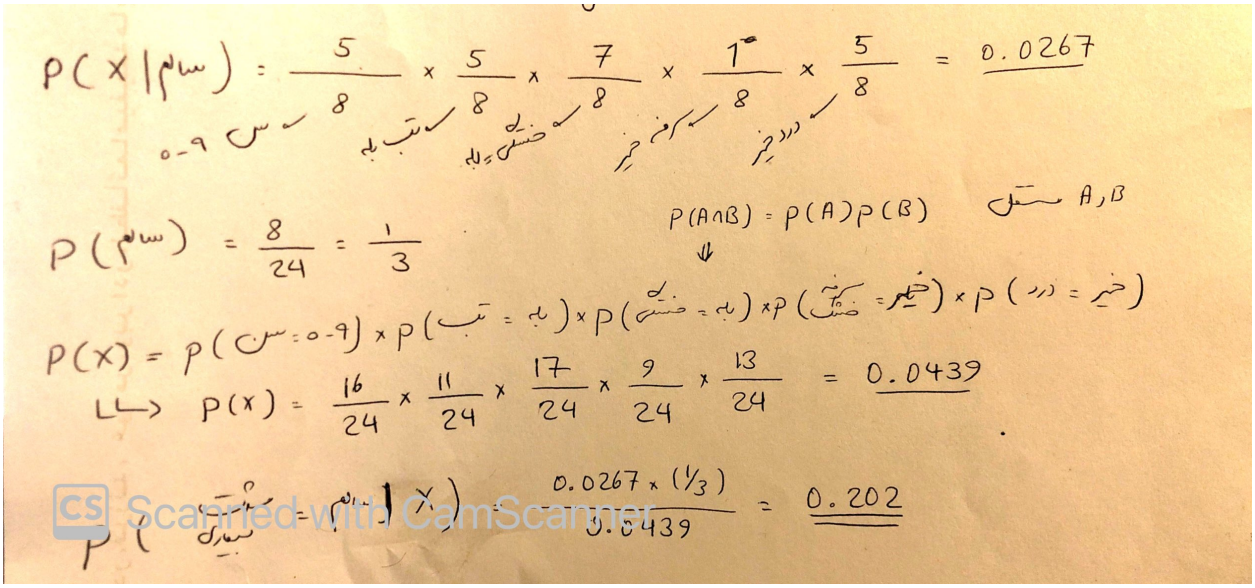
پاسخ های این قسمت به طور کامل در نوتبوک آپلود شده برای این سوال، موجود است.

۲ Feature Selection

پاسخهای این قسمت به طور کامل در نوتبوک آپلود شده برای این سوال، موجود است.

۳ Naive Bayes





۴ سوال چهارم

۱.۴ Preprocessing

پاسخ های این قسمت به طور کامل در نوتبوک آپلود شده برای این سوال، موجود است.

۱.۱.۴ Part F

بیشترین ویژگی هایی که با area همبستگی دارند، عبارتند از: temp, DMC, DC, month-dec, month-sep

۲.۴ Feature Selection and Linear Regression

در این قسمت، تنها بخش هایی که نیاز به توضیح دارد، آورده شده است و کد تمامی بخش ها به صورت کامل در نوتبوک آپلود شده، قرار دارد.

۱.۲.۴ Part K

در مدل مبنا یا Baseline مقدار نتیجه پیش بینی شده، برابر میانگین مقادیر خروجی خواهد بود. بنابراین مدلی که با Linear Regression ساخته می شود، نباید بدتر از مدل مبنا باشد. چرا که برای همه ورودی ها، یک عدد ثابت خروجی داده می شود. در این بخش مشاهده می کنیم، که خطای مدل مبنا، برابر ۱۶.۵ است. در حالی که، خطای مدل Linear Regression برابر ۷۳.۴ شده است. این مقادیر نشان می دهد که مدل Linear Regression نسبت به مدل مبنا کمی بهتر عمل کرده است.

۲.۲.۴ Part L

اکثر ویژگی ها، p-value بیشتر از ۰۵.۰ دارند. و این نشان می دهد که اکثر ویژگی ها، تاثیری در نتیجه نهایی ندارند. ویژگی هایی که p-value کمتری دارند و نسبتا مهم تر هستند، عبارتند از: X, DMC, DC, month_dec, month_mar, month_sep

در قسمت قبل، ویژگی هایی که با area همبستگی بیشتری را داشتند را استخراج کرده بودیم. که ویژگی های زیر بودند: temp, DMC, DC, month-dec, month-sep میبینیم که بیشتر این ویژگی ها، ویژگی هایی هستند که p-value کمتری دارند و نسبتا مهم تر هستند

۳.۲.۴ Part N

در این حالت، خطای مدل Linear Regression نسب به مدل Baseline بیشتر شده است. این مقدار نشان می دهد که مدل به خوبی عمل نکرده است. و به همین علت است که خطای مدل از مدل Dummy Regressor کمتر شده است. البته نتایج این قسمت ها، به طور مستقیم با انتخاب Random-state در train-test-split در رابطه است. و با تغییر Random-state ، نتایج هم تغییر می کنند.

۴.۲.۴ Part O

Ridge Regression همان L2 Regularization است.

In L2 regularization we try to minimize the objective function by adding a penalty term to the sum of the squares of coefficients

مقایسه با Linear Regression

In the linear regression objective function we try to minimize the sum of squares of errors. In ridge regression we add a constraint on the sum of squares of the regression coefficients. Ridge regression objective function:

در مدل بنده، Ridge Regression ، عملکرد بهتری نسبت به Linear Regression نداشته است. این موضوع به علت انتخاب Random-state است. با تغییر Random-state می توان به نتایج متفاوتی دست یافت.

$$\text{Min } (\sum \varepsilon^2 + \lambda \sum \beta^2) = \text{Min } \sum (y - (\beta_1 + \beta_2 X_2 + \beta_3 X_3 + ... + \beta_k X_k))^2 + \lambda \sum \beta^2$$

Part P ۵.۲.۴

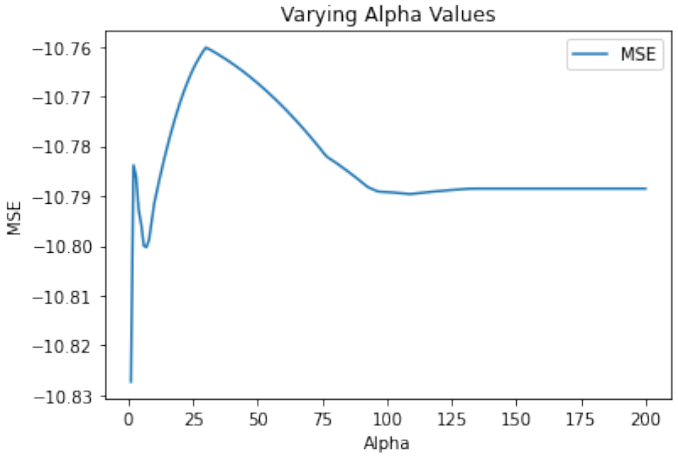
ElasticNet به طور کلی بهتر از ridge و lasso عمل می کند. چراکه ترکیبی از این دو است. و برای مواقعی مناسب است که داده های ما به شدت correlated هستند. در مدل بنده، ElasticNet ، عملکرد بهتری نسبت به Ridge Regression نداشته است. این موضوع به علت انتخاب Random-state است. با تغییر Random-state می توان به نتایج متفاوتی دست یافت.

Part Q ۶.۲.۴

علت این تفاوت این است که Mean Absolute Error قدر مطلق خطاها و Mean Squared Error مربع خطاها را محاسبه می کند. و طبیعی است که مربع خطاها از قدر مطلق آنها بیشتر باشد.

Model Selection ۳.۴

Part S ۱.۳.۴

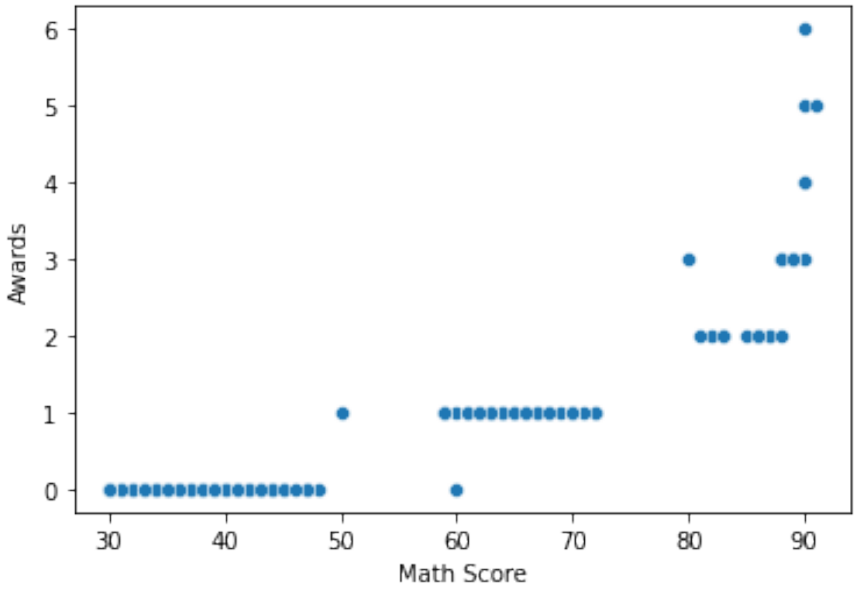


به دلیل اینکه metric ما در این مسئله، neg mean squared error است، بهترین نتیجه برابر نقطه max نمودار پایین است. (در این حالت قدر مطلق MSE مینیمم می شود) یعنی مقدار alpha بهینه برابر ۳۰ است. علت کاهش مدار میانگین MSE بعد از نقطه ماکزیمم: هنگامی که alpha افزایش می یابد، مدل برای جلوگیری از overfitting ، بیشتر نرمال می شود. زیاد شدن بیش از حد alpha باعث می شود مدل به خوبی یادگیری نداشته باشد و این قضیه باعث کاهش کارایی و دقت مدل می شود.

۵ Poisson Regression

در این قسمت، تنها بخش هایی که نیاز به توضیح دارد، آورده شده است و کد تمامی بخش ها به صورت کامل در نوتبوک آپلود شده، قرار دارد.

۱.۵ Part B



این نمودار نشان می دهد که هر چه نمره ریاضی بیشتر باشد، تعداد جوایز دریافت شده بیشتر خواهد بود. برای مثال فردی که نمره ریاضی ۹۰ گرفته است، ۶ جایزه دریافت کرده است. در حالیکه فردی با نمره ریاضی ۴۰ یا ۵۰ ، هیچ جایزه ای دریافت نکرده است. افرادی که هیچ جایزه ای دریافت نکرده اند، اغلب نمراتی بین ۳۰ تا ۵۰ داشته اند. افرادی که ۱ جایزه دریافت کرده اند، اغلب نمراتی بین ۶۰ تا ۷۵ داشته اند. افرادی که دو جایزه یا بیشتر گرفته اند، نمراتشان بیشتر از ۸۰ بوده است. افراد با نمره ریاضی بیشتر از ۸۰، حداقل ۲ جایزه را گرفته اند. و افراد با نمره ریاضی کمتر از ۷۵، حداکثر یک جایزه گرفته اند.

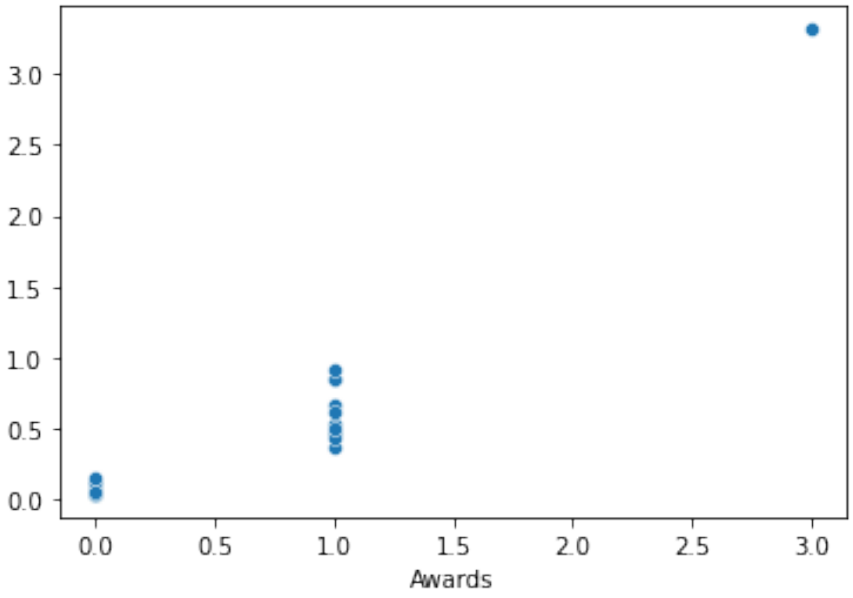
۲.۵ Part E

R2 score داده های تست برابر 0.84 شده است. یعنی ۸۴ درصد تغییرات y یا در اینجا award توسط متغیرهای پیش بین که در اینجا Math Score است، پوشش داده می شود. در واقع معیار R2 score نشان دهنده درصد واریانس یا تغییرات متغیر وابسته یا y است که توسط متغیرهای پیش بین پوشش داده می شوند. برای مثال یک مدل ثابت که همیشه مقدار y را بدون توجه به ویژگی های ورودی، پیش بینی می کند، R2 score صفر خواهد گرفت.

$$R2 = \frac{Variance-explained-by-model}{Total-Variance}$$

Part F ۳.۵

در محور X داده های تست واقعی و در محور Y داده های تست پیش بینی شده دیده می شود. همانطور که مشخص است، داده هایی که برچسب واقعی آنها صفر بوده است، اغلب نزدیک به صفر پیش بینی شده اند. اما داده هایی که برچسب آنها یک بوده است، در بازه بزرگتری پیش بینی شده اند. این بازه از حدود 0.4 تا ۱ است. یک داده تست با برچسب ۳ موجود است که آن حدود 3.3 پیش بینی شده است که بسیار به ۳ نزدیک است.



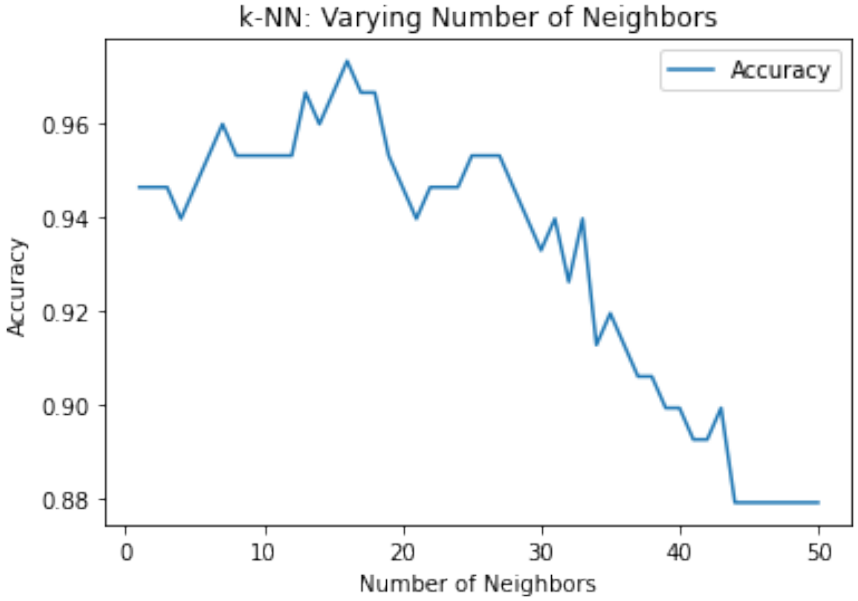
Part G ۴.۵

خیر نمی توانیم از Logistic Regression استفاده کنیم. Logistic Regression برای مواقعی استفاده می شود که متغیر خروجی یا y ، باینری باشد و ما قصد داشته باشیم دو مقدار را پیش بینی کنیم. برای مثال یک شخص بیماری را داشته است یا خیر. اما Poisson Regression در مواقعی استفاده می شود که متغیر وابسته یا y از جنس count یا شمارشی باشد. برای مثال پیش بینی اینکه یک شخص چند بار دچار حمله قلبی شده است. یا تعداد روز های زنده بودن بعد از سکته قلبی. در این دیتاست، متغیر خروجی ما از جنس count یا شمارشی بود و قصد داشتیم تعداد جوایز اهدا شده به یک دانش آموز را پیش بینی کنیم. لذا، Logistic Regression برای این کار مناسب نیست.

KNN ۶

Part B ۱.۶

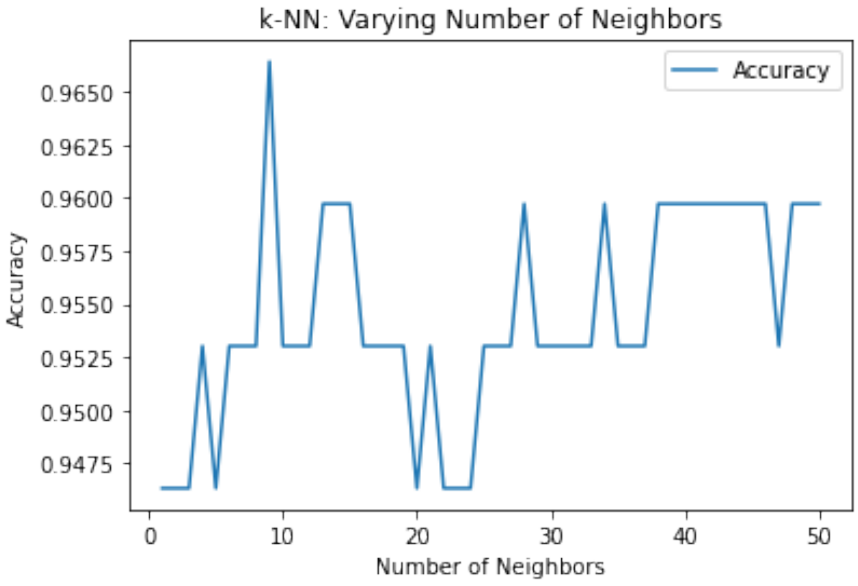
کمترین نرخ خطا برابر 0.02685 ااست.(بیشترین دقت برابر 0.97315 است) که مربوط به k=16 است. این نمودار نشان می دهد که با افزایش k تا مقداری مشخص ، دقت افزایش می یابد. اما بعد از آن مقدار مشخص، هر چه k بیشتر می شود، دقت کاهش می یابد. علت آن، این است که هنگامی که k کوچک است، مدل به تعداد همسایه های کمی برای پیش بینی کلاس، اکتفا می کند و در واقع قدرت تعمیم پذیری مدل کم است. می توان گفت که در این حالت مدل overfit شده است. اما در k های بزرگ ، مدل روی تعداد زیادی از همسایه ها برای پیش بینی اکتفا می کند. این قضیه در واقع تاثیر همسایه های نزدیک را از بین میبرد و مدل توجهی به local ها ندارد. همین قضیه باعث افت دقت آن می شود.



Part C ۲.۶

در این روش کمترین نرخ خطا برابر 0.03356 ااست.(بیشترین دقت برابر 0.96644 است) که مربوط به k=9 است. در این روش، تاثیر راس هر سمپل، با وزن معکوس مجذور فاصله آن سمپل محاسبه می شود. بنابراین مشاهده می کنیم که در k های بزرگ، اتفاق قسمت اول نیوفتاده است. یعنی از آنجا که همسایه های نزدیک یک داده، وزن بیشتری نسبت به همسایه های دورتر دارند، مدل توجه خود را به local ها از دست نمی دهد. و در نتیجه توانسته است دقت بهتری نسبت به حالت اول کسب کند.

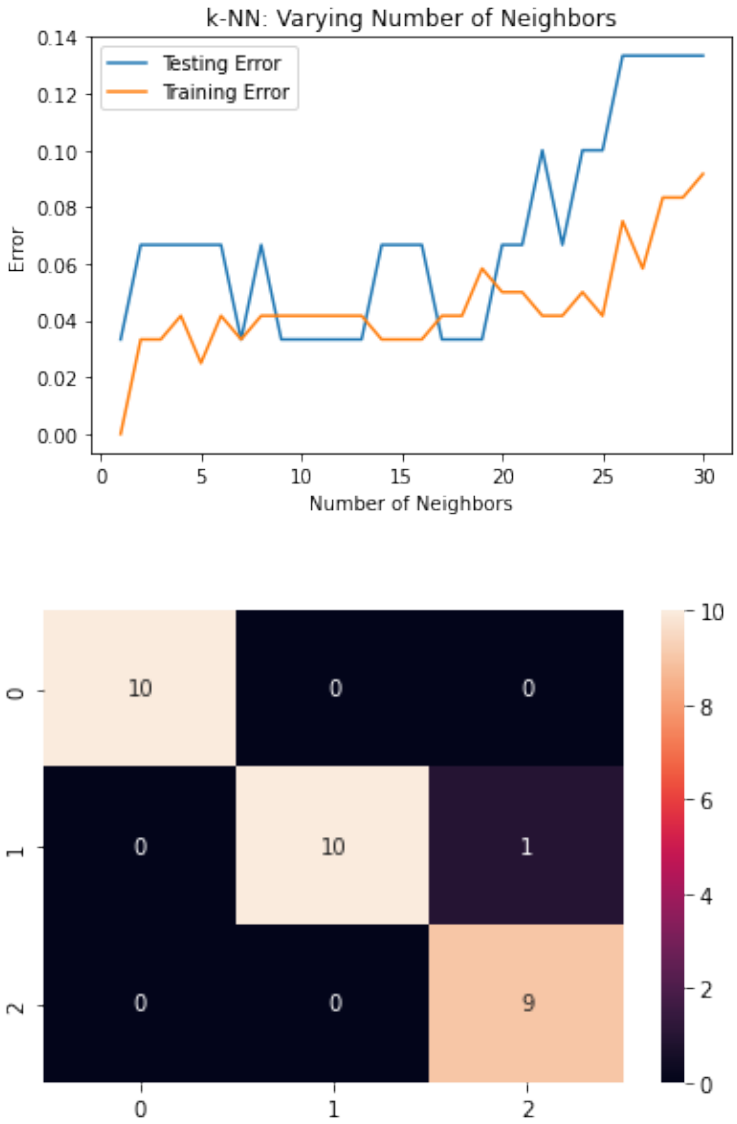
اما در k های کوچک، داستان به همان صورت است. چرا که مدل قدرت تعمیم پذیری ندارد.


k-NN: Varying Number of Neighbors
Accuracy
0.9650
0.9625
0.9600
0.9575
0.9550
0.9525
0.9500
0.9475
Accuracy
0
10
20
30
40
50
Number of Neighbors

۷ سوال هفتم

۱.۷ Part A

همانطور که از نمودار پایین مشخص است، مقدار K بهینه را ۷ انتخاب کرده ام. با اینکه برای K=1 مقدار خطای کمتری در training بود، اما overfit شده بود و توانایی تعمیم پذیری مدل کم بود. البته ممکن است با random-state های متفاوت هنگام split کردن داده ها، مقدار K بهینه تغییر کند.



شکل ۱: نمودار Confusion Matrix برای K بهینه.

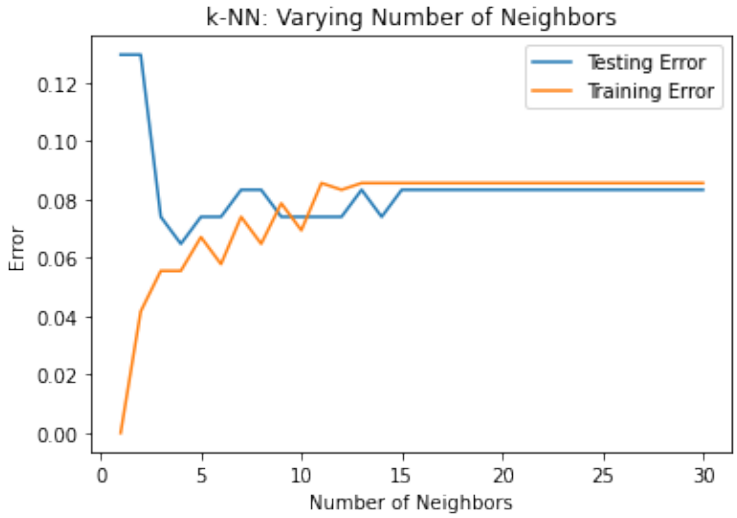
با توجه به نمودار بالا، مدل ما تمامی رکورد های کلاس ۰ و ۱ را به درستی تشخیص داده است. اما یکی از سمپل های کلاس ۲ را به اشتباه، در کلاس ۱ تشخیص داده است.

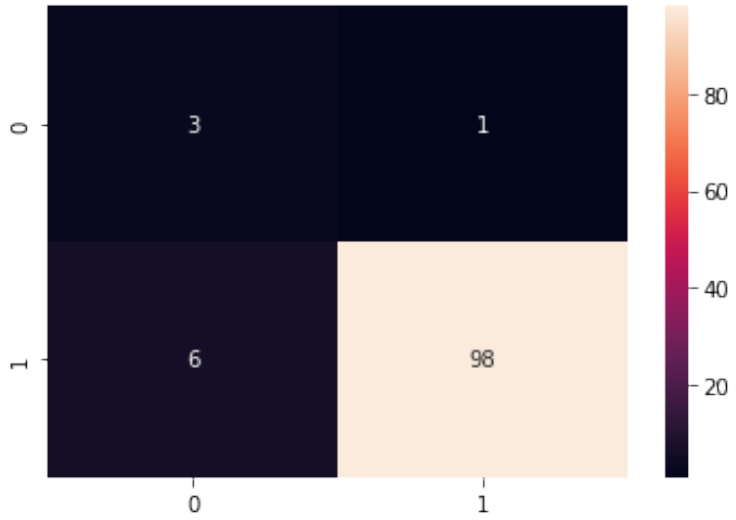
Part B ۲.۷

نتایج الگوریتم BallTree دقیقا مانند حالت اول شد. به تحلیل های قسمت اول مراجعه کنید.

Part C ۳.۷

نمودار پایین، نشان دهنده نتایج خطا برای K=1 تا ۳۱ با الگوریتم KDTree است. نتایج برای الگوریتم BallTree هم دقیقا مانند KDTree بود. لذا نمودار های پایین نشان دهنده نتایج هر دو این الگوریتم ها هستند. مقدار K بهینه را با توجه به نمودار پایین، ۴ انتخاب کرده ام.

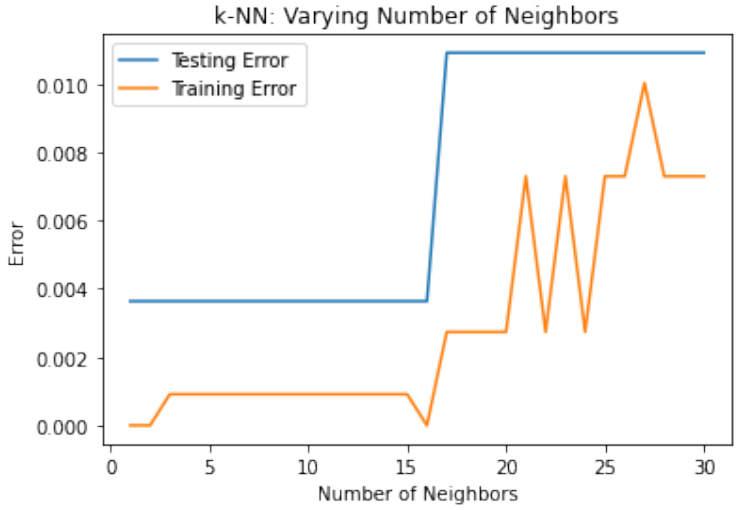


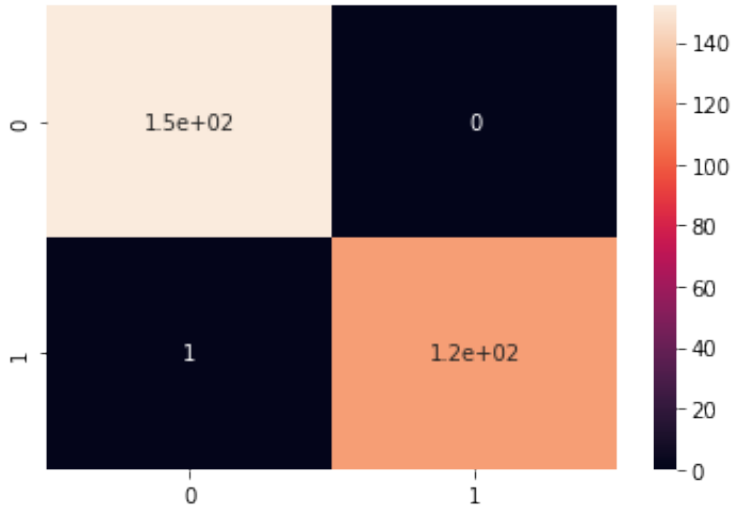


با توجه به Confusion Matrix برای K بهینه، متوجه می شویم که از ۹۹ رکورد موجود در کلاس ۱، ۹۸ رکورد از آنها به درستی تشخیص داده شده اند و تنها یکی در کلاس صفر پیش بینی شده است. همچنین از ۹ رکورد موجود در کلاس صفر، ۳ تا از آنها به اشتباه پیش بینی شده اند.

Part D ۴.۷

نمودار پایین، نشان دهنده نتایج خطا برای K=1 تا ۳۱ روی دیتاست $KDTreeBanknote_{a}uthentication$ است. نتایج برای الگوریتم BallTree هم دقیقا مانند KDTree بود. لذا نمودار های پایین نشان دهنده نتایج هر دو این الگوریتم ها هستند. مقدار K بهینه را با توجه به نمودار پایین، ۱۶ انتخاب کرده ام.





با توجه به Confusion Matrix برای K بهینه، متو جه می شویم که تمامی ۱۲۲ رکورد موجود در کلاس ۱ ، به درستی پیش بینی شده اند. و از ۱۵۳ رکورد موجود در کلاس صفر، تنها یکی به اشتباه تشخیص داده شده است.

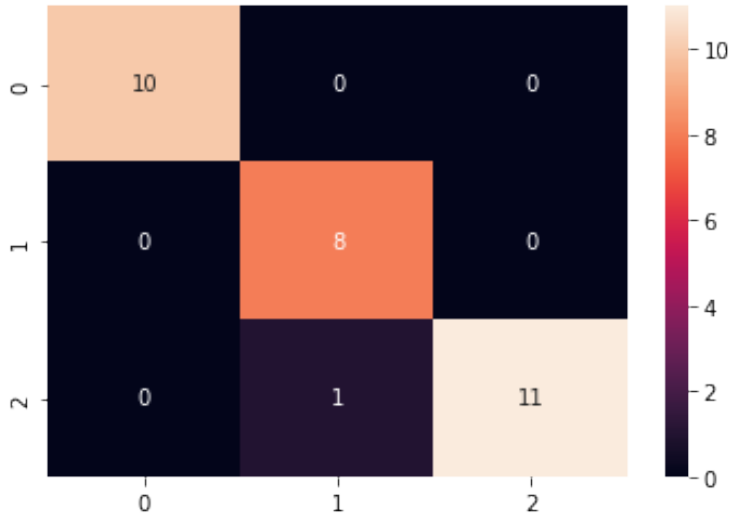
Part E ۵.۷

مانند قسمت های قبل نتایج هردو یکسان بود. و مقدار K بهینه را ۹ انتخاب کرده ام اجرای الگوریتم KDTree حدود ۱۲ دقیقه طول کشید. در حالیکه الکوریتم BallTree بسیار زمان بر تر بود.

Part F ۶.۷

پیچیدگی زمانی الگوریتم KDTree از مرتبه (O(log(n)) است. در حالیکه پیچیدگی زمانی الگوریتم BallTree از مرتبه (O(nlog(n)) است. لذا بهتر است از الگوریتم BallTree برای مجموعه دادگانی که تعداد رکورد کمی دارند استفاده کرد و برعکس برای مجموعه دادگان بزرگ از KDTree استفاده کرد. اما در مورد تعداد ویژگی های موجود در مجموعه داده، دقیقا عکس مطلب گفته شده برقرار است. یعنی بهتر است برای مجموعه دادگان با ویژگی های زیاد از BallTree و برای دادگانی با ویژگی های کمتر از KDTree استفاده کنیم. درستی مطالب گفته شده در قسمت E که حجم داده ها بیشتر از بقیه قسمت ها بود، اثبات شد.

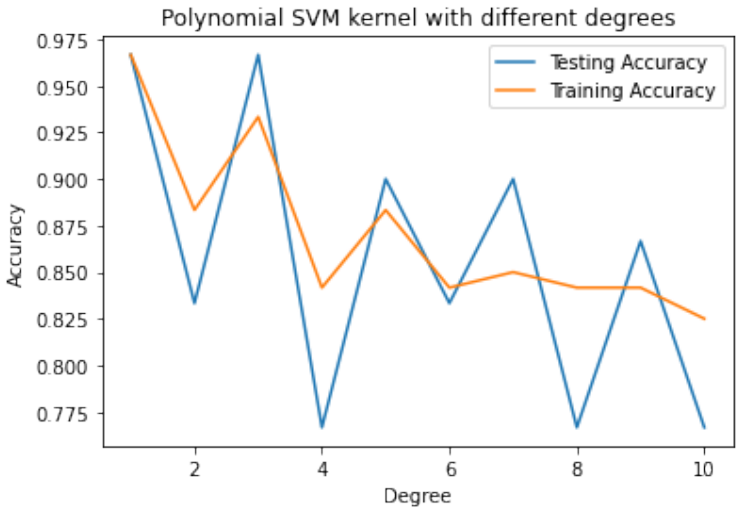
۸ SVM



همانطور که مشخص است، ۱۰ سمپل در کلاس ۰ موجود است که همگی به درستی تشخیص داده شده اند. همچنین از ۹ سمپل موجود در کلاس ۱، ۸ مورد آنها به درستی و ۱ مورد در کلاس ۲ پیش بینی شده اند. همگی دیتا های کلاس ۲ ، به درستی پیش بینی شده اند.

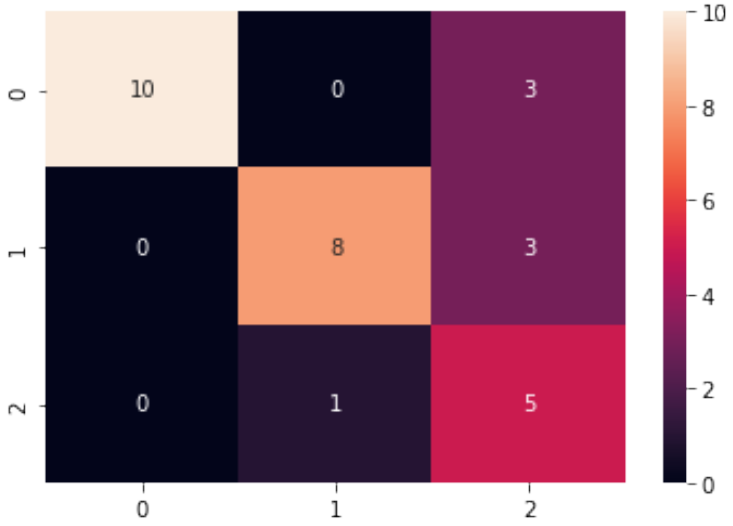
۱.۸ Part A

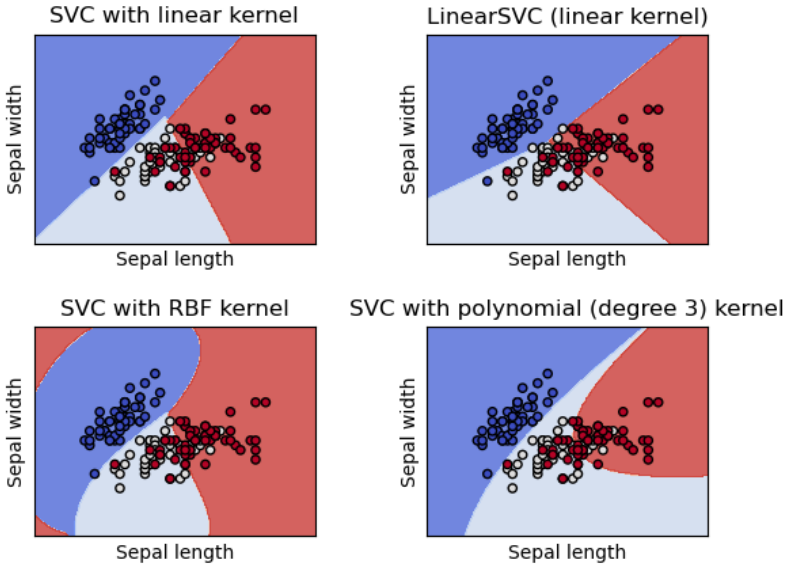
به طور کلی می توان نتیجه گرفت که هرچه درجه polynomial افزایش می یابد، دقت کاهش یافته و خطا چه روی train و چه test افزایش می یابد. اما به طور دقیق تر، در درجات ۴ و ۸، بیشترین میزان خطا و کمترین میزان دقت را داریم. همچنین بیشترین دقت مربوط به درجه ۱ است.



Part B ۲.۸

به طور کلی، دقت از قسمت اول کمتر شده است. در مورد کلاس های ۰ و ۱، مدل این قسمت مانند مدل قسمت اول عمل کرده است. اما در مورد کلاس ۲ ، مدل اول توانسته بود، تمامی ۱۱ داده موجود در این کلاس را به درستی تشخیص دهد. در حالیکه مدل این قسمت با کرنل polynomial ، تنها توانسته ۵ تا از ۱۱ داده در کلاس ۲ را به درستی تشخیص دهد و ۳ داده را در کلاس ۰ و ۳ تای دیگر را در کلاس ۱ پیش بینی کرده است. با توجه به شکل پایین علت این مسئله مشخص می شود.





با توجه به شکل داده ها، مشخص است که SVM خطی می تواند بهتر از SVM با کرنل polynomial عمل کند. SVM خطی توانسته است داده های کلاس ۲ را بهتر از SVM با کرنل polynomial جدا کند. همین موضوع باعث شده است که مدل قسمت اول، دقت بهتری نسبت به مدل قسمت سوم داشته باشد.

۹ کاهش ابعاد

پاسخ های این قسمت به طور کامل در نوتبوک آپلود شده برای این سوال، موجود است.

۱۰ AdaBoost

پاسخ های این قسمت به طور کامل در نوتبوک آپلود شده برای این سوال، موجود است.

۱۱ XGBoost

پاسخ های این قسمت به طور کامل در نوتبوک آپلود شده برای این سوال، موجود است.

۱۲ Stacking Ensemble

۱.۱۲ Part B

دقت مدل stacking برابر 0.953 شده است. و دقت مدل های قسمت اول به شرح زیر است:

knn: 0.92, svm: 0.924, bayes: 0.951

همچنین دقت در مقایسه با قسمت c سوال هفتم، حدود 1.5 درصد افزایش داشته است. نتایج به دست آمده نشان می دهد که در این مجموعه داده، استفاده از ensemble learning با روش stack کردن مدل های متفاوت، کارآمدتر بوده است.