غزل رفیعی | شماره دانشجویی: ۹۷۲۲۲۰۴۴ | تاریخ: ۱۷/ ۱۰/ ۱۴۰۰

گزارش تمرین سری ۲ | درس: مبانی علوم داده | استاد درس: دکتر خردپیشه

مقدمه:

هدف از این تمرین تحلیل و بررسی داده‌های کلیک روی تبلیغات بازی روی پلتفرم‌های متفاوت و پیش‌بینی کلیک با توجه به داده‌های موجود است. در ادامه، با استفاده از ۶ الگوریتم معروف یادگیری ماشین داده‌ها را بر حسب کلیک شدن یا نشدن تقسیم‌بندی می‌کنیم و روش‌های مختلف را مقایسه می‌کنیم.

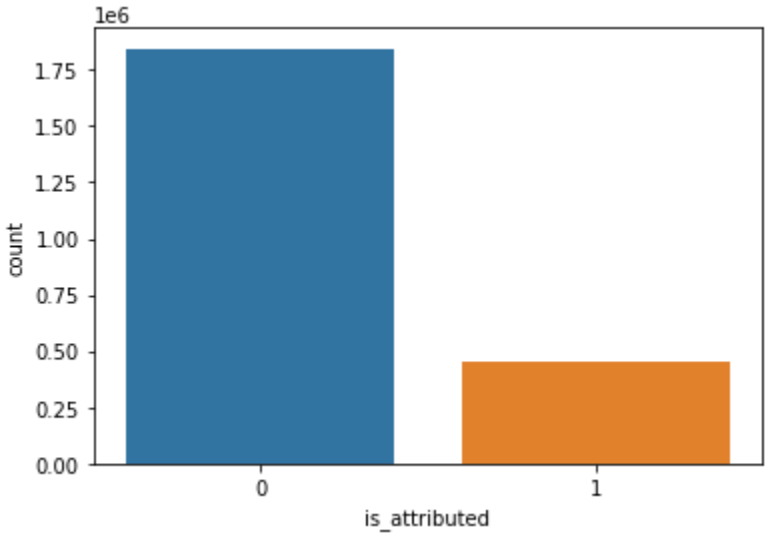
پیش‌پردازش:

این دادگان شامل ۲ میلیون و ۳۰۰ هزار ردیف و ۷ ستون است. این ستون‌ها عبارتند از:

Ip, app, device, os, channel, click_time, attributed_time, is_attributed

همچنین به جز ستون attributed_time این دادگان دارای مقدار پوچ دیگری نیست. در این ستون تنها رکوردهایی که مربوط به کلیک شده هستند دارای مقدار هستند.

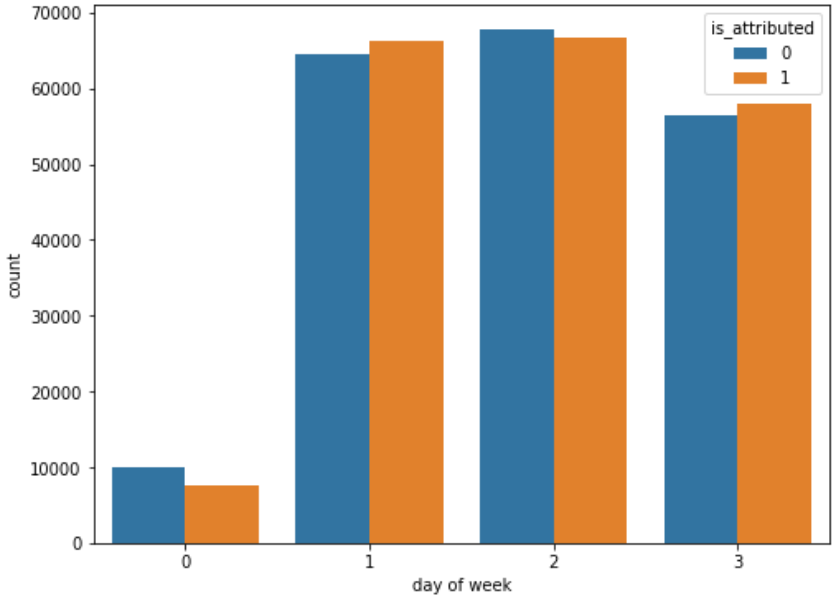
مصورسازی:



شکل ۱: توزیع ستون کلیک شده و نشده

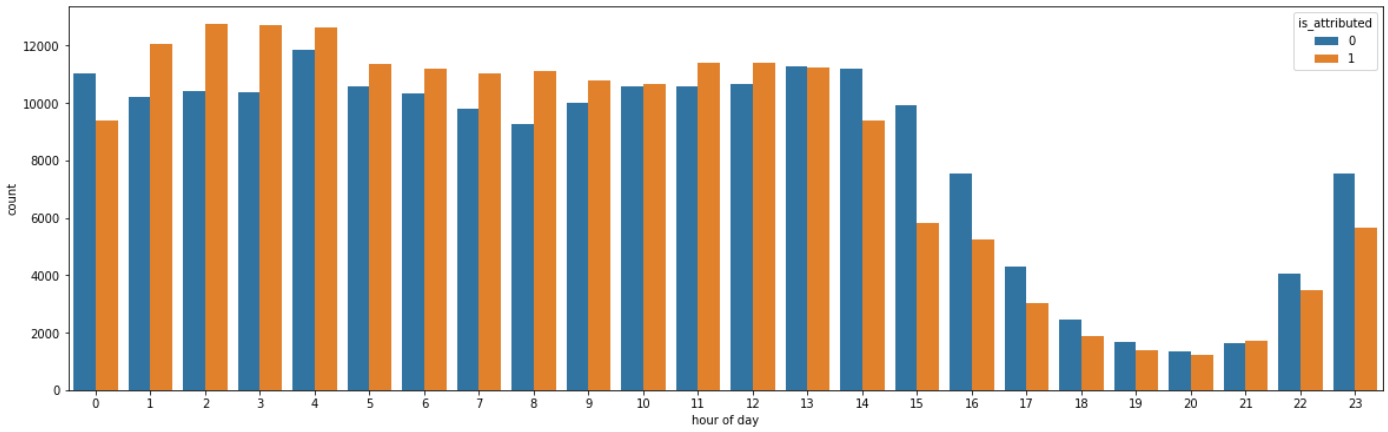
همان طور که در شکل ۱ مشخص است، در این دادگان توزیع دادگان کلیک شده و نشده برابر نیست. بنا براین تعدادی از داده‌های کلیک نشده را حذف می‌کنیم تا اندازه‌ی هر دو کلاس برابر شود.

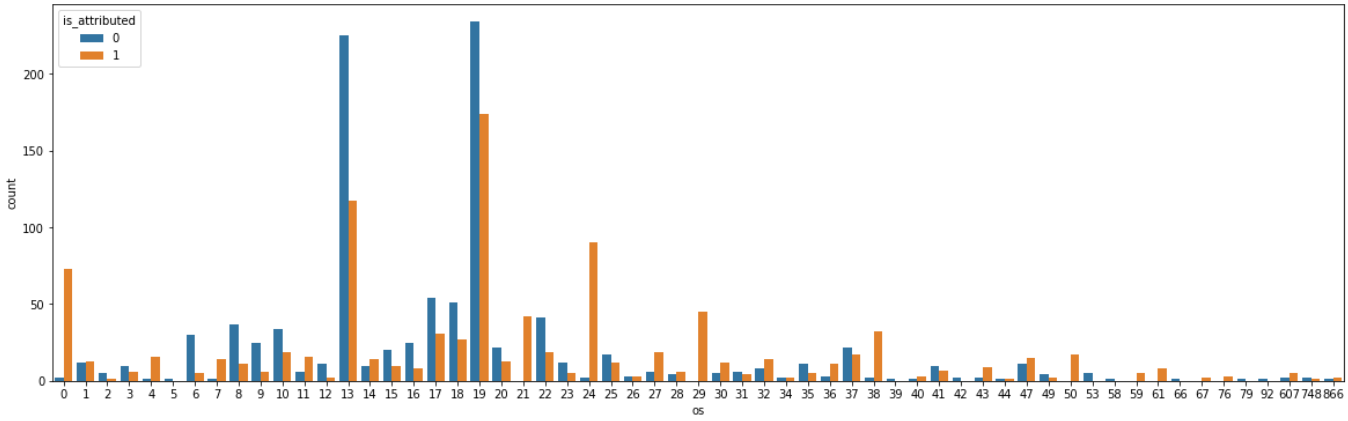
این داده‌ها مربوط به تاریخ 09-11-2017 16:00:00 تا 06-11-2017 15:13:23 هستند که برابر ۳ روز و ۱۵ ساعت است.

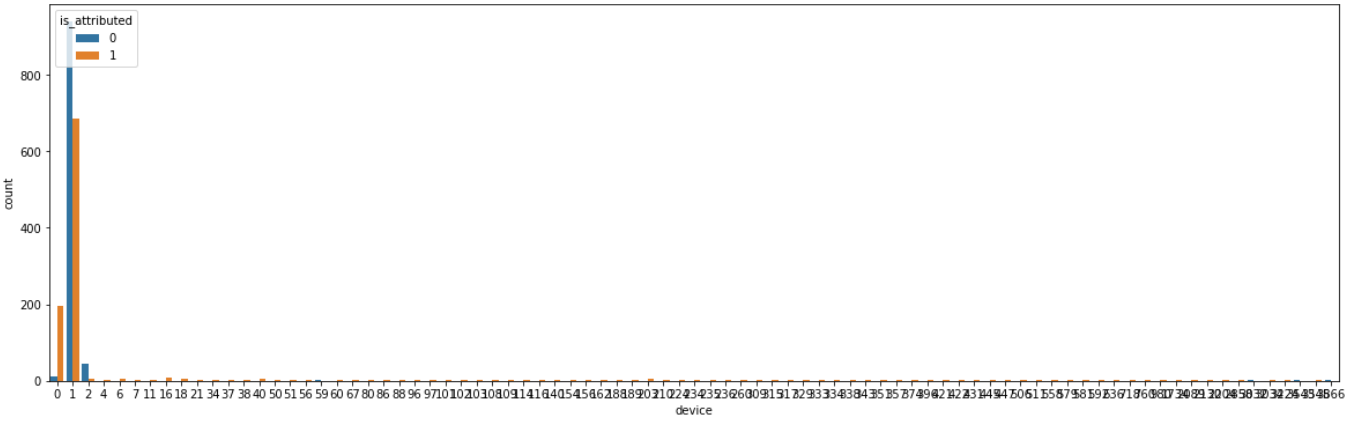


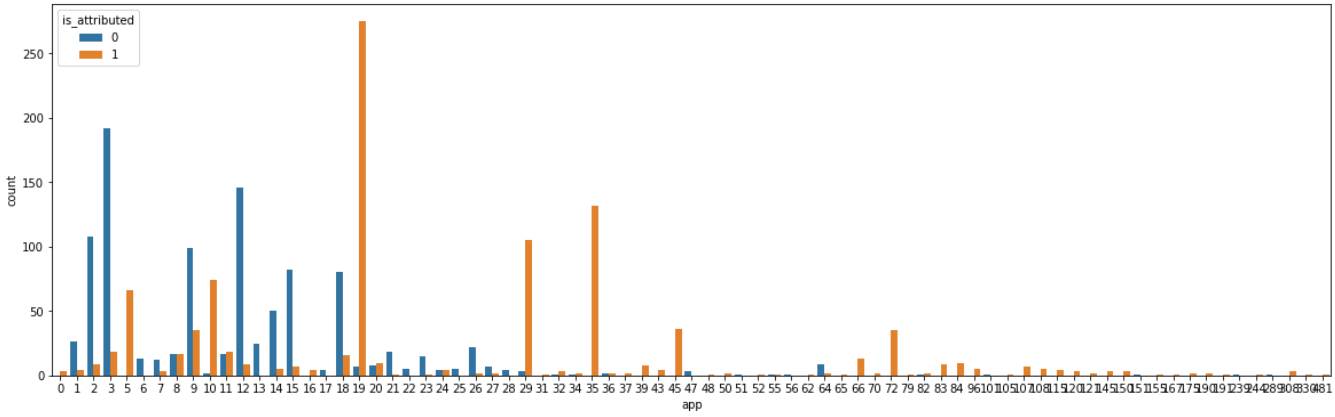
شکل ۲: توزیع ستون کلیک شده و نشده بر حسب روز هفته

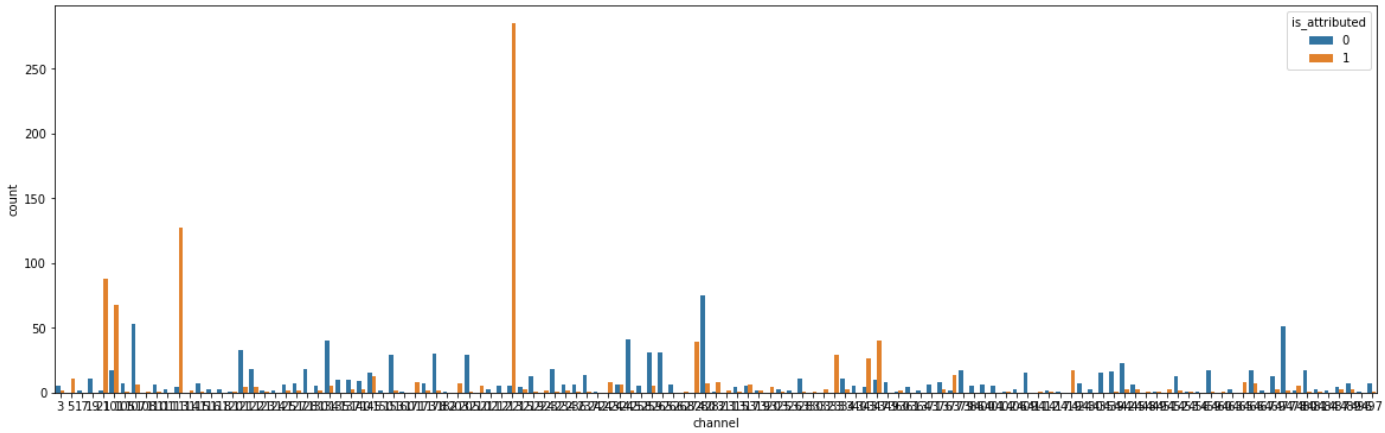
در این شکل ۲ تعداد رکوردهای دو کلاس را در روزهای مختلف هفته مشاهده می‌کنیم. و در ادامه، این اطلاعات را برای ساعت روز، سیستم عامل، دستگاه، اپلیکیشن و کانال مشاهده می‌کنیم.




is_attributed
0
1
count
os


is_attributed
0
1
count
device


is_attributed
0
1
count
app



شکل ۳

از آن جا که مقادیر ستون‌ها جز ستون زمان معنای عددی واقعی ندارند، آن ها را به صورت categorical در می‌آوریم و استاندارد نمی‌کنیم. تنها ستون زمان را بر حسب int در آورده و آن را نرمال می‌کنیم.

مدل‌ها:

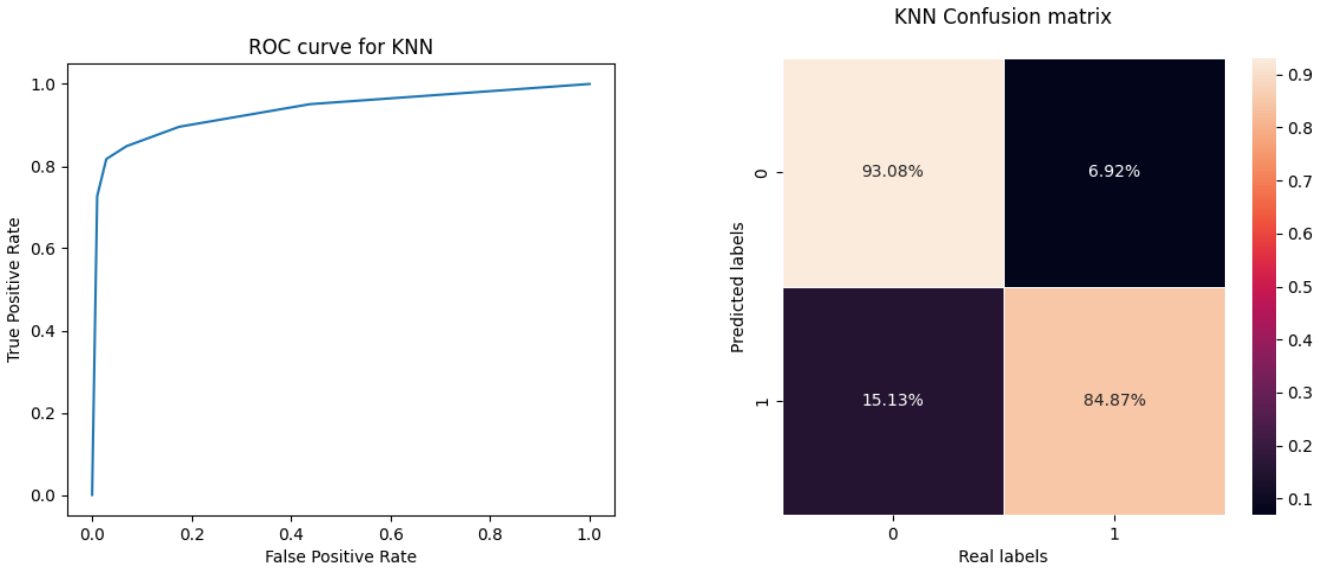
در این مرحله ابتدا فاصله‌ی بین دو نقطه از این داده را این گونه تعریف می‌کنیم:
(شبه کد!)

```
dist = 0
for c in columns:
     if c == ‘click_time’
          dist += abs(row1.c - row2.c)
     else
          if row1.c != row2.c
               dist+=1
```

به این معنا که برای فاصله‌ی دو نقطه برابر تعداد ستون‌هایی‌ست که مقدار آن ها برابر نیست، به علاوه اختلاف زمانی استاندارد شده، که بین ۰ و ۱ است.

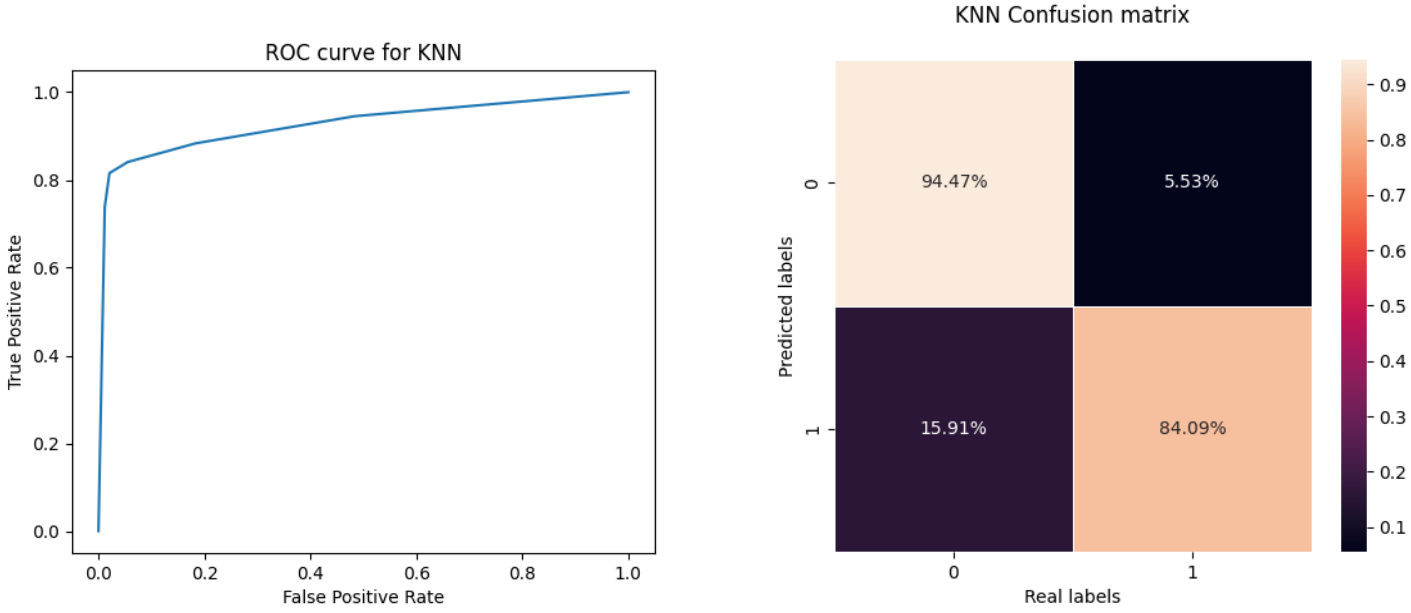
حال به ترتیب الگوریتم‌های کلاس‌بندی Naive Bayes, SVM, Random Forest, Logistic Regression, KNN را روی این دادگان اعمال می‌کنیم و نتایج را در قالب accuracy, confusion matrix, ROC curve و decision boundary نشان می‌دهیم.
همچنین هر الگوریتم ۲ بار برای یک نمونه‌ی تصادفی ۵۰۰۰ تایی اجرا شده:
۱ بار برای کل دادگان
و یک بار برای داده‌هایی تنها ستون app و click time

نتایج KNN:



شکل ۴

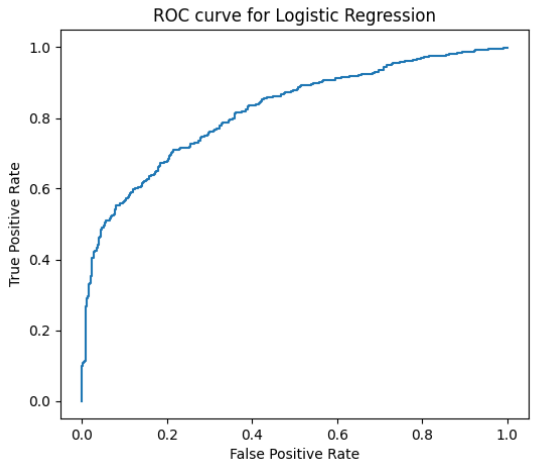
داده‌ی کاهش یافته:

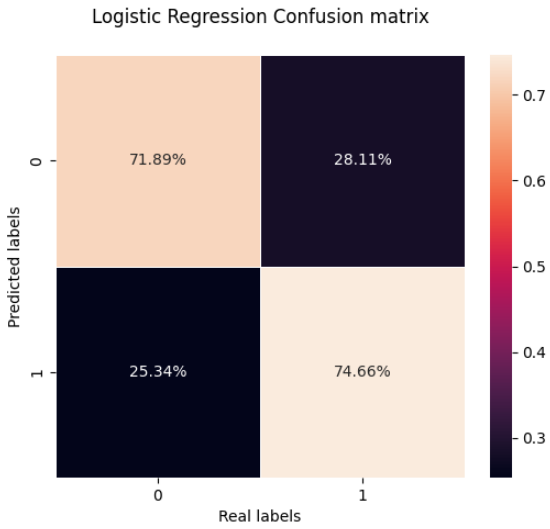


شکل ۵

در این الگوریتم نتایج بین دو آزمایش تفاوت زیادی نکرده و مدل، به ترتیب برای برچسب‌های ۰ و ۱ ۹۴ و ۸۴ درصد و با اختلاف کمی نسبت به حالت همه‌ی ستون‌ها عمل کرده است. که برای ۰ بهتر است.

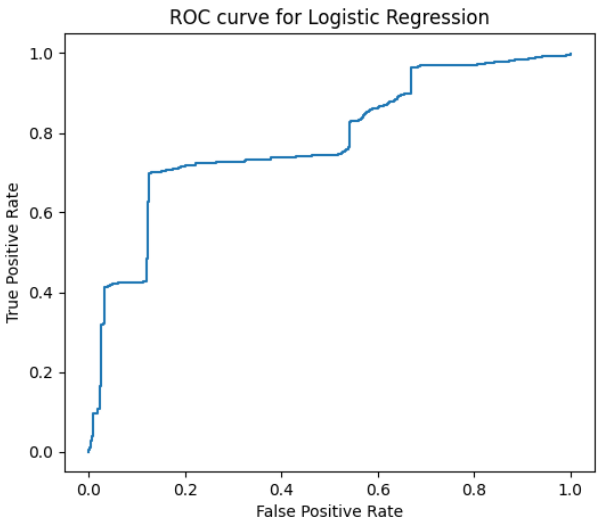
نتایج logistic regression:

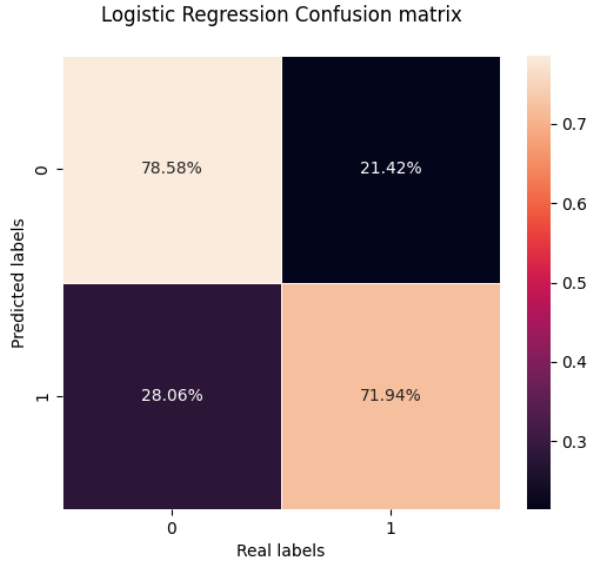


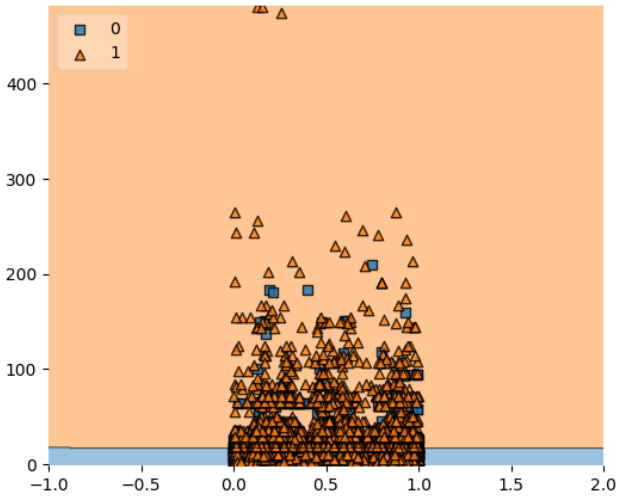


شکل ۶

داده‌ی کاهش یافته:



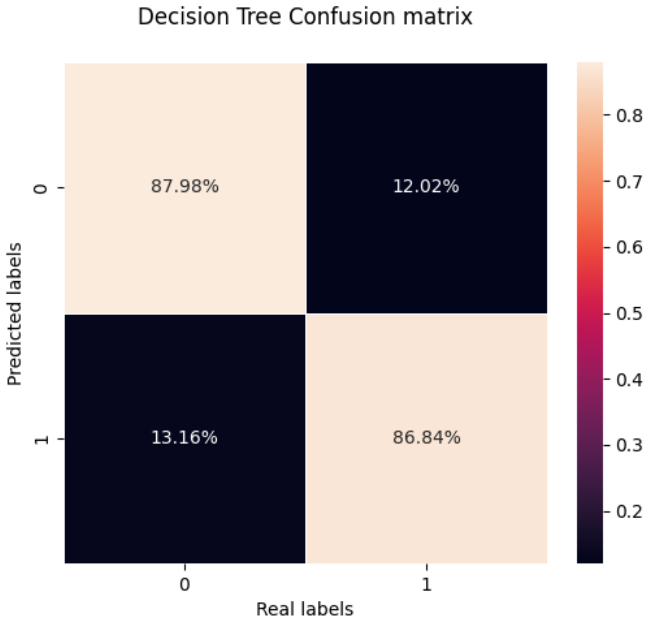


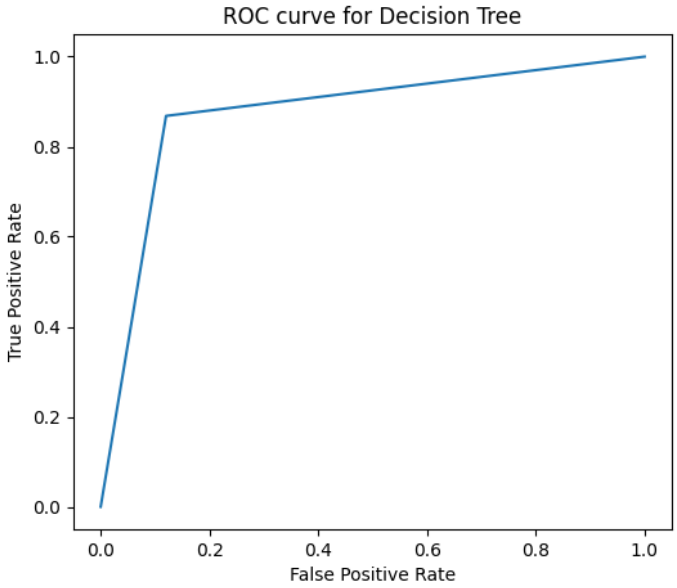


شکل ۷

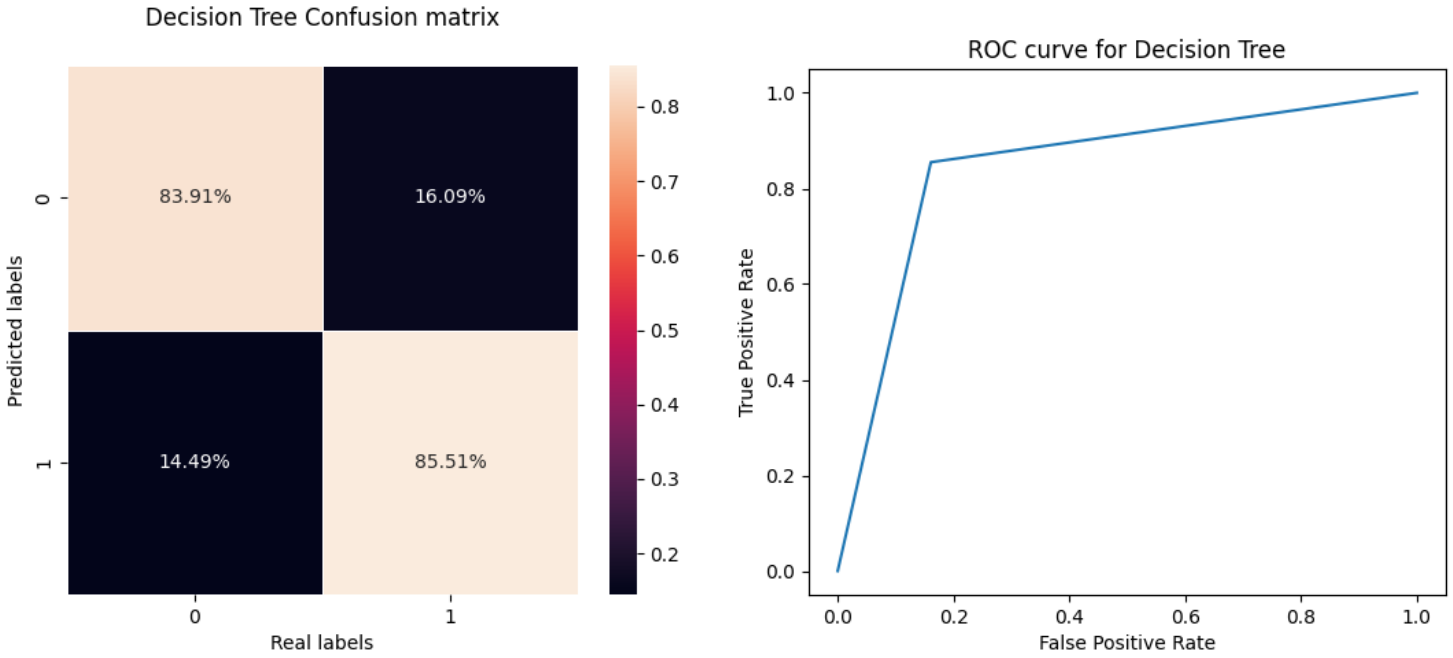
این الگوریتم برای هر دو حالت آزمایش، در حدود ۷۱ و ۷۸ درصد برای برچسب‌های ۰ و ۱ عمل کرده است. هم‌چنین در شکل ۷ می‌توانیم مرز جداسازی نقاط را در فضای app و time ببینیم.

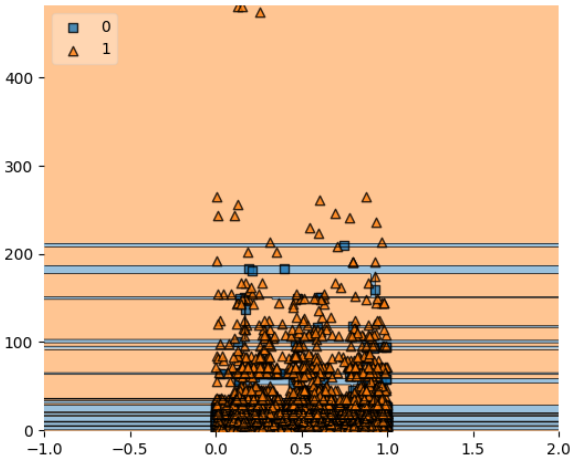
نتایج درخت تصمیم:





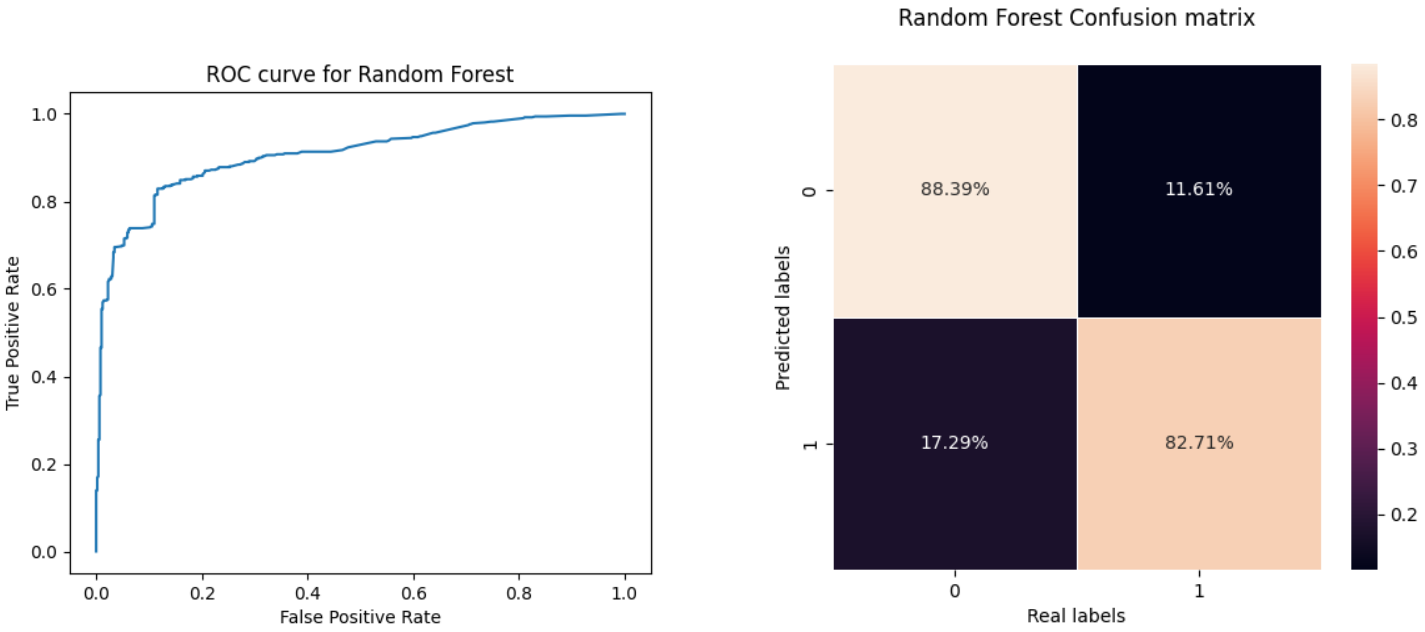
داده‌ی کاهش یافته:





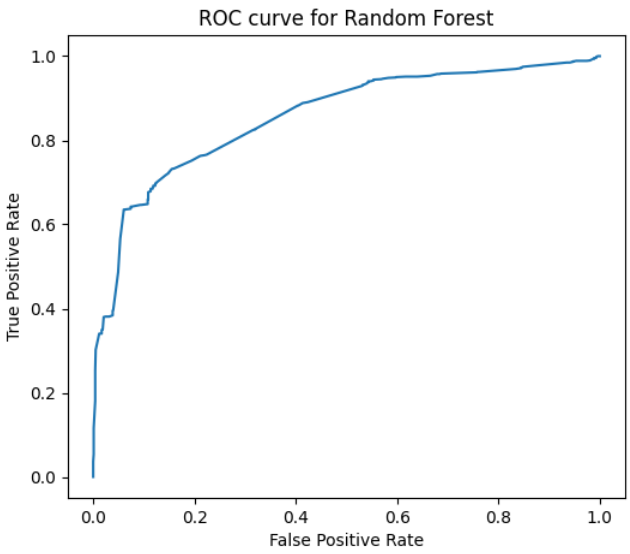
شکل ۸

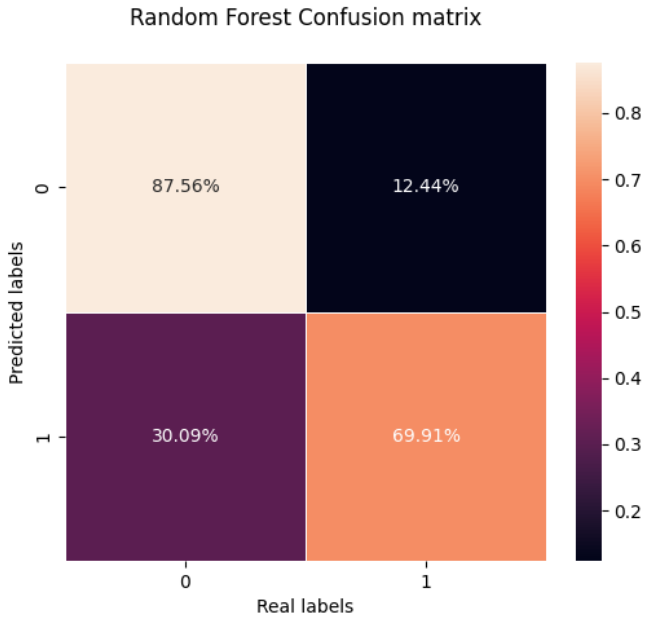
نتایج Random Forest:



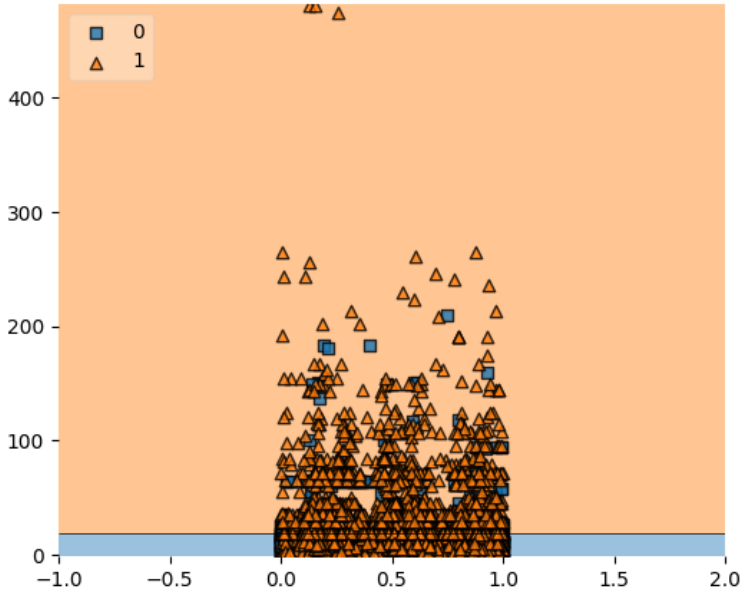
شکل ۹

داده‌ی کاهش یافته:


ROC curve for Random Forest
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

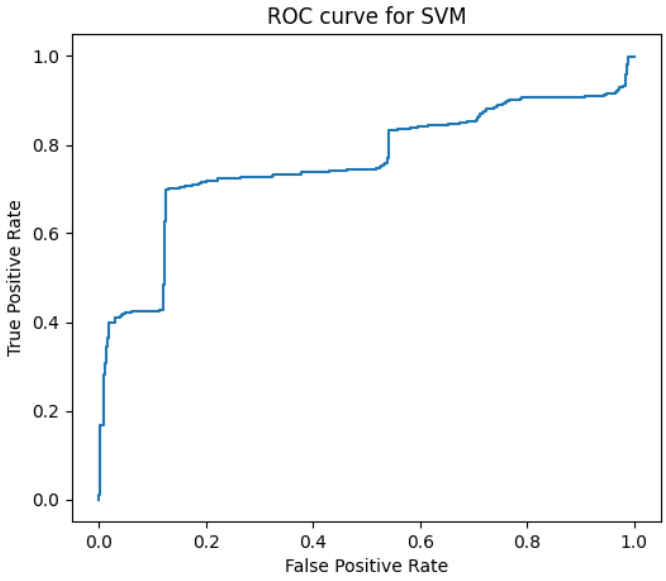

Random Forest Confusion matrix
87.56%
12.44%
30.09%
69.91%
Predicted labels
Real labels
0
1

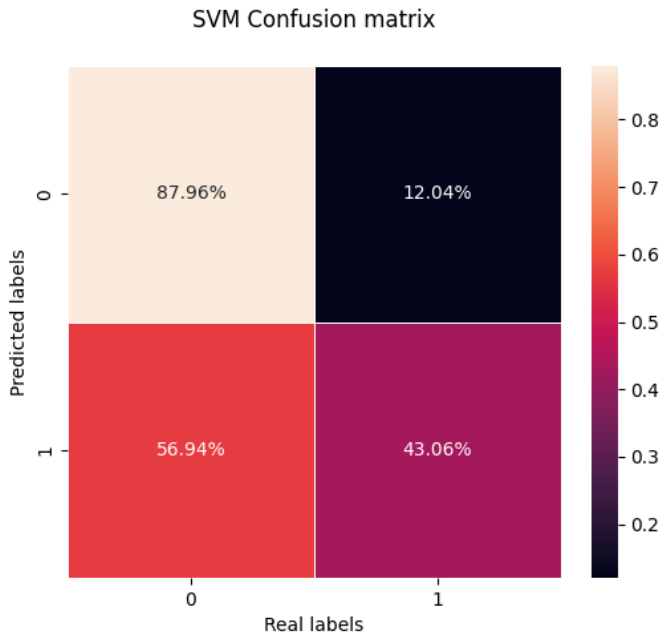
:


0
1
400
300
200
100
0
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

شکل ۱۰

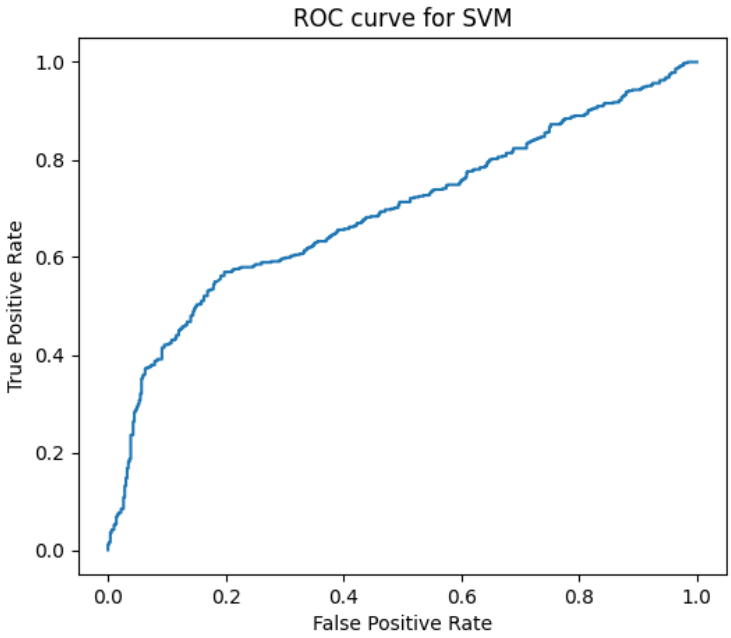
نتایج SVM:

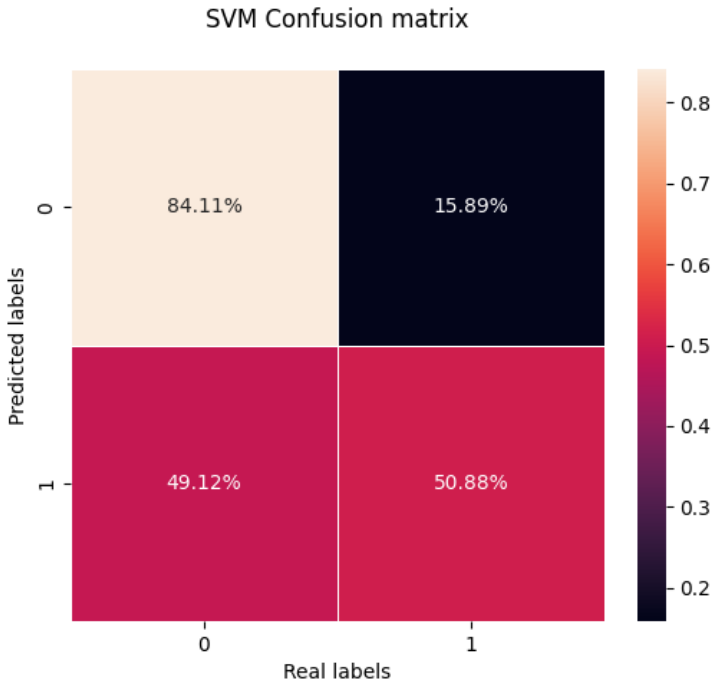


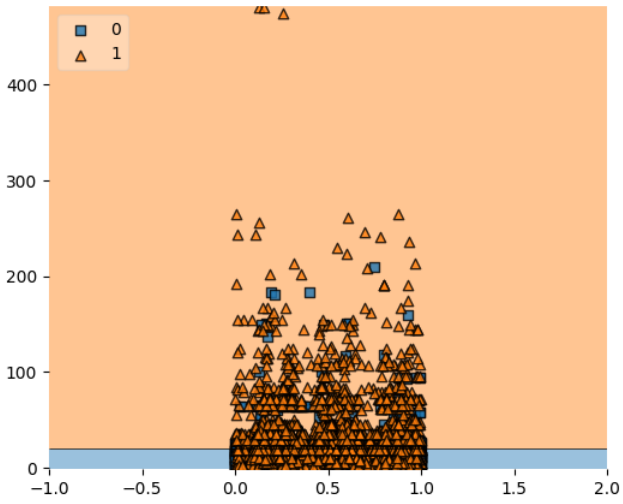


شکل ۱۱

داده‌ی کاهش یافته:

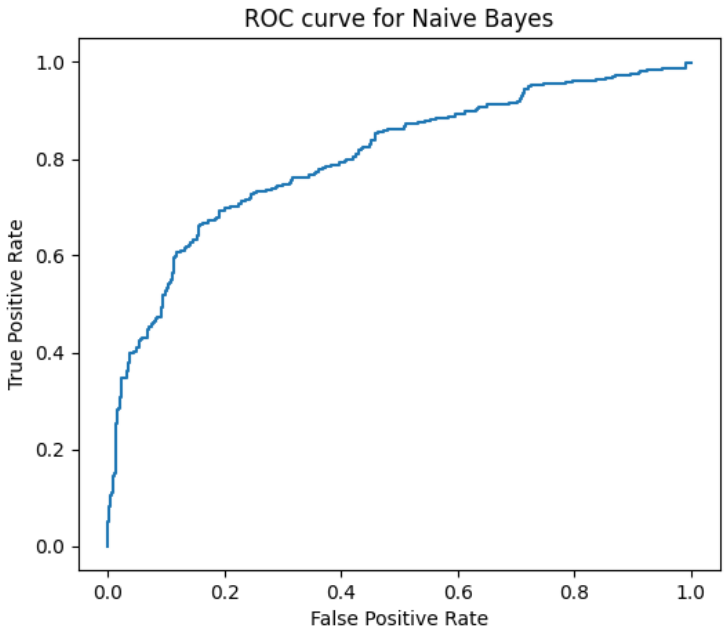

ROC curve for SVM
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

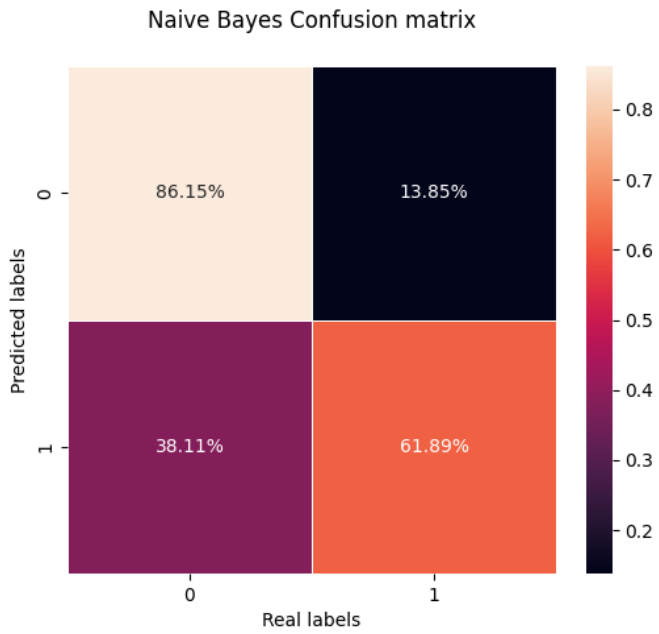

SVM Confusion matrix
84.11%
15.89%
49.12%
50.88%
Predicted labels
Real labels
0
1


0
1
400
300
200
100
0
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

شکل ۱۲

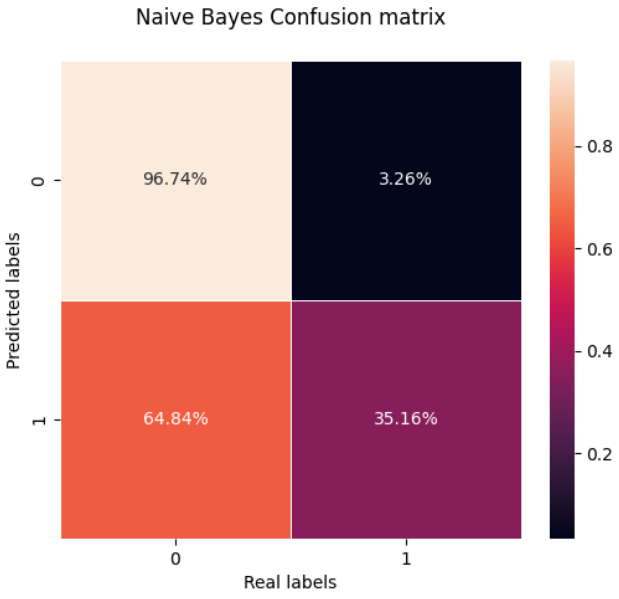
نتایج Naive Bayes:

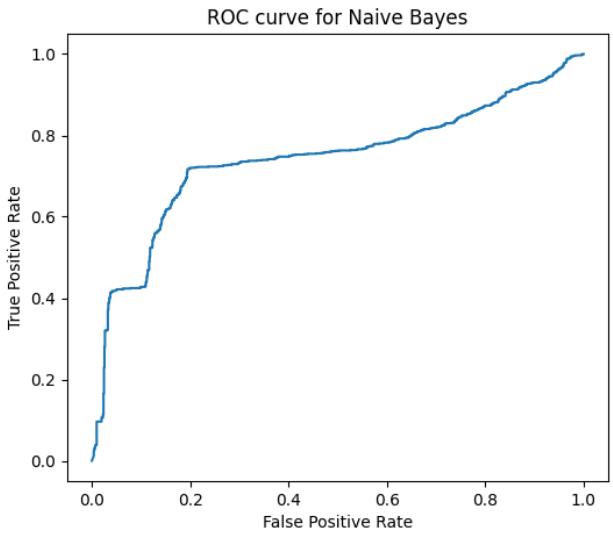


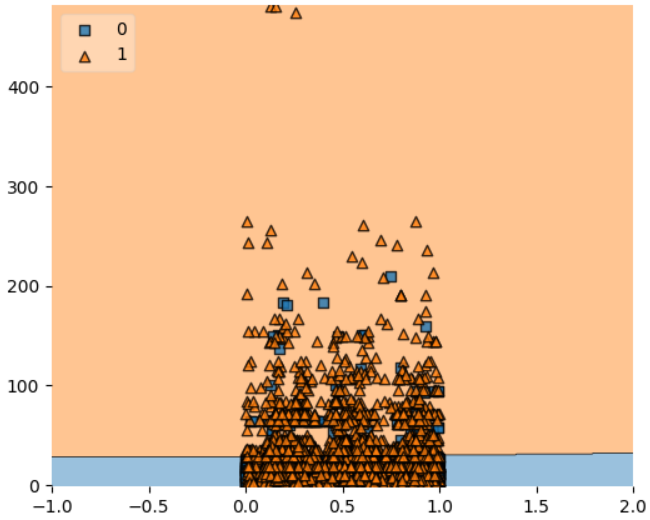


شکل ۱۳

با داده‌ی کاهش یافته:







شکل ۱۴

نتیجه‌گیری:

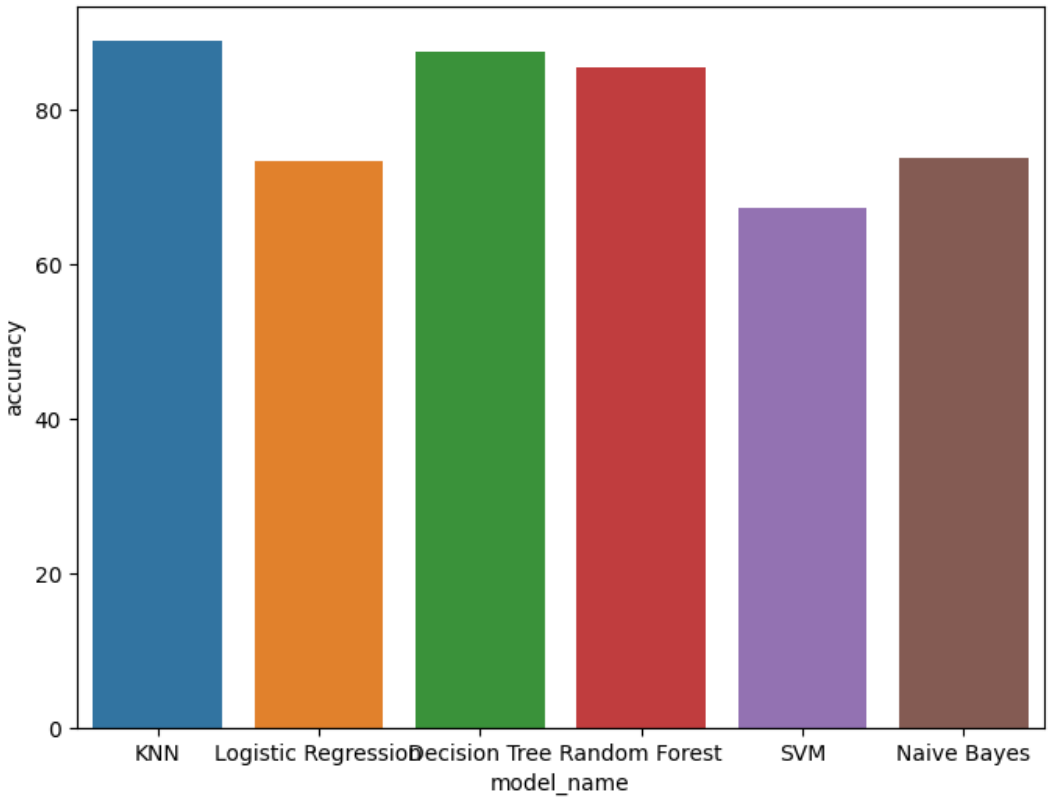
ردیف	الگوریتم	درصد درستی	زمان اجرا	امتیاز مدل (acc/sqrt(time))
0	KNN	88.9	0.088	299.248
1	Logistic Regression	73.3	0.048	335.921
2	Decision Tree	87.4	0.023	581.062
3	Random Forest	85.5	0.188	197.432
4	SVM	67.2	1.827	49.722
5	Naive Bayes	73.8	0.003	1362.211

جدول ۱: نتایج مدل‌ها با همه‌ی ستون‌ها

همانطور که مشاهده می‌کنید در جدول بیشترین درصد درستی مربوط به مدل knn و کمترین مربوط به SVM است. همچنین زمانی که الگوریتم Naive Bayes صرف می‌کند، حدود دو صدم زمانی است که SVM صرف می‌کند. در این جدول ستون پنجمی به نام امتیاز اضافه شده که به صورت زیر به دست می‌آید:

$$Score = \frac{Accuracy}{\sqrt{(time)}}$$

علت اینکه از زمان ریشه دوم گرفته شده این است که تاثیر درصد درستی بیشتر شود.
با این حساب، الگوریتم naive bayes بیشترین امتیاز را کسب کرده است.



شکل ۱۵

در این شکل، فارغ از فاکتور زمان، می‌توانیم مشاهده کنیم درصد درستی الگوریتم knn از دیگر الگوریتم‌ها بیشتر است.

برای داده‌ی کاهش یافته

ردیف	الگوریتم	درصد درستی	زمان اجرا	امتیاز مدل ((acc/sqrt(time))
0	KNN	89.35	0.097	287.316
1	Logistic Regression	75.3	0.032	419.692
2	Decision Tree	84.7	0.017	644.72
3	Random Forest	78.85	0.224	166.496
4	SVM	65.8	6.264	26.29
5	Naive Bayes	66.35	0.002	1387.394

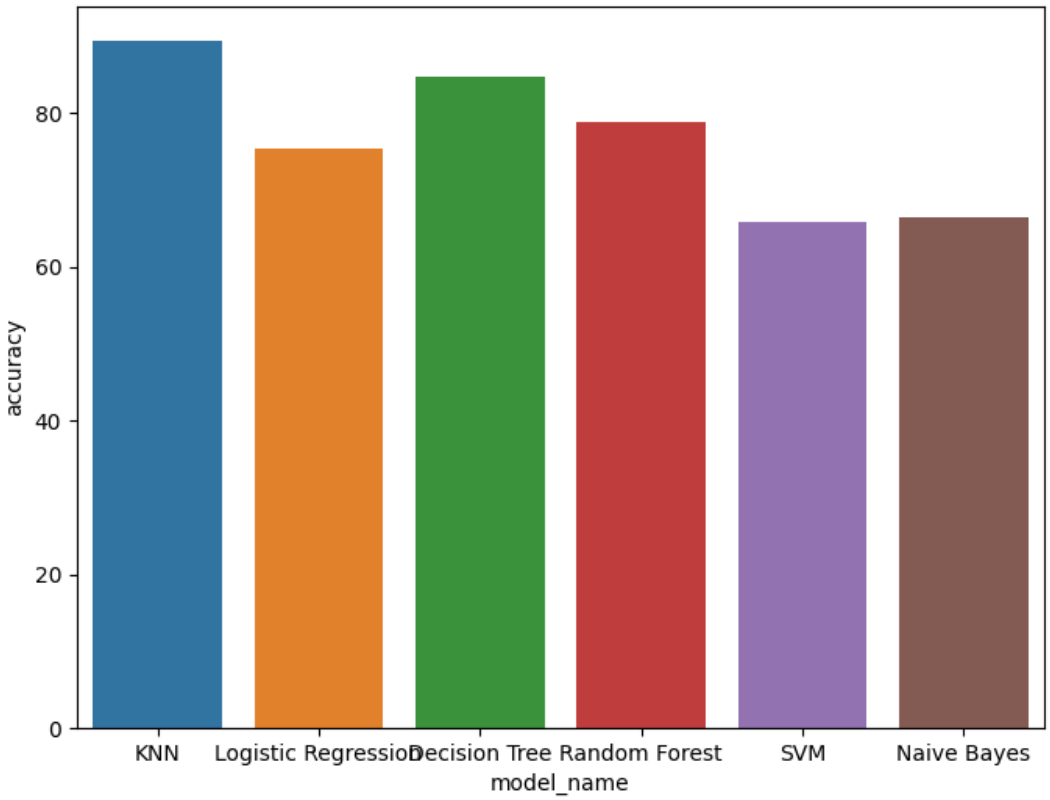
جدول ۲ :جدول ۱: نتایج مدل‌ها با داده‌ی کاهش یافته

در آزمایش دوم با تنها دو ستون app و click time زمان اجرا نسبت به حالت قبلی اختلاف خیلی زیادی ندارد.

الگوریتم	درستی برای ستون‌های کاهش یافته	درستی همه‌ی ستون‌ها
KNN	89.35	88.9
Logistic Regression	75.3	73.3
Decision Tree	84.7	87.4
Random Forest	78.85	85.5
SVM	65.8	67.2
Naive Bayes	66.35	73.8

جدول ۳ مقایسه‌ی نتایج داده با همه‌ی ستون‌ها در مقایسه با داده‌ی کاهش‌یافته

در جدول شماره ۳ می‌توانیم درصد درستی را برای دو آزمایش با ستون‌های متفاوت مقایسه کنیم. همان‌طور که مشاهده می‌کنید، در دو الگوریتم اول درصد درستی با اختلاف جزئی پس از حذف ستون‌ها افزایش یافته، اما در چهار الگوریتم دیگر، این مقدار کاهش یافته. بنابراین نتیجه می‌گیریم ستون‌های حذف شده، در پاسخ موثر بوده‌اند.



شکل ۱۶

و در این شکل، فارغ از فاکتور زمان، می‌توانیم مشاهده کنیم درصد درستی الگوریتم knn از دیگر الگوریتم‌ها بیشتر است.