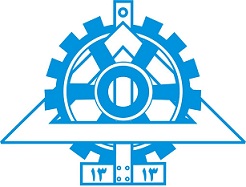
به نام پروردگار باران



­

دانشگاه تهران

پردیس دانشکده های فنی

دانشکده مهندسی برق و کامپیوتر

**درس یادگیری ماشین**

**گزارش پروژه پایانی**

**آنیتا قندهاری**

**810195533**

**سهیل شیروانی**

**810195416**

**بهمن 99**

* **آشنایی با مقالات مرتبط**

در این بخش هدف بررسی کامل مقالات مرتبط با کاربرد یادگیری ماشین در تشخیص بیماری­ها می­باشد و در این بخش مقاله زیر بررسی شده است:

A Multiple-Classifier Framework for Parkinson’s Disease Detection Based on Various Vocal Tests

1. خلاصه ای از مقدمه: بیماری پارکینسون ابتدا در سال 1817 توسط دکتر جیمز پارکینسون معرفی شد. این بیماری که پس از آلزایمر شایع ترین بیماری سیستم عصبی انسان می باشد، دارای دو گروه مختلف علائم در بیمار می باشد:

الف) علائم حرکتی (Motor): علائمی که بر حرکت و عضلات تاثیر می گذارند.

ب) علائم غیرحرکتی (Nonmotor): علائمی همچون مشکل در سیستم عصبی شناختی، مشکل در خواب و اختلال در حواس، می باشند.

تحقیقات در زمینه این بیماری نشان داده است که بیش از 90 درصد از افراد مبتلا به پارکینسون دچار اختلال در صحبت کردن هستند که یکی از علائم حرکتی در این بیماری می باشد.

بنابراین تحلیل صوت حاصل از صحبت بیمارن پارکینسون می تواند در تشخیص این بیماری بسیار مناسب باشد. این روش علاوه بر هزینه کم، قابلیت تشخیص این بیماری و همچنین نظارت بر بیمار را از راه دور دارا می ­باشد. در بیماران پارکینسون، اختلالات گفتاری ناشی از اختلالات عصبی می باشد که با ضعف، کندی و یا عدم هماهنگی عضلات برای تولید صدا همراه می باشد. اختلال گفتاری دارای 3 شکل: 1) hypophonia 2) monotonic 3) festination می باشد.

آزمون های آوایی مختلفی برای برای یافتن اختلال های گفتاری طراحی شده اند، که از مهم ترین آن ها:

الف) Sustained Phonation: در این آزمایش از فرد خواسته می شود تا یک صدا را به صورت ممتد بگوید و تا جایی که برایش ممکن است گام صدای خود را ثابت نگه دارد.

ب) Running Speech: در این شیوه یک جمله استاندارد به بیمار داده می شود تا آن را بازگو کند.

در اکثر مقالات برای ارزیابی تنها از داده هایی شامل یکی از این شیوه های آزمایش استفاده شده است. هرچند در این مقاله برای تشخیص با دقت بالاتر از مجموعه داده شامل چند صدای ضبط شده با آزمون های آوایی متفاوت استفاده شده است.

هدف اصلی در این مقاله، ابتدا یافتن طبقه بندی که بتواند به صورت منحصر به فرد برای هر صدا مثلا حرف a اعمال شود و در قدم بعد با استفاده از الگوریتم Voting بتوان نتایجی از طبقه بندها از صداهای مختلف را که تاثیر چندانی در تصمیم گیری ندارند را حذف کنیم.

1. در این مقاله از مجموعه داده Parkinson Speech Dataset with Multiple Types of Sound Recordings استفاده شده است.
2. این مجموعه داده توسط خود نویسنده تهیه نشده است و از وبسایت دانشگاه University of California, Irvine برداشته شده است.
3. در این پژوهش از مجموعه داده ای شامل اطلاعات صوتی از 40 فرد که از هر فرد 26 صدا شامل گفتن، 43 حرف الفبا، اعداد 1 تا 10، 9 کلمه و 4 جمله کوتاه، می باشد. از این سیگنال های صوتی 26 ویژگی مختلف استخراج شدند. در این تحقیق داده ها به 26 دسته تقسیم شدند که صدای های مربوط به هر دسته ذکر شده در بالا در یک بخش قرار گرفتند. برای مثال همه صداهای مربوط به حرف a در یک دسته قرار داده شده اند. این تقسیم بندی به این دلیل انجام شده است که دادن ترکیب حاصل از کلمات و حروف و اعداد مختلف به Classifier باعث خطا در تشخیص تمایز و خطا در دسته بندی می شود. برای شروع ابتدا داده ها با روش𝑧-score normalization، نرمالایز شده اند. در واقع مقدار میانگین داده ها در این حالت برابر یا صفر شده است و مقدار انحراف معیار به 1 تغییر یافته است.

در مرحله بعد برای کاهش تعداد ویژگی ها با توجه به کم بودن داده های موجود، از روش Pearson Correlation Coefficient Feature Selection استفاده شده است. در این روش ویژگی هایی که بیشترین Correlation را با خروجی داده ها را دارند، یافته شده است. سپس برای هر صدا از 26 صدای موجود، ویژگی های باقی مانده متناظر را پیدا شده است. در این مرحله تعدادی از اصوات هیچ ویژگی متناظری ندارند (unsuccessful vocal tests)، برای حل این مشکل دو راهکار استفاده شده است:

الف) MCFS: در این روش 4 تا از پرتکرار ترین ویژگی ها در اصوات دیگر یافته و سپس برای توصیف اصوات بدون ویژگی متناظر از این 4 ویژگی استفاده شده است.

ب) A-MCFS: در این روش تمامی اصوات بدون ویژگی متناظر حذف می شوند.

1. در این پژوهش طبقه بندهای k-NN با مقادیرk برابر با 1و 3و 5 و 7و همچنین SVM با Linear Kernel و RBF Kernel و به علاوه Naïve Bayes Classifier و Discriminant Analysis مقایسه شده اند. در آخر نیز جواب به دست آمده از به یک مدل Voting داده می شود تا نتیجه نهایی مشخص شود.
2. پارامترهای ارزیابی: 1) Accuracy 2) Sensitivity 3) Specifity 4) Matthew’s correlation coefficient(MCC)

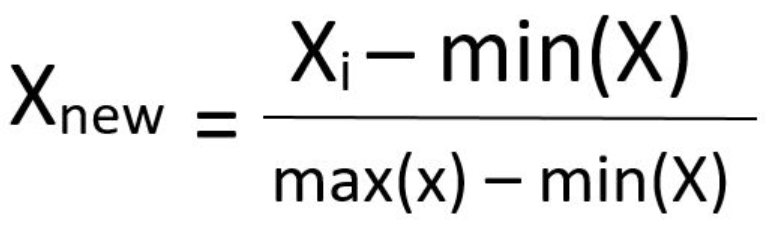
همچنین Method های مختلف برای اعمال داده به مدل ها عبارتند از: 1) LOSO 2) s-LOO 3) MCFS 4) A-MCFS

معیارها استفاده شده در این پژوهش معیارهای مناسبی هستند، اما با توجه به این که در ساختار داده های پزشکی تعادل در دسته های مختلف دیده نمی شود، بهتر است از Confusion Matrix و ROC Curve برای درک بهتر این که Classifier چقدر در تمایز بین دسته های مختلف مناسب است، استفاده شود.

1. در نهایت Discriminant analysis انتخاب شده است. همچنین دو جدول در این مقاله داده شده است که نشان دهنده برتری متود A-MCFS بر دیگر متودها دارد. در این پژوهش مشخص شد که بعضی از کلمات و حروف برای بررسی این که فرد به پارکینسون مبتلا می باشد یا نه، مناسب نیستند. در نهایت پیشنهاد شده است با استفاده از روش ارائه شده در این مقاله، می توان به بررسی و یافتن حروف و کلماتی پرداخت که ضبط آن ها باعث تشخیص مناسب بیماری پارکینسون شود.

* **روش های پیش پردازش و انتخاب ویژگی**

1. در ابتدا Id را از مجموعه داده حذف می کنیم. در بخش بعد تمامی مجموعه ویژگی ها را Scale کردیم. برای این کار از Min-Max-Scaler استفاده شده است. در این روش تمامی داده ها مطابق رابطه زیر به عددی بین 0 و 1 اسکیل می شوند.



* **مدل های گروهی**

مدل های گروهی مدل هایی است که پیشبینی های دو یا چند مدل را با هم ترکیب می کند. معمولا در بسیاری از پروژه ها که نتیجه عملکرد مهم است استفاده از یادگیری گروهی اهمیت ویژه ای دارد. این مدل ها ویژگی های مهمی دارند که نیاز است آنها را بدانیم چرا که ممکن است همیشه بهترین جواب را ندهند. این ویژگی ها عبارت اند از:

1. مدل های گروهی باعث ایجاد robustness

ساده ترین مجموعه این است که مدل را چندین بار در مجموعه داده های آموزش قرار دهید و با استفاده از یک آمار مانند میانگین رگرسیون یا مود طبقه بندی ، پیش بینی ها را ترکیب کنیم. نکته مهم این است که، هر مدل به دلیل الگوریتم یادگیری تصادفی ، تفاوت در ترکیب مجموعه داده آموزش یا تفاوت در خود مدل ، باید کمی متفاوت باشد.

این باعث کاهش تفاوت در پیش بینی های انجام شده توسط مدل می شود. میانگین عملکرد ​ تقریباً یکسان خواهد بود ، اگرچه بدترین و بهترین حالت عملکرد را به عملکرد متوسط ​​نزدیک می کند. در واقع این کار مقدار عملکرد قابل انتظار مدل را هموار می کند. که به این عملکرد قابل انظار Robustness نیز می گویند. در واقعا با این کار عملکرد مدل را برای حالت های مختل تقریبا نزدیک به هم کردیم تا مدل در همه جا جواب شکسان بدهد.

1. بالانس بین بایاس، واریانس

خطا در یک مدل یادگیری ماشین در 2 صورت گفته می شود، بایاس و واریانس. بایاس در واقع نشان دهنده ی این است که مدل با استفاده از یک تابع نگاشت چقدر می تواند نزدیک به تغییر ورودی به خروجی باشد. و واریانس در واقع مقدار تفاوت عملکرد مدل وقتی بر روی داده های آموزش مختلف اجرا می شود است. این دو مفهوم به هم وابسته اند. در واقع ما دنبال مدلی هستیم که بایاس و واریانس کمی داشته باشد ولی معمولا این کار ممکن نیست و زمانی که بایاس را کم می کنیم واریانس زیاد می شود و یا بر عکس.

استفاده از مدل های گروهی می تواند واریانس کلی مدل را کاهش دهد که در نتیجه می تواند باعث بهبود عملکرد مدل در پیشبینی داده ها شود. استفاده از مدل های گروهی برای کاهش خصوصیات واریانس خطاهای پیش بینی منجر به منجر به سود اصلی استفاده از مجموعه ها در وهله اول می شود: بهبود عملکرد مدل.

1. مدل گروهی برای افزایش عملکرد

ما به صراحت از یادگیری گروهی برای جستجوی عملکرد پیش بینی بهتر استفاده می کنیم ، مانند خطای کمتر در رگرسیون یا دقت بالا برای طبقه بندی. وقتی برای این دلیل استفاده می شود ، عملکرد مدل گروهی باید به طور متوسط ​​عملکرد بهتری نسبت به سایر مدل های گروه داشته باشد. اگر اینگونه نیست ، در عوض باید از عضوی که عملکرد بهتری دارد استفاده شود.

ممکن است و حتی معمول است که عملکرد یک مدل گروهی بهتر از بهترین اعضای گروه نباشد. این اتفاق می تواند در صورتی رخ دهد که گروه دارای یک مدل با عملکرد عالی باشد و اعضای دیگر هیچ مزیتی نداشته باشند یا گروه نتواند سهم خود را به طور موثر مهار کند. همچنین ممکن است عملکرد یک گروه نسبت به بهترین عضو گروه عملکرد بدتری داشته باشد. این نیز معمول است و معمولاً شامل یک مدل با عملکرد عالی می شود که پیش بینی های آن با یک یا چند مدل دیگر با عملکرد ضعیف بدتر می شود و گروه قادر به کمک موثر آنها نیست.

به همین ترتیب ، مهم است تاب یک مدل گروهی مناسب پیدا کرده و رفتار ان را با پارامتر ها تعیین کنیم و همین کار را برای هر یک اعضای آن مدل نیز انجام دهیم تا بتوانیم بهترین دقت را انتخاب کنیم.

* روش Bagging

Bootstrap به نمونه گیری تصادفی با جایگزینی اشاره دارد. Bootstrap به ما امکان می دهد بایاس و واریانس با مجموعه داده را بهتر درک کنیم. بوت استرپ شامل نمونه گیری تصادفی از زیر مجموعه کوچک داده از مجموعه کل داده است. این زیر مجموعه قابل جایگزینی است. انتخاب همه مثالها در مجموعه داده از احتمال برابر برخوردار است. این روش می تواند به درک بهتر میانگین و انحراف معیار از مجموعه داده کمک کند.

aggregating Bootstrap (یا به طور خلاصه Bagging) ، یک روش ساده و بسیار قدرتمند برای مدل های گروهی است. Bagging یک مدل از روش Bootstrap است که در الگوریتم یادگیری ماشین با واریانس بالا ، به طور معمول درخت تصمیم استفاده می شود.

1. فرض می کنیم N مشاهدات و M ویژگی وجود دارد. نمونه ای از مشاهده به طور تصادفی با جایگزینی (بوت استرپینگ) انتخاب می شود.
2. زیر مجموعه ای از ویژگی ها برای ایجاد مدلی با نمونه مشاهدات و زیر مجموعه ویژگی ها انتخاب می شوند.
3. ویژگی ای از زیر مجموعه انتخاب می شود که بهترین تقسیم را در داده های آموزش ارائه می دهد.
4. این کار تکرار می شود تا مدل های زیادی ایجاد شود و هر مدل به طور موازی آموزش داده می شود
5. پیش بینی براساس تجمیع پیش بینی ها از همه مدل ها ارائه می شود.

هنگام Baggin با درختان تصمیم ، کمتر نگران تک تک درختان هستیم که بر روی داده های آموزشی Overfitt شوند.

* مدل های گروهی دیگر استفاده شده:

1. یک مدل دیگر Random Forest است که بر اساس آموزش تعداد زیادی درخت تصمیم که هر کدام بر بخشی از داده اموزش می بینند است.
2. Adaboost روش دیگری از boosting است که داده ها را با استفاده از فرمولی تیکه به تیکه آموزش می دهد و سپس می تواند داده های ندیده را پیشبینی کند.
3. Votting این روش درواقع یک رای گیری بین چند مدل است. در این روش چند مدل با کل داده آموزش می بیند و سپس برای پیشبینی رای هر کدام از مدل ها را می گیرد و جواب با بیشترین رای را در نظر می گیرد.

* **روش های طبقه بندی**

**الف) طبقه بندهای Discriminative:**

1. Logistic Regression:

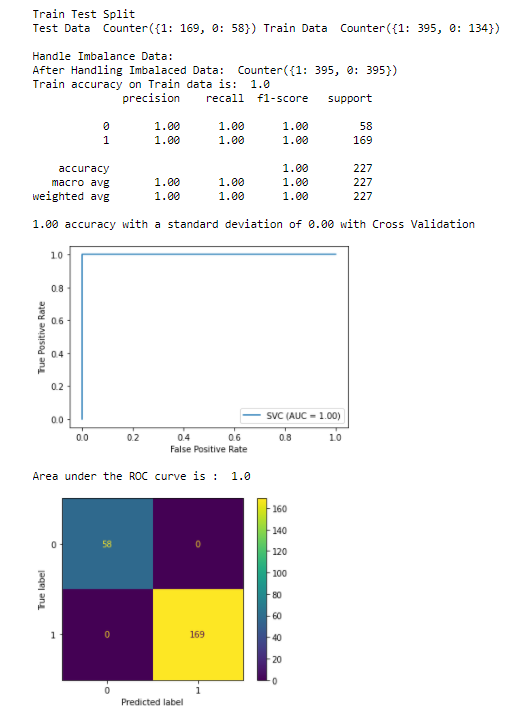
|  |  |
| --- | --- |
| روش انتخاب ویژگی | روش حل داده نامتعادل |
| LDA | Undersample |

نتایج:

1. Support Vector Machines:

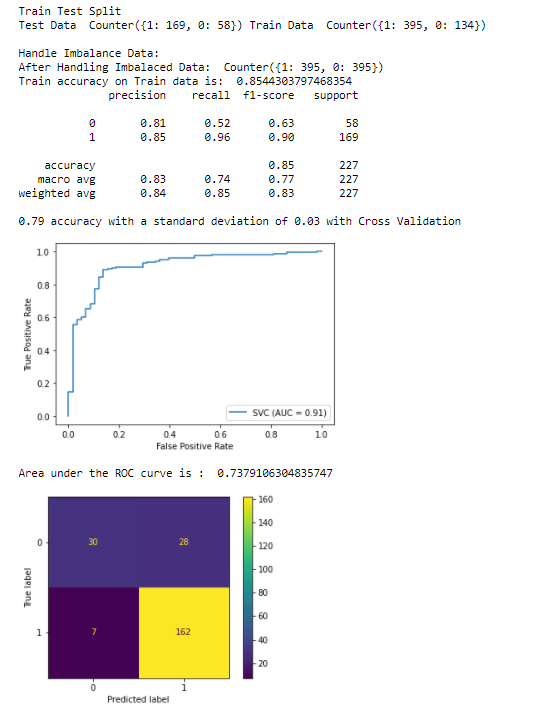
|  |  |
| --- | --- |
| روش انتخاب ویژگی | روش حل داده نامتعادل |
| LDA | SMOTE |

نتایج:



|  |  |
| --- | --- |
| روش انتخاب ویژگی | روش حل داده نامتعادل |
| LDA | SMOTE |

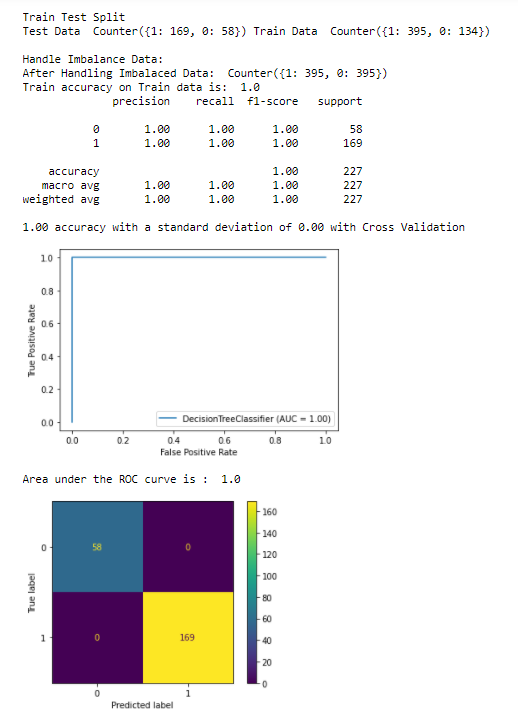
نتایج:



1. Decision Tree:

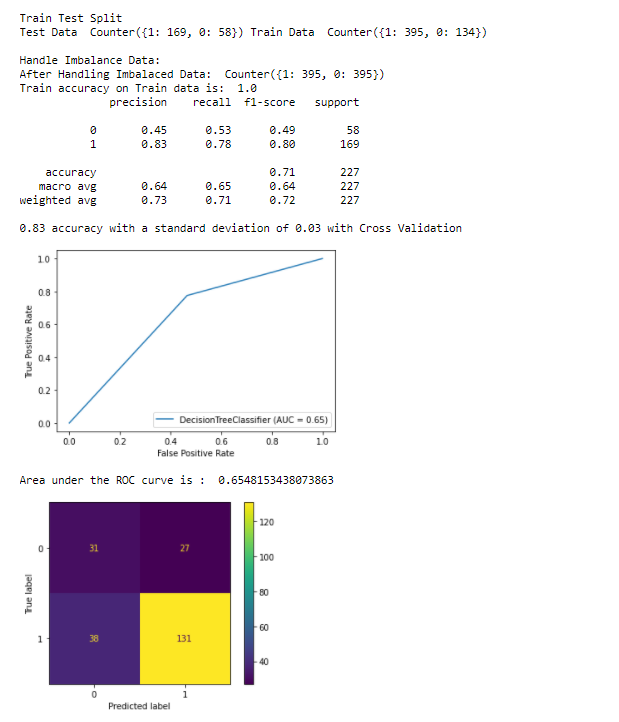
|  |  |
| --- | --- |
| روش انتخاب ویژگی | روش حل داده نامتعادل |
| LDA | SMOTE |

نتایج:



|  |  |
| --- | --- |
| روش انتخاب ویژگی | روش حل داده نامتعادل |
| PCA | SMOTE |

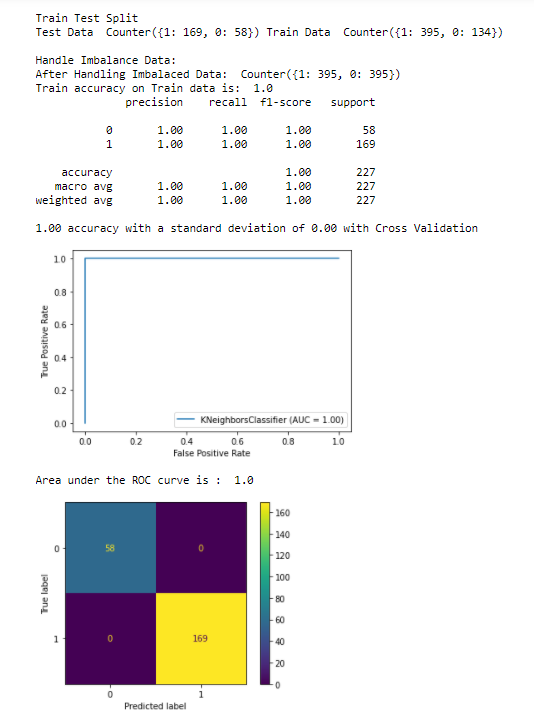
نتایج:



1. K Nearest Neighbor:

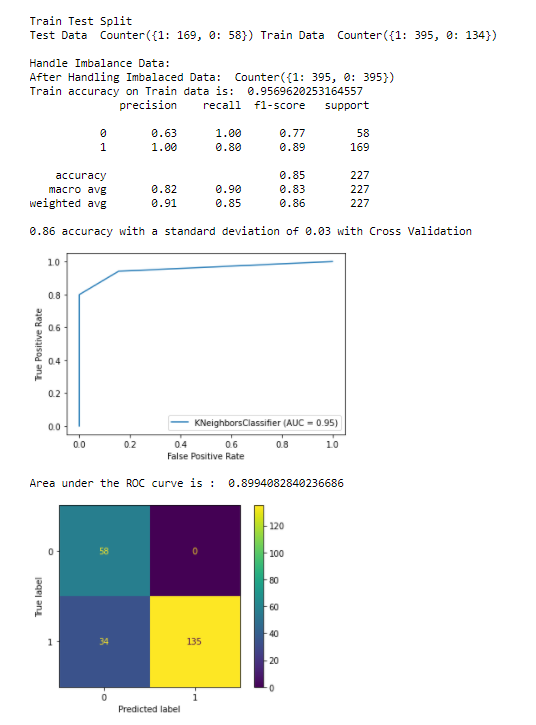
|  |  |
| --- | --- |
| روش انتخاب ویژگی | روش حل داده نامتعادل |
| LDA | SMOTE |

نتایج:



|  |  |
| --- | --- |
| روش انتخاب ویژگی | روش حل داده نامتعادل |
| PCA | SMOTE |

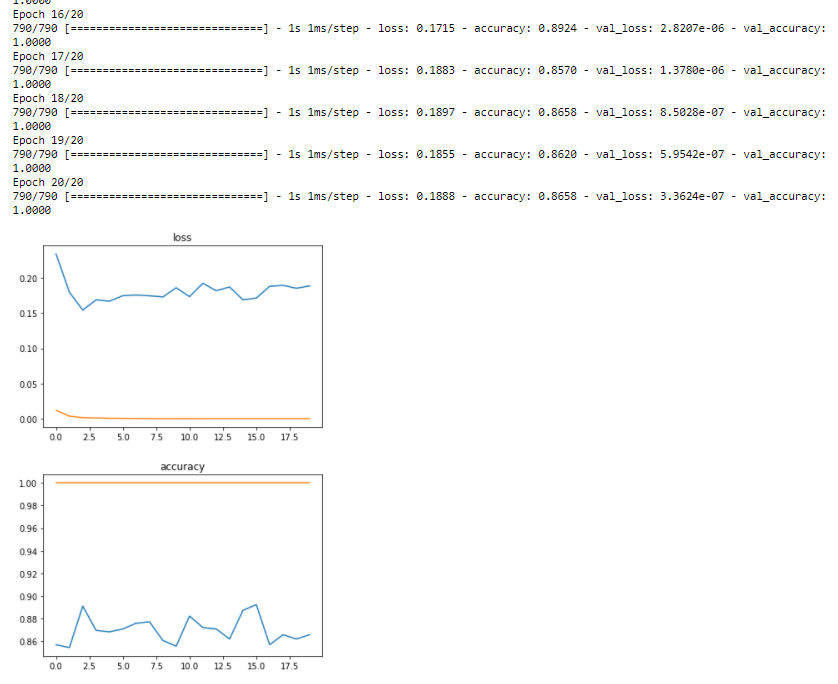
نتایج:



1. Multi Layer Perceptron:

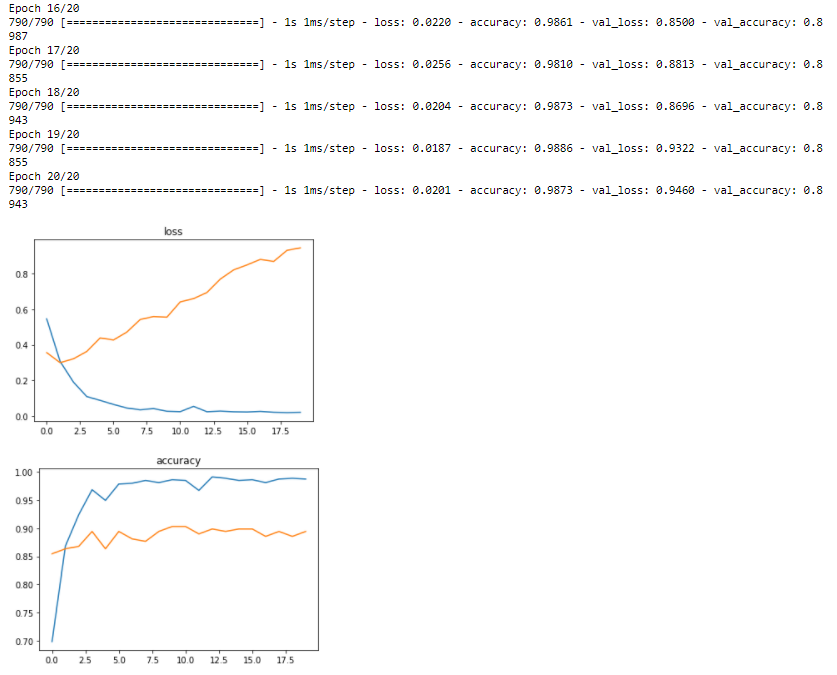
|  |  |
| --- | --- |
| روش انتخاب ویژگی | روش حل داده نامتعادل |
| LDA | SMOTE |

نتایج:



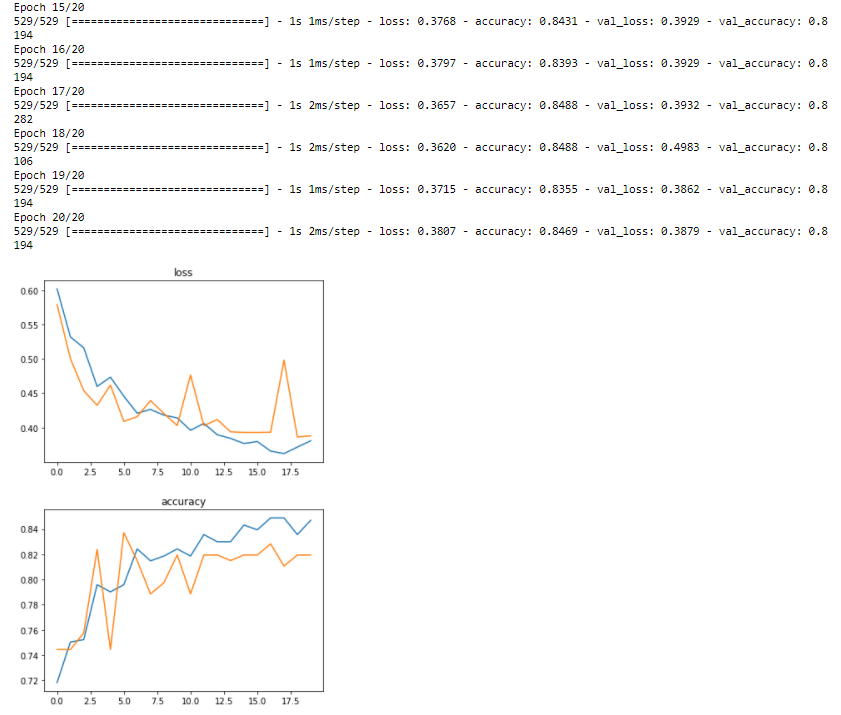
|  |  |
| --- | --- |
| روش انتخاب ویژگی | روش حل داده نامتعادل |
| PCA | SMOTE |

نتایج:



|  |  |
| --- | --- |
| روش انتخاب ویژگی | روش حل داده نامتعادل |
| PCA | SMOTE |

نتایج:

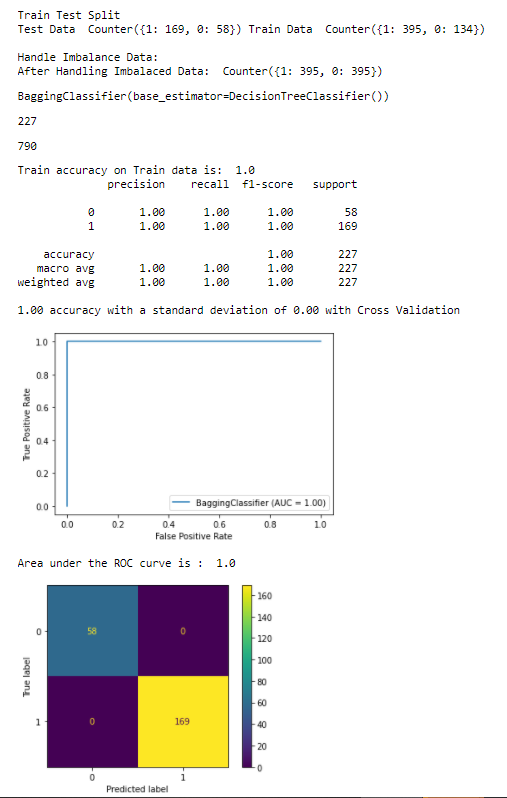


**ب) طبقه بندهای Ensemble:**

1. Bagging**:**

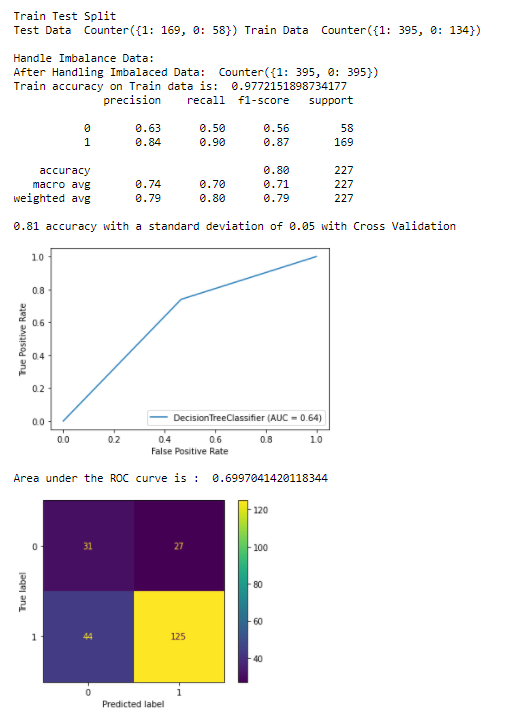
|  |  |
| --- | --- |
| روش انتخاب ویژگی | روش حل داده نامتعادل |
| LDA | SMOTE |

**نتایج:**



|  |  |
| --- | --- |
| روش انتخاب ویژگی | روش حل داده نامتعادل |
| PCA | SMOTE |

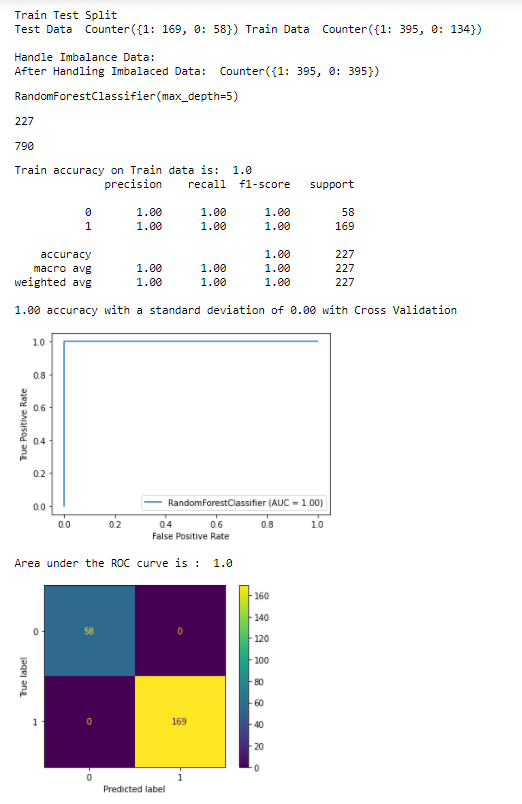
**نتایج:**



1. Random Forest

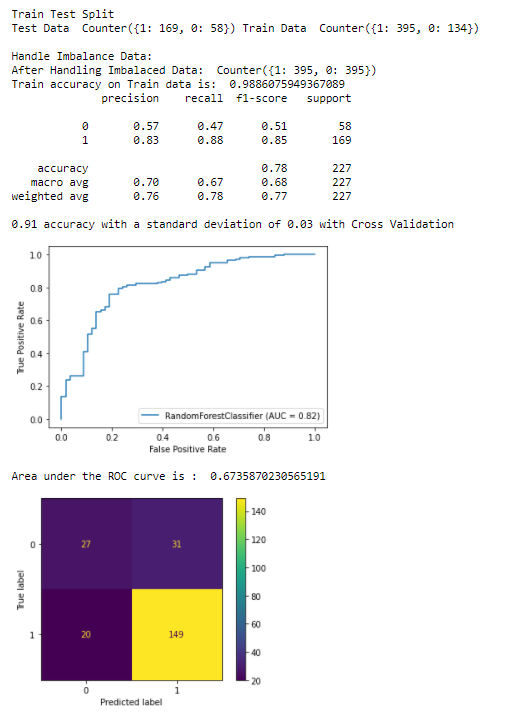
|  |  |
| --- | --- |
| روش انتخاب ویژگی | روش حل داده نامتعادل |
| LDA | SMOTE |

نتایج:



|  |  |
| --- | --- |
| روش انتخاب ویژگی | روش حل داده نامتعادل |
| PCA | SMOTE |

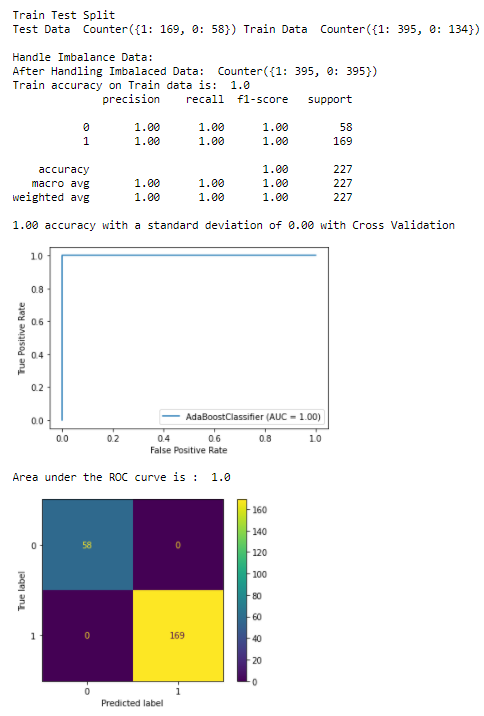
نتایج:



1. Adaboost

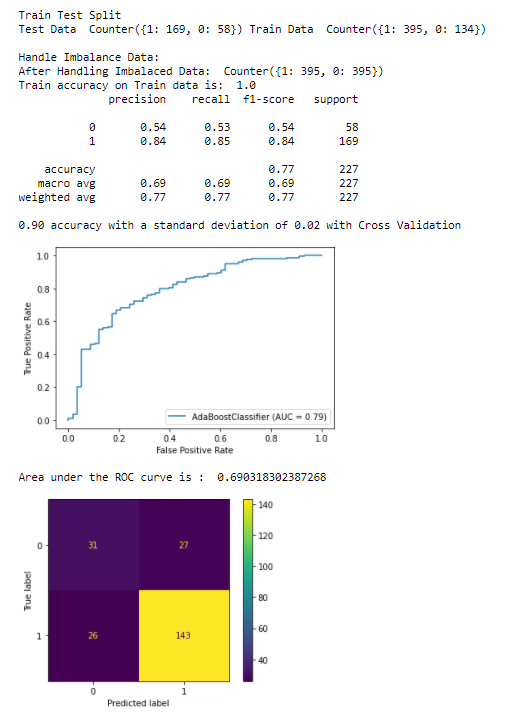
|  |  |
| --- | --- |
| روش انتخاب ویژگی | روش حل داده نامتعادل |
| LDA | SMOTE |

نتایج:



|  |  |
| --- | --- |
| روش انتخاب ویژگی | روش حل داده نامتعادل |
| PCA | SMOTE |

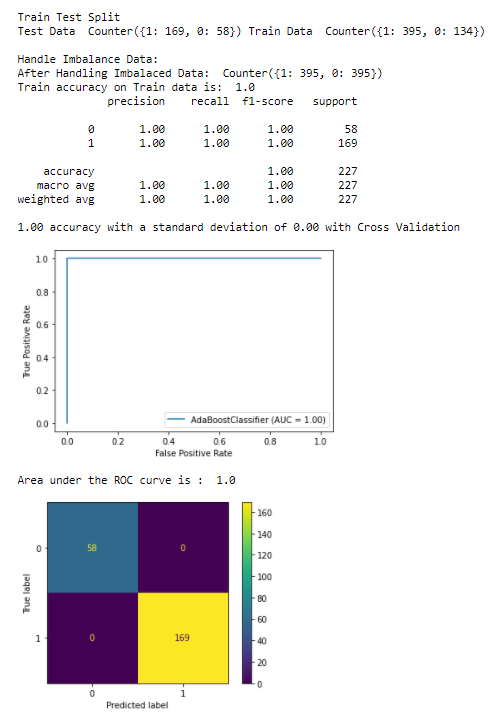
نتایج:



1. Voting:

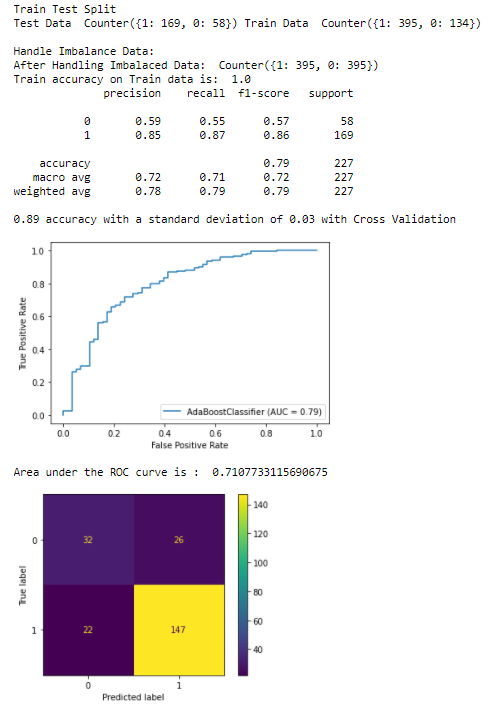
|  |  |
| --- | --- |
| روش انتخاب ویژگی | روش حل داده نامتعادل |
| LDA | SMOTE |

نتایج:

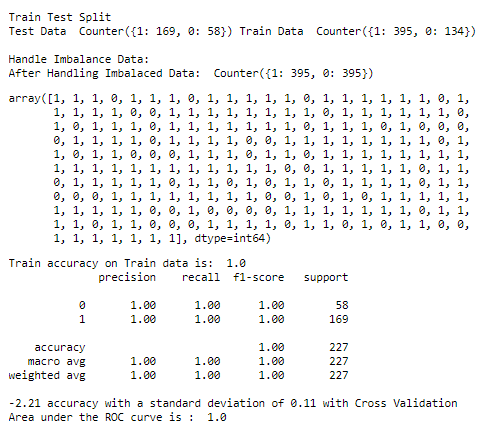


|  |  |
| --- | --- |
| روش انتخاب ویژگی | روش حل داده نامتعادل |
| PCA | SMOTE |

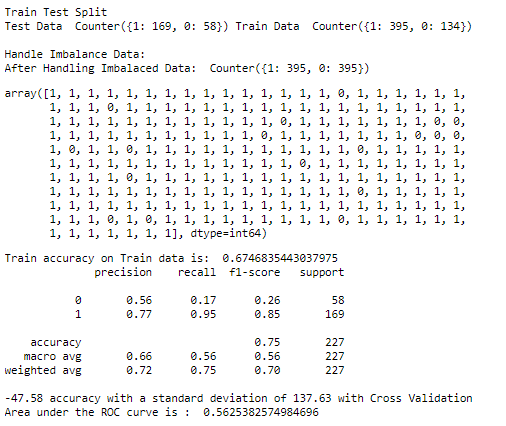
نتایج:



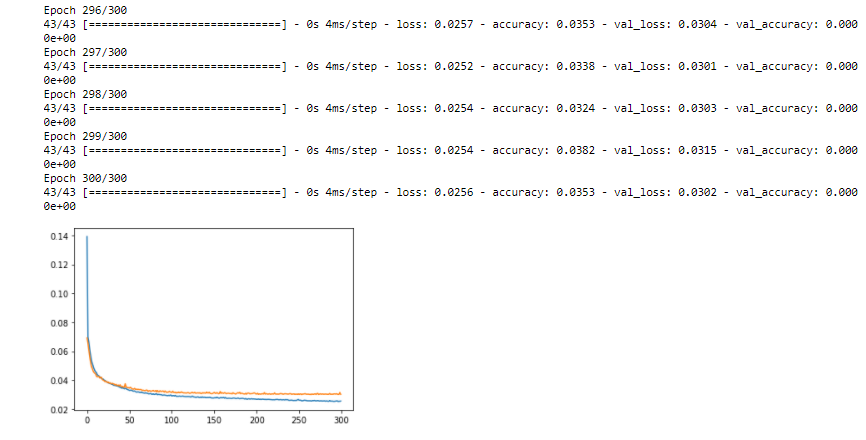
LDA, sMote

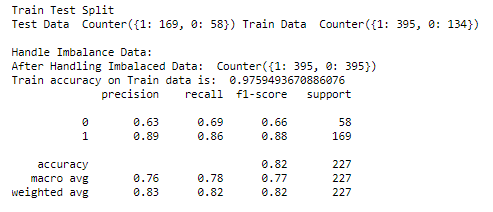


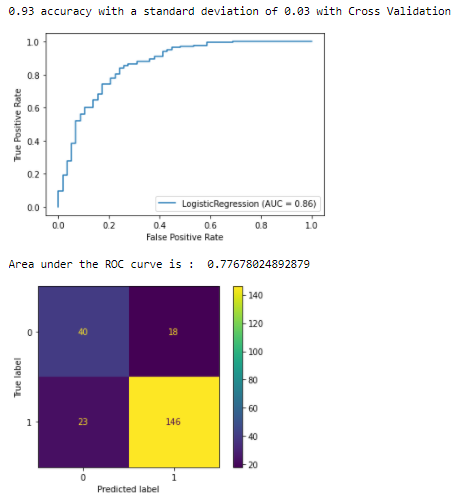
PCA, Smote



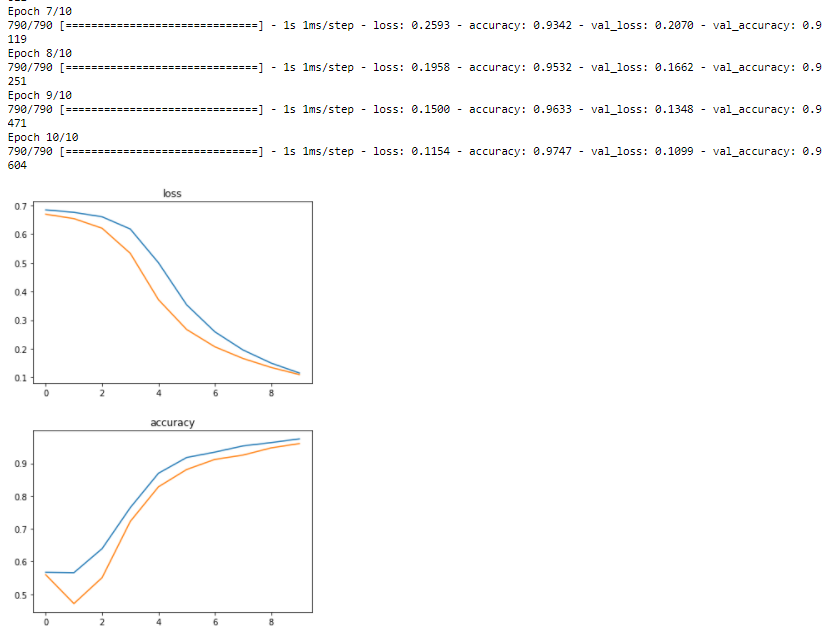
Auto Encoder, Smote







RBF, Smote, LDA



RBF, Smote, PCA

