Artificial Intelligence

Classification Algorithms and their analysis

Assignment 01

Zeeshan Ahmed]
5/29/2021

Table of Contents

Purpose of data set (Referenced to Kaggel)

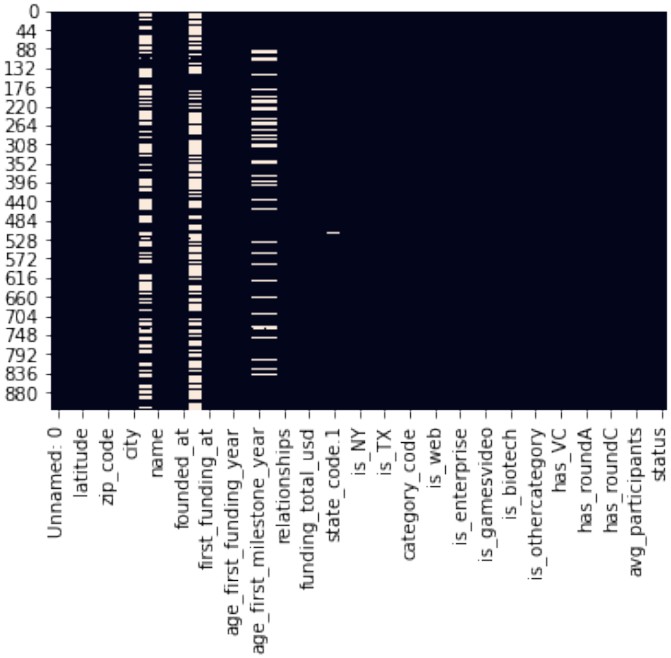
A startup or start-up is a company or project begun by an entrepreneur to seek, develop, and validate a scalable economic model. While entrepreneurship refers to all new businesses, including self-employment and businesses that never intend to become registered, startups refer to new businesses that intend to grow large beyond the solo founder. Startups face high uncertainty and have high rates of failure, but a minority of them does go on to be successful and influential. Some startups become unicorns: privately held startup companies valued at over US$1 billion. [Source of information: Wikipedia]

Source : https://www.kaggle.com/manishkc06/startup-success-prediction

Description of dataset

Before the pre-processing

- ✓ This data set contains 923 rows
- ✓ This data set contain 49 columns
- ✓ This data set contains the null values
    - Unnamed: 6 493
    - closed_at 588
    - age_first_milestone_year 152
    - age_last_milestone_year 152
    - state_code.1 1



- ✓
- ✓ This dataset has a column named status which will be used for classification
    - 597 Acquired class related records
    - 326 Closed class related records
        - These shows unbalancing in dataset

Then I have performed the preprocessing on this data set. You can find the file here (This generates two files): https://github.com/ZeeWING-Pr

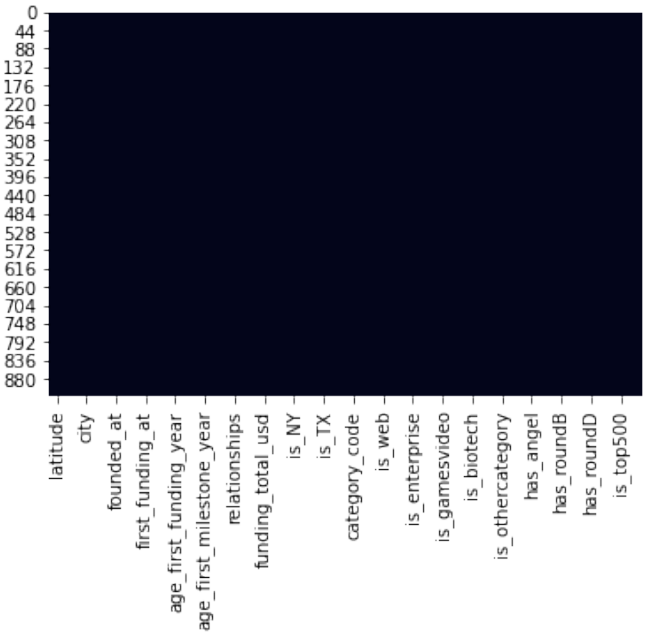
ojects/Start-up-pridiction-dataset-analysis/blob/main/Preprocessing-dataset.ipynb
After pre-processing we have two files (pre-processed datasets on is with normalized values and other is with normal values).

So for further process we will use the normalized dataset pre-processed file.

So after pre-processing we have following differences in dataset.

After pre-processing (SMOTE is applied)

- ✓ This data set contains 1194 rows
- ✓ This data set contain 40 columns
- ✓ This data set contains the null values
  - Unnamed: 6 0
  - closed_at 0
  - age_first_milestone_year 0
  - age_last_milestone_year 0
  - state_code.1 0



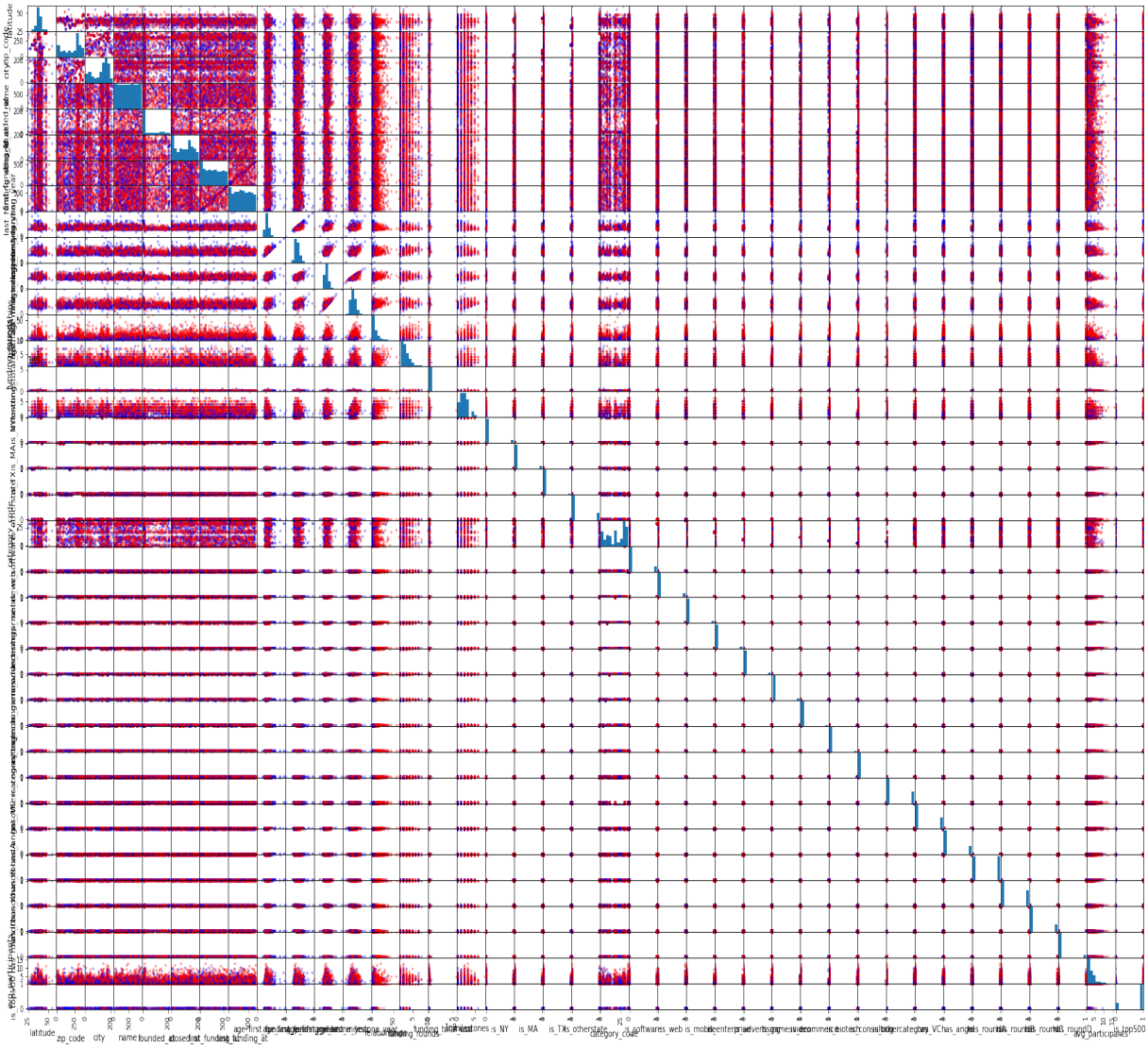
- ✓
- ✓ This dataset has a column named status which will be used for classification (These results are after applying the smote)
  - 597 Acquired class related records
  - 597 Closed class related records

    These shows unbalancing in dataset

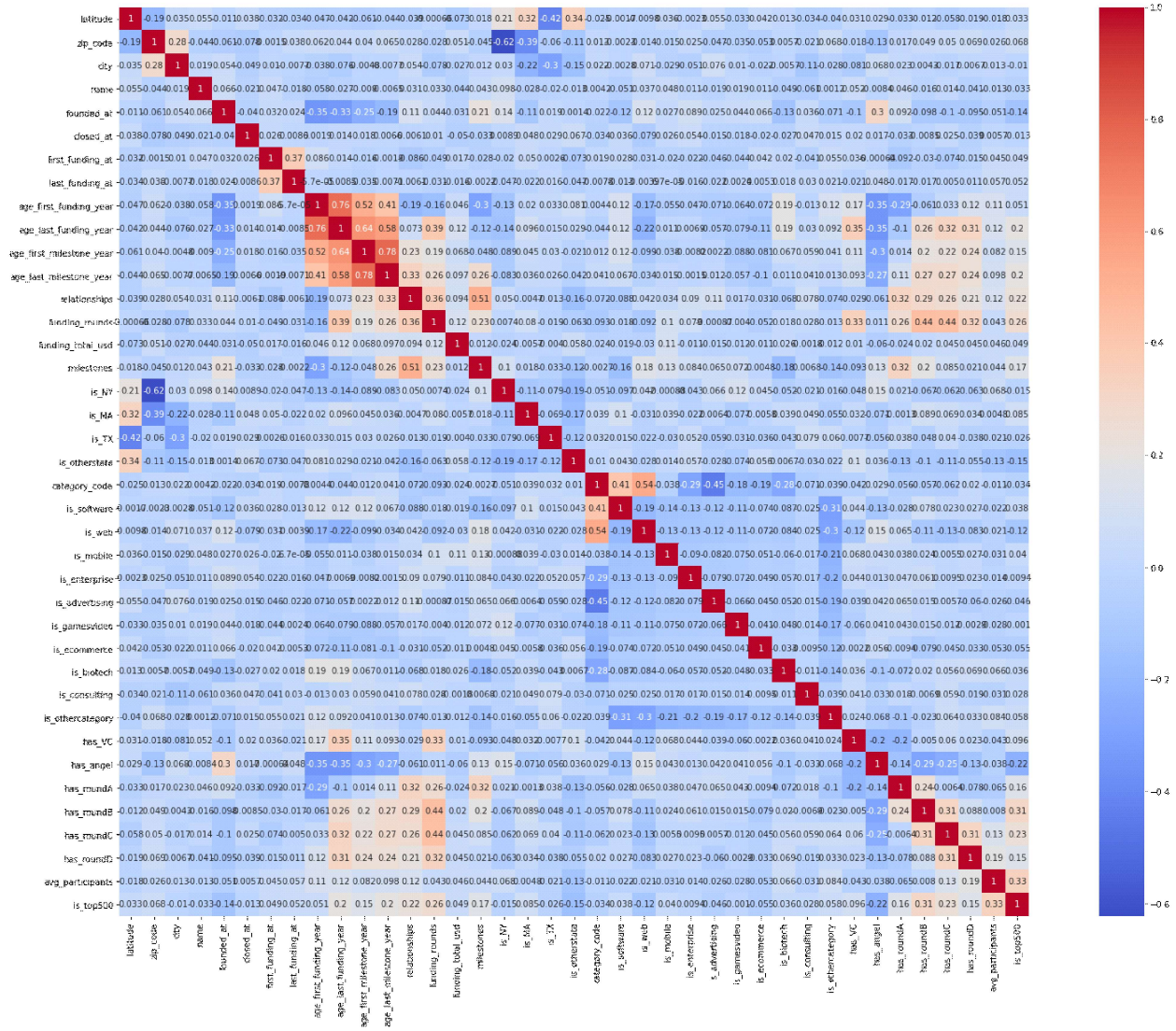
Corelation graphs

Note that while pre-processing I settled threshold around 0.7>= for strong co-relation and on this basis I have done feature selection.

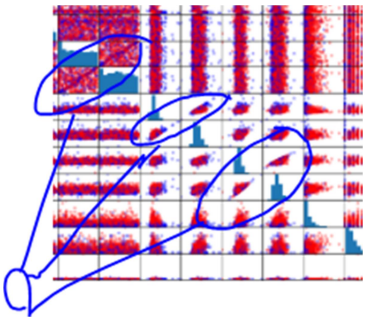
Scatter plot

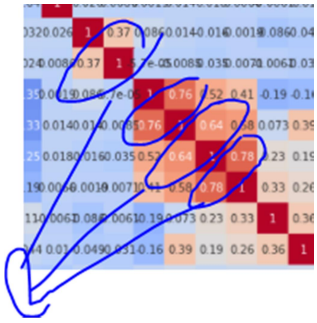


Heat graph.



So here are some close shots of both showing same good co-related features.





Applying classifier algorithms

Find the code at : https://github.com/ZeeWING-Projects/Start-up-pridiction-dataset-analysis/blob/main/Classifcation-Algorithms.ipynb

Now we apply multiple type of classification algorithms on our pre-processed datasets. As we are using dataset-2 and its standardized version. While performing these classification algorithms we will note few things like we will notice the results for with varying different parameters like split size, providing non-normalized values and providing un smote data (For just information note that we use somting when we have unequal number of records for each class). From word somted I mean smoting process is applied. It is known as class-imbalance issue in data mining. If we don't make data equal with respect to labels then our classifier will be biased.

Following is the list of algorithms which we will be using in each test.

1. SVC
2. NuSVC
3. LinearSVC
4. KNeighborsClassifier
5. GaussianNB
6. RandomForestClassifier
7. ExtraTreesClassifier
8. DecisionTreeClassifier

Note that we have selected all columns which we got after pre-processing.

Test 01: [Different test-sizes, dataset Is normalized, and dataset is smoted]

Results: Following table shows the accuracy of all mentioned algorithms on different test size.

	Training Part					
	95%	90%	80%	60%	50%	30%
SGD Classifier	58.33%	52.50%	54.81%	52.30%	52.26%	48.21%
SVG	53.33%	59.17%	53.97%	52.30%	52.26%	47.97%
NuSVC	53.33%	59.17%	53.97%	52.30%	52.26%	47.97%
LinearSVC	46.67%	55.83%	61.51%	60.67%	47.74%	48.21%
KNeighborsClassifier	61.67%	64.17%	66.53%	65.90%	62.98%	61.00%
GaussianNB	41.67%	46.67%	44.77%	48.33%	49.75%	48.92%
Random Forest	81.67%	85.00%	81.17%	81.17%	81.24%	77.27%
Extra Trees	83.33%	84.17%	80.33%	79.71%	80.57%	77.99%
Dedicion Tree	73.33%	78.33%	73.22%	71.97%	73.20%	67.82%

Following are the best points of split where the specific algorithm gives best accuracy.

Best points of spiriting dataset for training and testing

Best point to spilit dataset			
	Test Part	Traning Part	Accuracy
SGD Classifier	5%	95%	58.33%
SVG	10%	90%	59.17%
NuSVC	10%	90%	59.17%
LinearSVC	20%	80%	61.51%
KNeighborsClassifier	20%	80%	66.53%
GaussianNB	50%	50%	49.75%
Random Forest	10%	90%	85.00%
Extra Trees	10%	90%	84.17%
Dedicion Tree	10%	90%	75.83%

These are the points of split which I will be need when I will choose any of above algorithms for classification on my dataset.

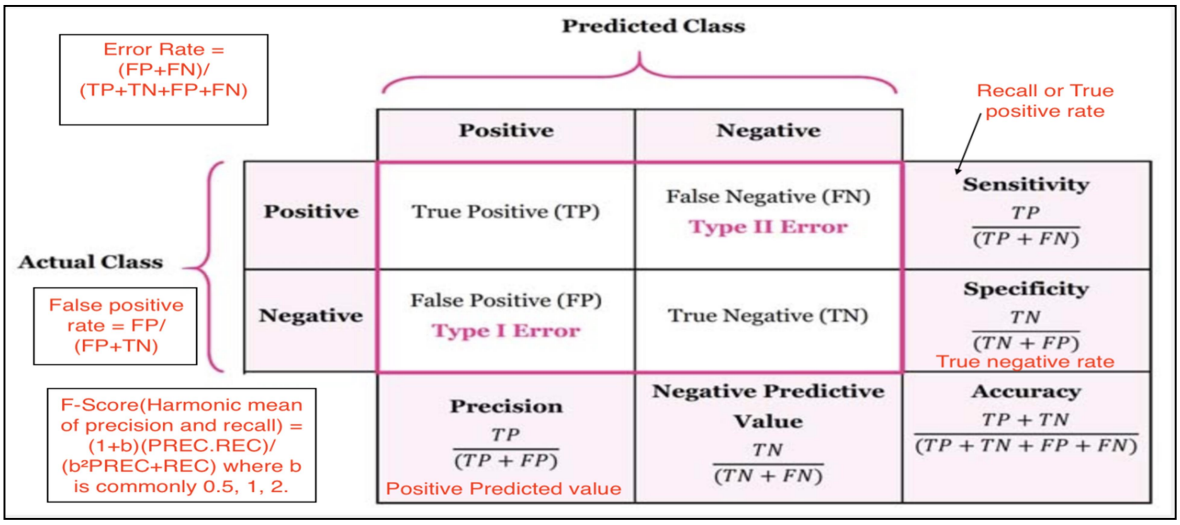
Worst points of spiriting dataset for training and testing

Wrost point to spilit dataset			
	Test Part	Traning Part	Accuracy
SGD Classifier	70%	30%	48.21%
SVG	70%	30%	47.97%
NuSVC	70%	30%	47.97%
LinearSVC	5%	95%	46.67%
KNeighborsClassifier	70%	30%	61.00%
GaussianNB	5%	95%	41.67%
Random Forest	70%	30%	77.27%
Extra Trees	70%	30%	77.99%
Dedicion Tree	70%	30%	67.82%

These are the points of split which I will be have to never select when I will choose any of above algorithms for classification on my dataset.

Now we will measure the further parameters which will help use to decided which is better performing algorithm.

So we will try to analyze following things



Sensitivity:

- True Positive recognition rate
  - **Sensitivity = TP/P**

Specificity

- True Negative recognition rate
  - **Specificity = TN/N**

Error rate

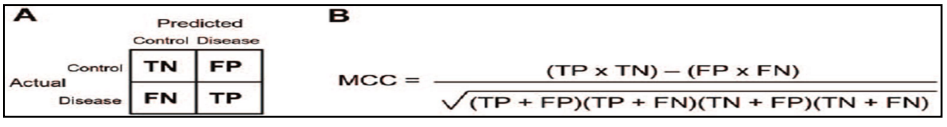
- *1 – accuracy*, or
  - **Error rate = (FP + FN)/All**

Precision

- exactness – what % of tuples that the classifier labeled as positive are actually positive
- $$precision = \frac{TP}{TP + FP}$$

Cross-validation (By using python)

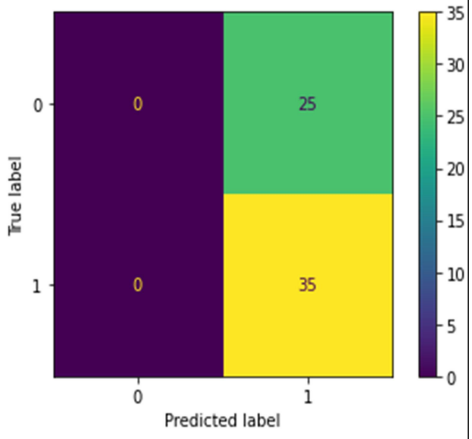
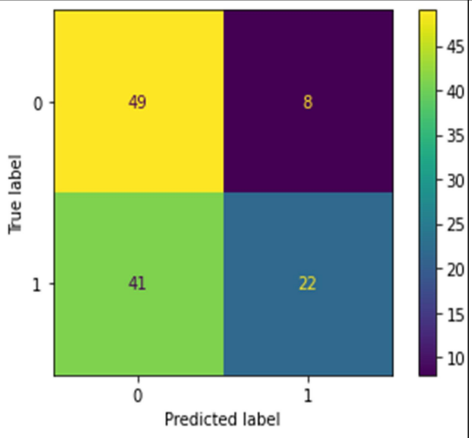
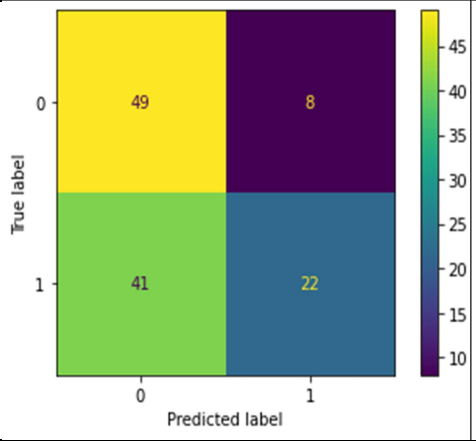
MCC

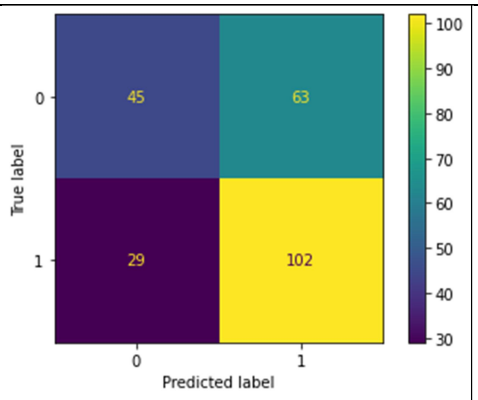
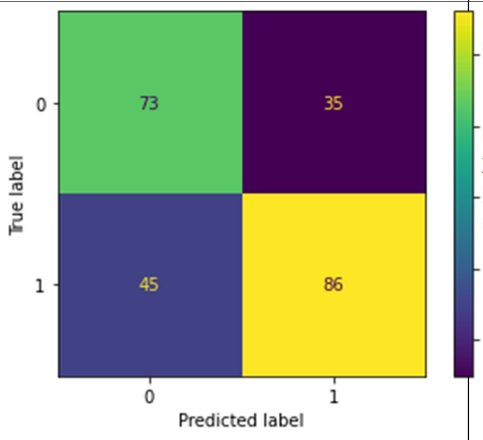
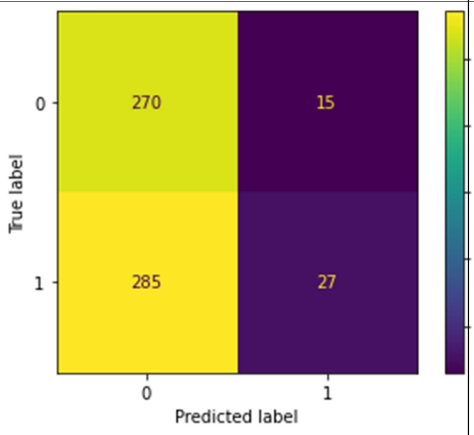
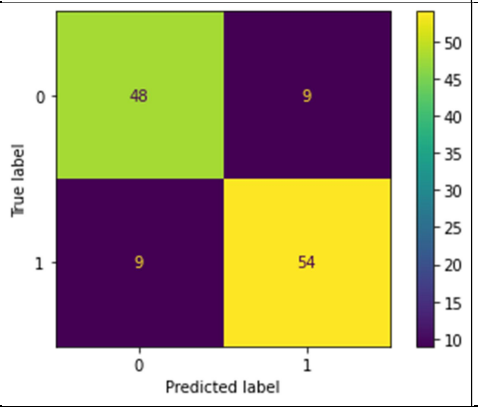
- 

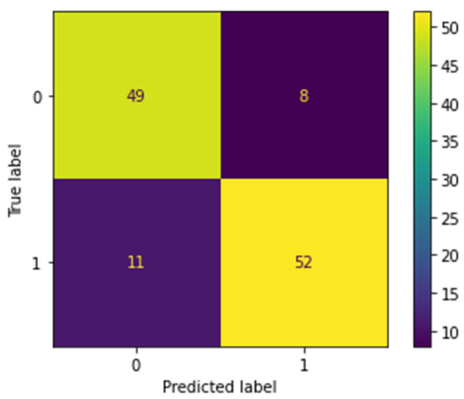
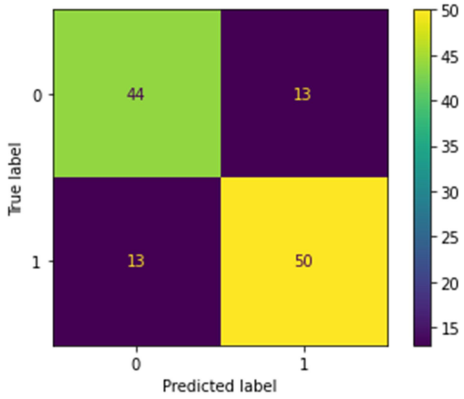
Note that I have used the dataset-2 for classification and following results are based on that dataset.

1 : “Acquired ” and 0: “Closed” Number of records while classification 1194

Classification results summary and details

Algorithm Name	Accuracy	Training part	Confusion matrix	Description Results
SGD Classifier	58.33%	95%		Sensitivity is 100.0 % Specificity is 0.0 % Precision is 58.333 % MCC is 0.352 Error rate is 0.417 %
SVC	59.17%	90%		Sensitivity is 34.921 % Specificity is 85.965 % Precision is 73.333 % -ve prid: are 54.444 % Calc: Accuracy is 59.167 % MCC is 0.241 Error rate is 0.408 % Cross validation score: 64.68% (+/- 0.39%)
NuSVC	59.17%	90%		Sensitivity is 34.921 % Specificity is 85.965 % Precision is 73.333 % -ve prid: are 54.444 % Calc: Accuracy is 59.167 % MCC is 0.241 Error rate is 0.408 % Cross validation score: 57.74% (+/- 12.39%)

LinearSVC	61.51 % %	80%		Sensitivity is 77.863 % Specificity is 41.667 % Precision is 61.818 % -ve prid: are 60.811 % Calc Accuracy is 61.506 % MCC is 0.21 Error rate is 0.385 % Cross validation score: 42.13% (+/- 27.06%)
KNeighborsClassifier	66.53%	80%		Sensitivity is 65.649 % Specificity is 67.593 % Precision is 71.074 % -ve prid: are 61.864 % Calc Accuracy is 66.527 % MCC is 0.331 Error rate is 0.335 % Cross validation score: 63.06% (+/- 5.59%)
GaussianNB	49.75%	50%		Sensitivity is 8.654 % Specificity is 94.737 % Precision is 64.286 % -ve prid: are 48.649 % Calc Accuracy is 49.749 % MCC is 0.066 Error rate is 0.503 % Cross validation score: 42.67% (+/- 21.14%)
RandomForestClassifier	85.00 %	90%		Sensitivity is 85.714 % Specificity is 84.211 % Precision is 85.714 % -ve prid: are 84.211 % Calc Accuracy is 85.0 % MCC is 0.699 Error rate is 0.15 % Cross validation score: 79.31% (+/- 2.50%)

ExtraTreesClassifier	84.17 % %	90%		Sensitivity is 82.54 % Specificity is 85.965 % Precision is 86.667 % -ve prid: are 81.667 % Calc Accuracy is 84.167 % MCC is 0.684 Error rate is 0.158 % Cross validation score: 75.84% (+/- 0.79%)
Decision TreeClassifier	78.33 %	90%		Sensitivity is 79.365 % Specificity is 77.193 % Precision is 79.365 % -ve prid: are 77.193 % Calc Accuracy is 78.333 % MCC is 0.566 Error rate is 0.217 % Cross validation score: 70.10% (+/- 5.29%)

Following is the piece of code which I have used for calculating the result.

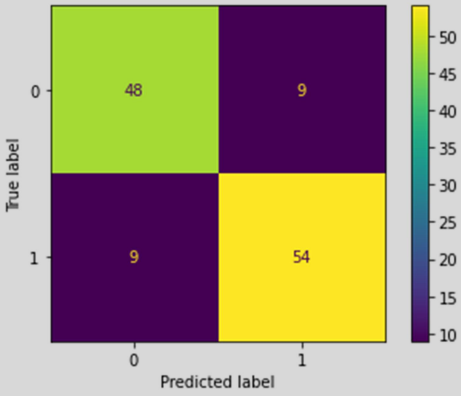
Find code at : https://github.com/ZeeWING-Projects/Start-up-pridiction-dataset-analysis/blob/main/Confusion%20matrix%20calculator.ipynb

Now further we will just take the best algorithm and will test and observe the effect of varying parameters on 0.1 testsize (Since it is the point where this algorithem has performed best)

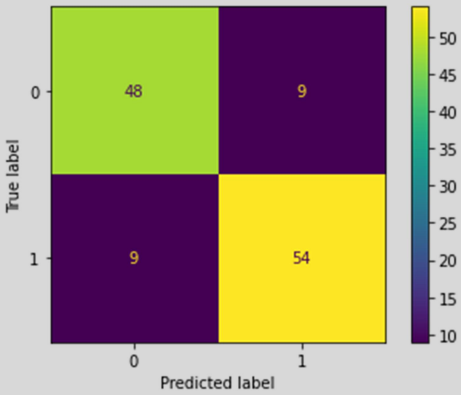
Test 02: [test-size=0.1, dataset is un normalized, and dataset is smoted]

.Results:

With normalized dataset

Algorithm Name	Accuracy	Training part	Confusion matrix	Description Results
RandomForestClassifier	85.00 %	90%		Sensitivity is 85.714 % Specificity is 84.211 % Precision is 85.714 % -ve prid: are 84.211 % Calc Accuracy is 85.0 % MCC is 0.699 Error rate is 0.15 % Cross validation score: 79.31% (+/- 2.50%)

Without normalized dataset values.

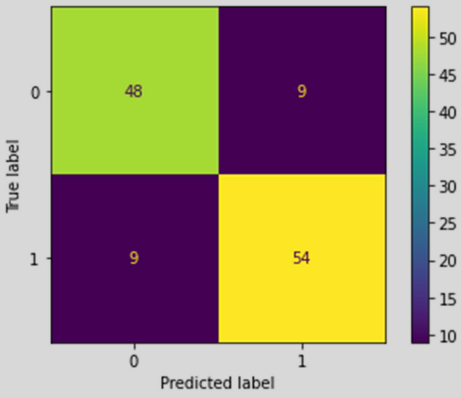
Algorithm Name	Accuracy	Training part	Confusion matrix	Description Results
RandomForestClassifier	85.00 %	90%		Sensitivity is 85.714 % Specificity is 84.211 % Percision is 85.714 % -ve prid: are 84.211 % Calc Accuracy is 85.0 % MCC is 0.699 Error rate is 0.15 % Cross validation score: 79.63% (+/- 2.09%)

No difference.

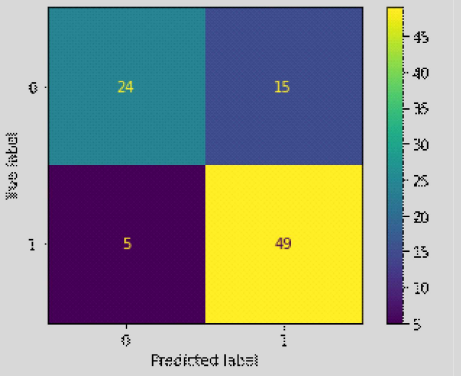
Test 03: [test-size=0.1, dataset is normalized, and dataset is un smoted]

This means there will unbalance in number of records with respect to classes.

With smoted dataset values

Algorithm Name	Accuracy	Training part	Confusion matrix	Description Results
RandomForestClassifier	85.00 %	90%		Sensitivity is 85.714 % Specificity is 84.211 % Precision is 85.714 % -ve prid: are 84.211 % Calc Accuracy is 85.0 % MCC is 0.699 Error rate is 0.15 % Cross validation score: 79.31% (+/- 2.50%)

Without smoted dataset values

Algorithm Name	Accuracy	Training part	Confusion matrix	Description Results
RandomForestClassifier	78.49 %	90%		Sensitivity is 90.741 % Specificity is 61.538 % Percision is 76.562 % -ve prid: are 82.759 % Calc Accuracy is 78.495 % MCC is 0.557 Error rate is 0.215 % Cross validation score: 80.28% (+/- 1.54%)

There is the difference.

Summary of report

For performing all required tasks of this assignment I chose a dataset from Kaggle which is about the prediction of a startup's success. This data set contained 49 columns and 923 rows before pre-processing and this number changed to 40 columns and 1194 rows. The reason for the increment in rows is the making rows balanced for unbiased training of the dataset. Later during the tests, it is also observed this unbalanced number of records with respect to class label affects the classification results. Other than this I found so many null values, string values, and unnormalized values which were converted to required forms while pre-processing. After describing the basic statics of pre-processed and un pre-processed datasets, I applied the classification algorithms. I applied all well know algorithms that we learned in labs. so I noticed the following interesting features while performing the tests.

- ✓ As int test 01 I observed that, with changing the test size or we can say by changing the training part, the accuracy of some algorithms is improved till the 95% of training parts like SGD classifier and some algorithms has increased till the 90%, for example, random forest classification algorithm. So it showed that different algorithms work best on different test sizes and some are highly affected by the change in partitioning and some are more robust.
- ✓ And after this, I noticed just noticed the best points of each algorithm and found the confusion matrix on that point. And by using that confusion matrix I calculated the model evaluation measures like MCC, accuracy, sensitivity, precision, and other things as well. So I found that for the chosen data set Randomforest is showing the best accuracy (85%) with MCC (0.699 which is showing it is a very good algorithm). And the percentage of data is 90% for this performance which means 10 % test size so this algorithm showed the best results here. Now if I want to make an application by using this dataset and I have to choose this algorithm for classification the now I know that how much training data I have to give for the best accuracy.
- ✓ After that found that this algorithm works the same for normalized and unnormalized values.
- ✓ And in the last test, I noticed the impact of balanced data by giving the imbalanced dataset values. As for balancing, I applied an overfitting algorithm called smoting in python. So the effect was like it reduced the accuracy by giving the imbalanced dataset values. And along with the reduction in accuracy the reduction in MCC score was also noticed.

Find everything related to this assignment at : https://github.com/ZeeWING-Projects/Start-up-pridiction-dataset-analysis