Startup Success

The aim of this project is to try to understand why some startups succeed and others failed, by using ML tools.

1. We have the data of 472 startups in total and 116 columns, of which ‘dependent company status’ is the label (value we need to predict).

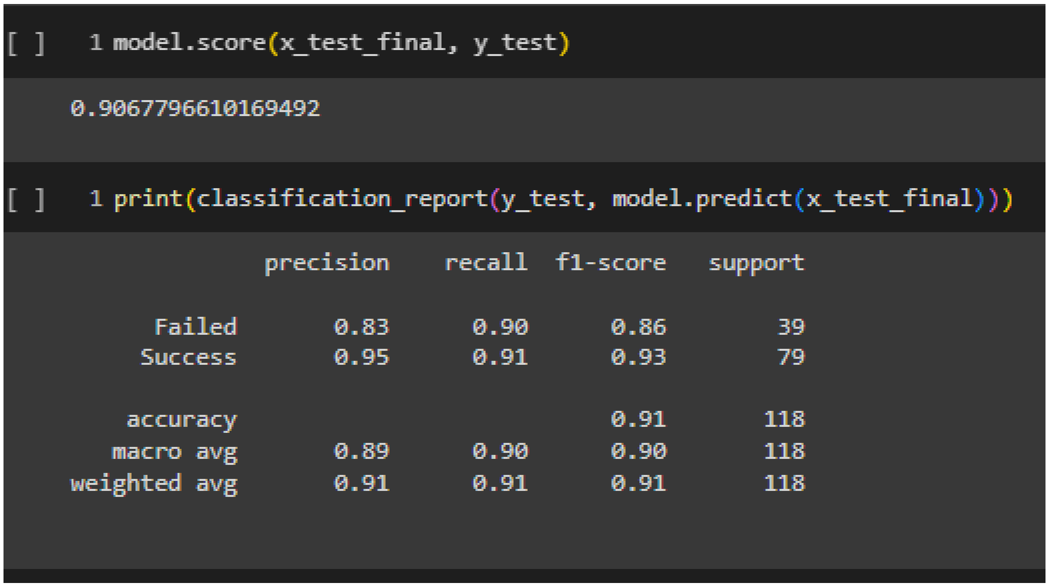
2. 305 of 472 startups have succeeded in our dataset, so the data is unbalanced.
3. We have nan values in our dataset.

Because the dataset size is not very large I decided to keep all the rows and delete the columns with nan values.

After dropping columns with nan values and also the ‘Company name’ column, 91 columns remain.

Because we have this many columns, instead of exploring them manually, I decided to train a random forest classifier model and then explore the features with high importance scores.

Below are the evaluation results of our first model, trained on all 90 features.

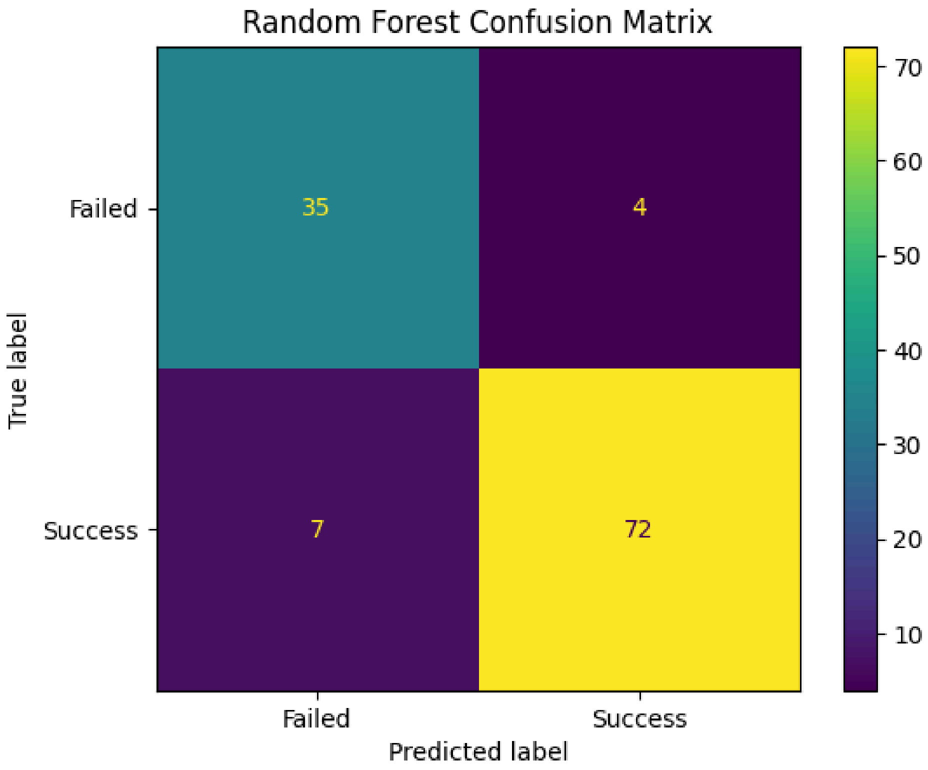


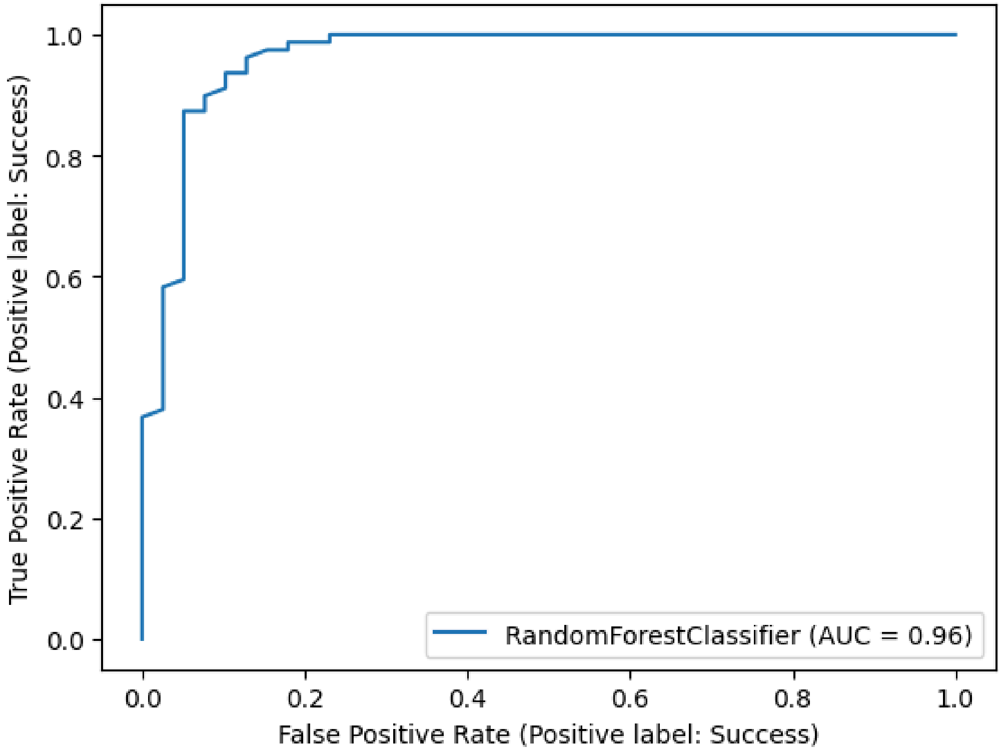
```
model.score(x_test_final, y_test)
```

```
0.9067796610169492
```

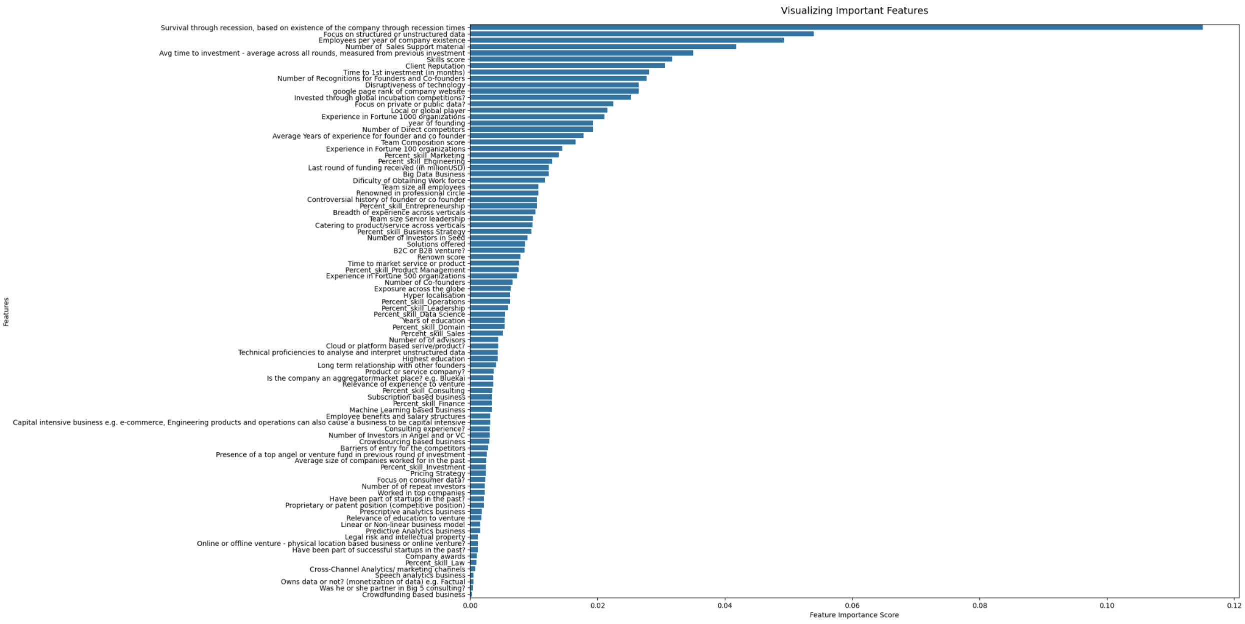
```
print(classification_report(y_test, model.predict(x_test_final)))
```

```
              precision    recall  f1-score   support

      Failed       0.83      0.90      0.86        39
     Success       0.95      0.91      0.93        79

    accuracy                           0.91       118
   macro avg       0.89      0.90      0.90       118
weighted avg       0.91      0.91      0.91       118
```

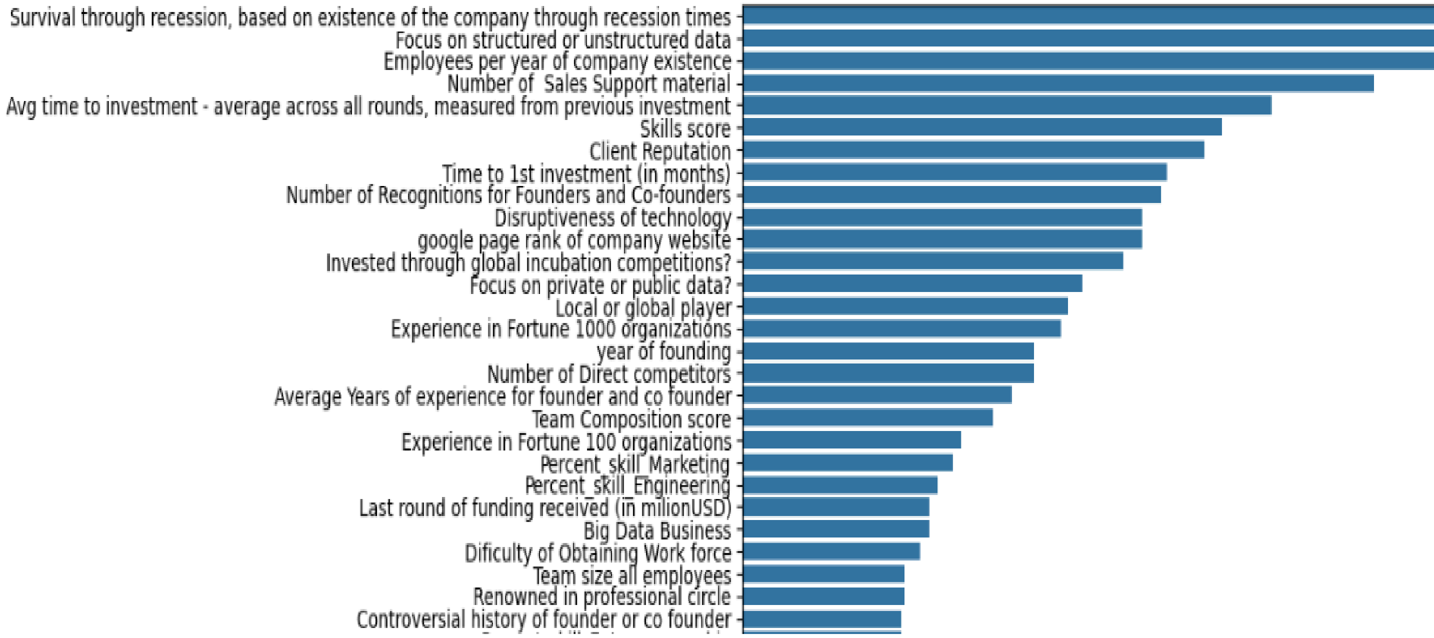

Random Forest Confusion Matrix
True label
Failed
Success
35
4
7
72
Failed
Success
Predicted label
70
60
50
40
30
20
10


True Positive Rate (Positive label: Success)
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate (Positive label: Success)
RandomForestClassifier (AUC = 0.96)

Now let's look at feature importance scores.



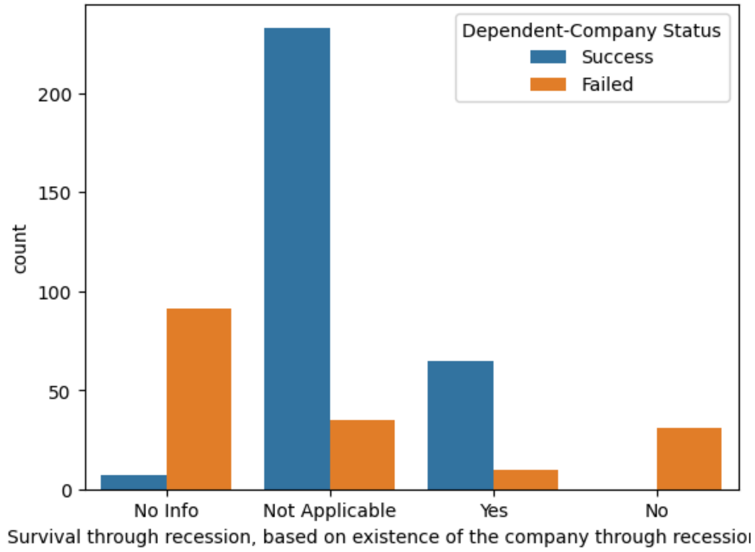
If we look closer:



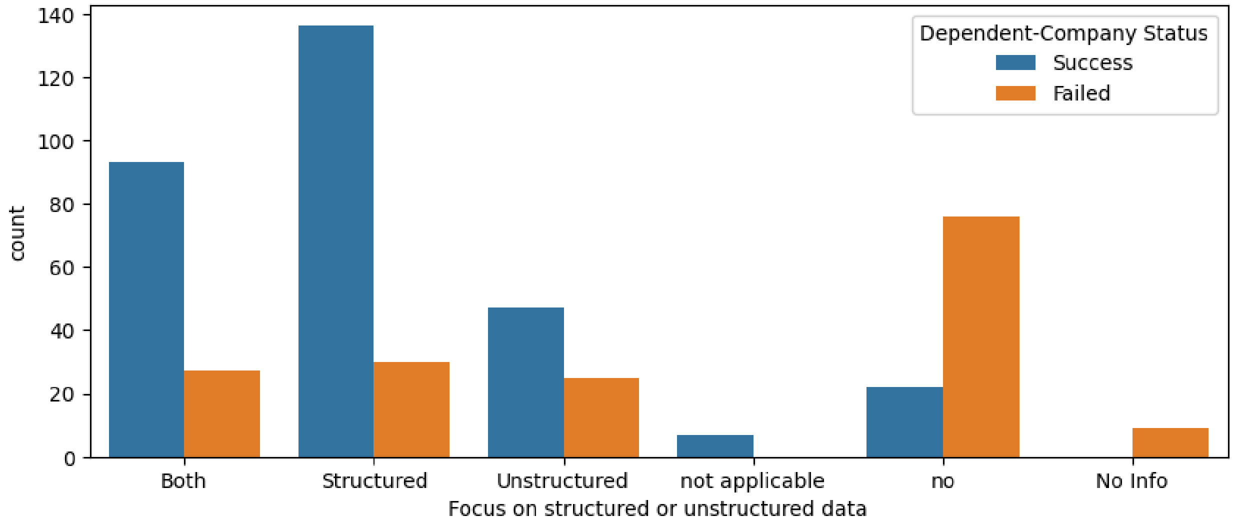
It’s important to note that I used ordinal encoder to encode categorical columns, which maybe worsened the model performance, but made the easily interpretable information extraction from importance scores possible.

Let’s see how top features with the highest scores relate to startup success.

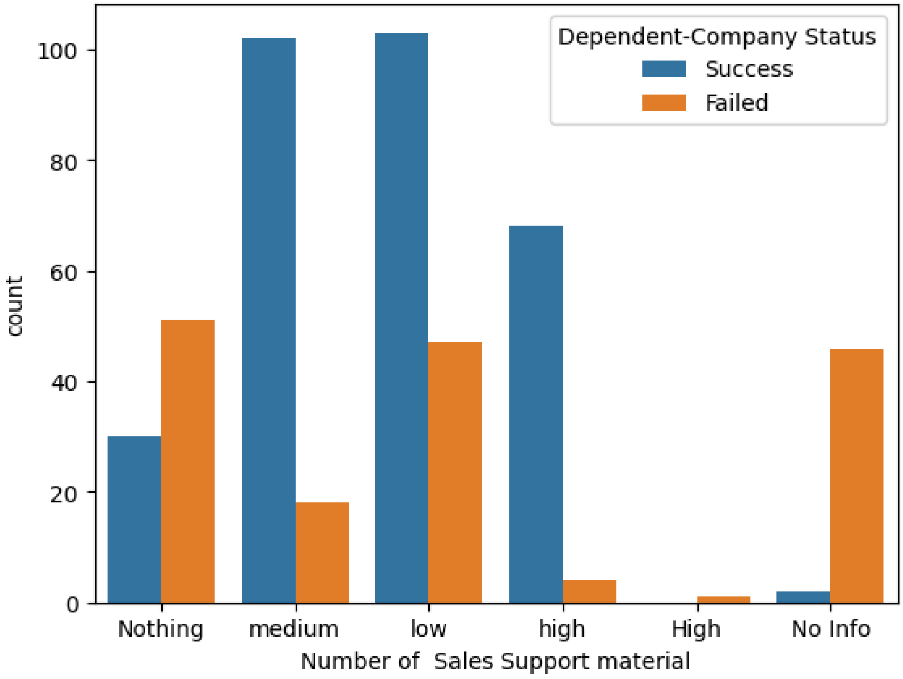
1. We see that startups, which survived through hard times, and the ones for which this metric is not applicable have higher chances of success, which is evident, because if the company didn’t survive, then there is no way it will succeed, which also means that this column leaks information about the label when the value is ‘no’, but I will keep it, by assuming it doesn’t leak too much.



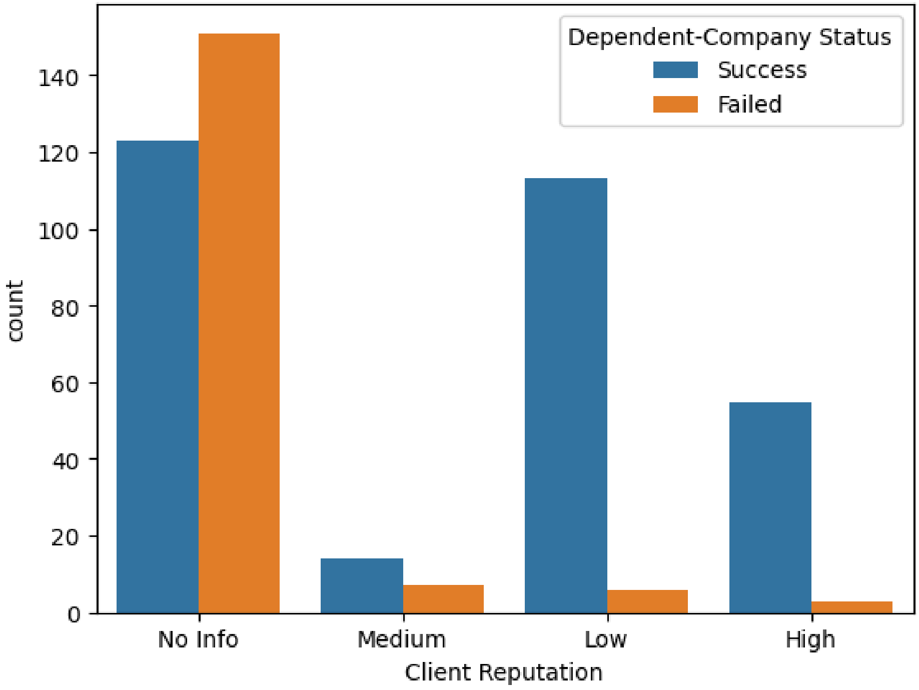
2. The startups which focus on structured and both structured and unstructured data have significantly higher chances of success.



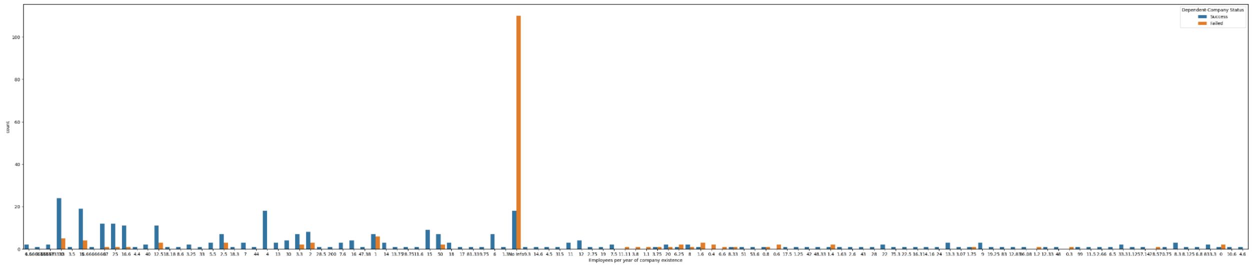
3. Here we see that companies with more sales support material have a higher chance of success.



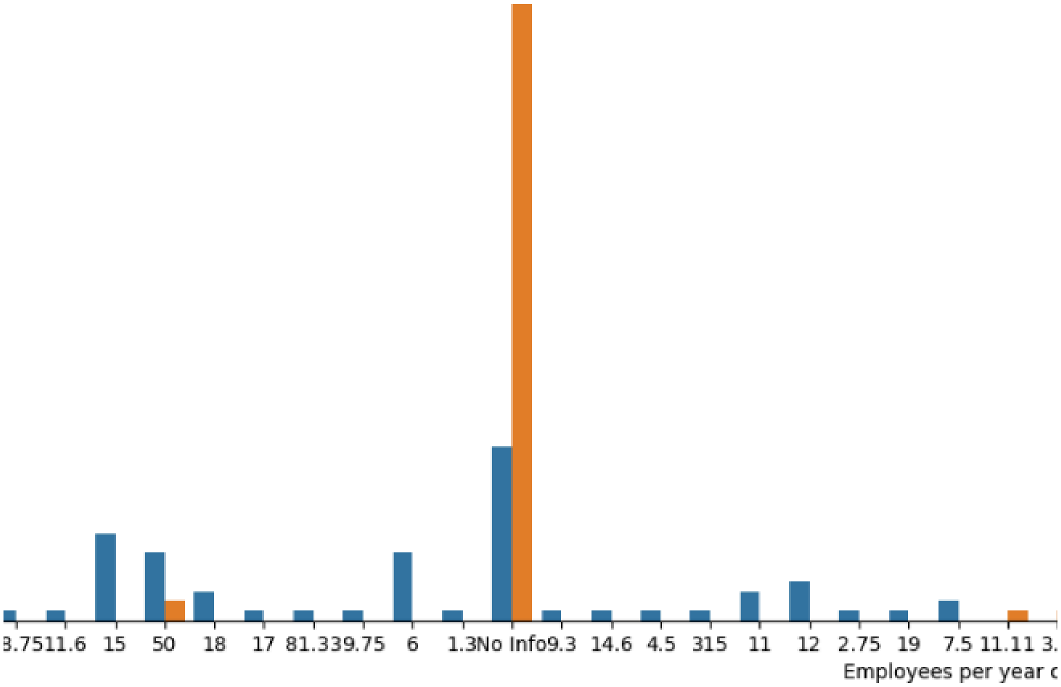
4. And here we see that there are too many ‘No Info’ values, which obscure the results. And because these ‘No Info’ values occupy a significant proportion of the dataset, by removing these rows we will lose significant information alongside it. So it would be better not to use this feature in the final model.



5. This is the ‘employees per year of company existence’ column.



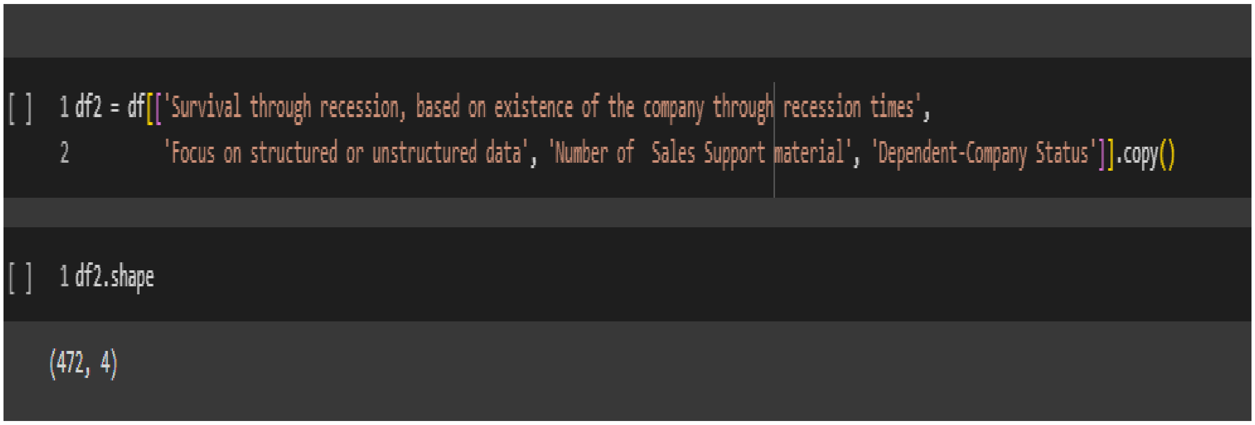
If we look closer, we will see that it’s mostly a numerical feature, but the ‘No Info’ value turns it into a categorical one. I decided to not use this column as well, instead of dropping the rows for the reasons discussed in 4.



6. And other numerical features with high importance scores are similar to the one shown above, so I will not use them too.

Building the second model, and training it only on important features.

Here is the final dataset, only with important features and the label.



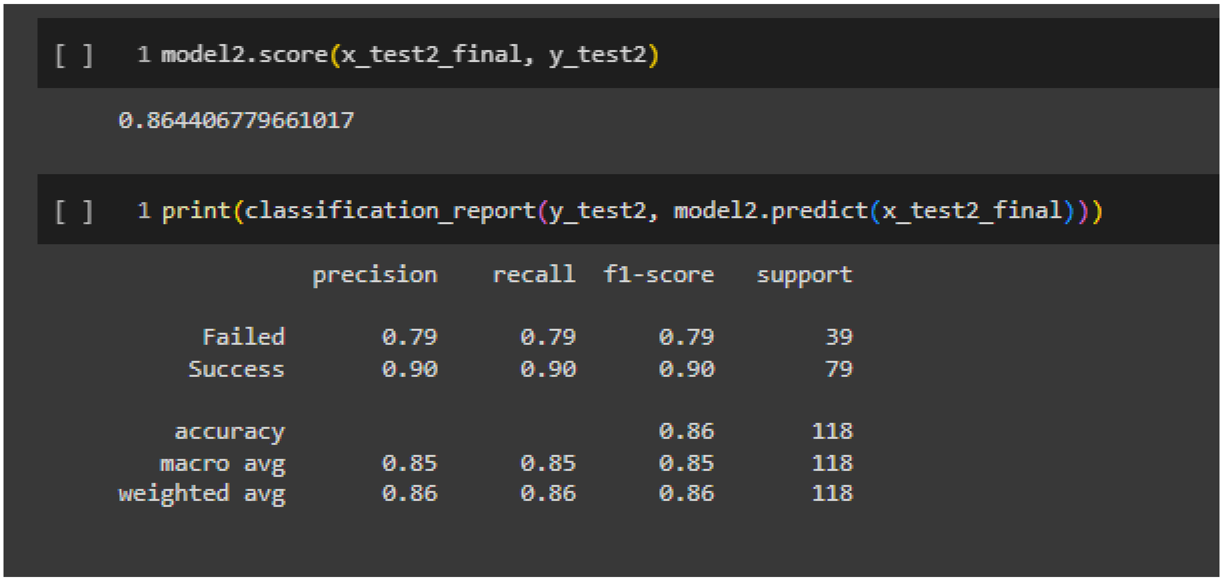
```
df2 = df[['Survival through recession, based on existence of the company through recession times',
         'Focus on structured or unstructured data', 'Number of  Sales Support material', 'Dependent-Company Status']].copy()
```

```
df2.shape
```

```
(472, 4)
```

This time I used one hot encoder to maximize the model's performance.

Evaluation results of the final model.

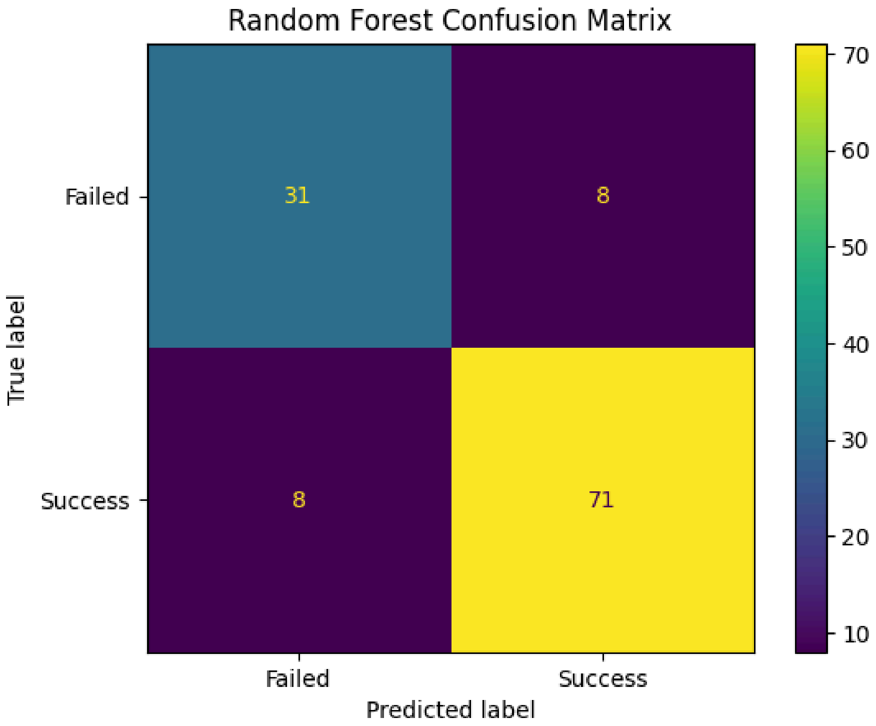


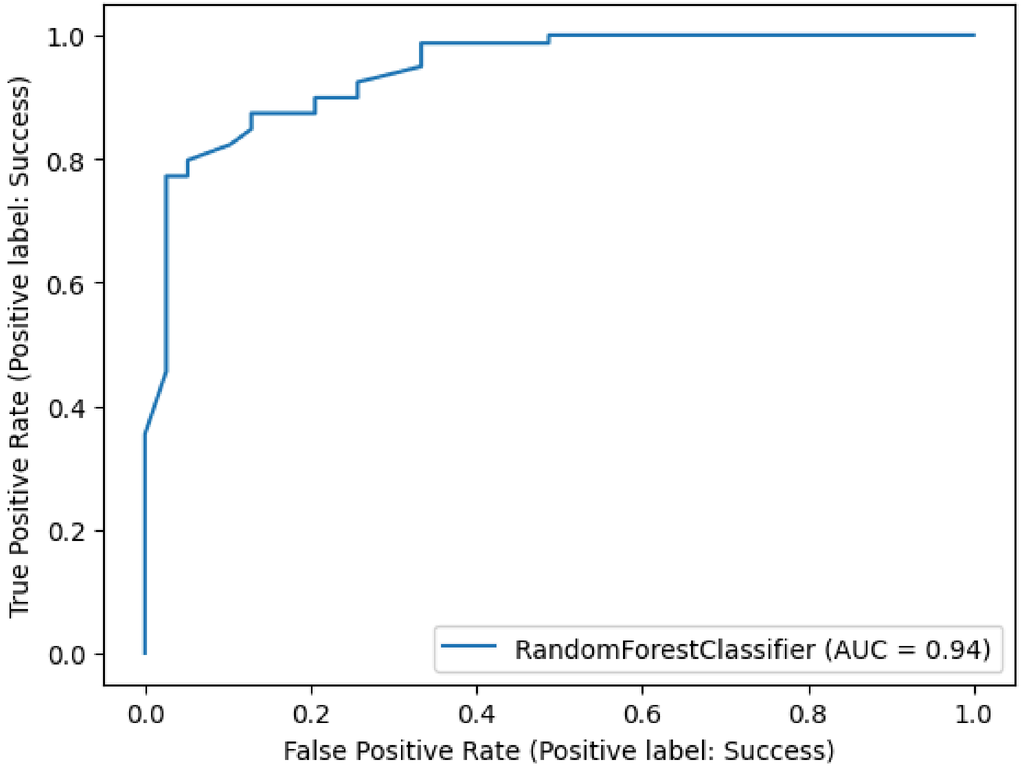
```
model2.score(x_test2_final, y_test2)
```

```
0.864406779661017
```

```
print(classification_report(y_test2, model2.predict(x_test2_final)))
```

```
              precision    recall  f1-score   support

      Failed       0.79      0.79      0.79        39
     Success       0.90      0.90      0.90        79

    accuracy                           0.86       118
   macro avg       0.85      0.85      0.85       118
weighted avg       0.86      0.86      0.86       118
```


Random Forest Confusion Matrix
True label
Failed
Success
31
8
8
71
Predicted label
Failed
Success
70
60
50
40
30
20
10


True Positive Rate (Positive label: Success)
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate (Positive label: Success)
RandomForestClassifier (AUC = 0.94)

Conclusions

We saw that most important features are:

1. "Survival through recession, based on existence of the company through recession times"
   If a startup survives through hard times, then it has a good chance of success.

2. "Focus on structured or unstructured data"
   Startups, which focus on structured data, or on both structured and unstructured, have higher chance of success.

3. 'Number of Sales Support material'
   And startups with more sales support material perform much better.