Detection of Oil Spills Using Machine Learning

Elias Al Alam [1]

Abstract

Oil spills can be extremely dangerous for aquatic animals. They can be dangerous if not cleaned and to do so we need to detect them first. In this study we will compare four classifiers to see which one is the best for detecting oil spills in the ocean.

1. Introduction

Ocean Oil spills happen more often than we realise. It can either happen by accident because of human mistakes, equipment failure, natural disasters, oil ships sinking or drilling operations gone wrong. On the other hand, oil spills can happen intentionally, which is the most dangerous part, because illegal dumping is not reported and just left in the ocean hoping no one would find it. This is usually done by greedy oil companies that dump millions of tons of drilling and oil waste in the ocean, especially in the arctic ocean, without any concern to the damages that this could lead to as long as it economically profitable for them. There are many ways to clean oil dumping but to do so we need to first locate them and that's where machine learning plays its part.

In this study we will be using four different machine learning algorithms to classify the same data set and then compare their accuracy and results in order to see which classifier in the best for oil spill detection.

2. Data-Set

Satellite images of the ocean surface with and without oil spills were taken and processed using a computer vision algorithm that transformed the images into sets of vectors that describes the content of each satellite image. In this study we will be using this processed data to try and detect the oil spills from the non oil spill areas.

In total the data consists of the 936 sets of variables and each variable has 51 features, of which 50 describe the variable and the 51st feature decides whether this variable is an oil spill or not (1 for oil spill and 0 for no oil spill). We can see in the table below the composition of our data-set.

Rows	Columns	Non-Oil Spills	Oil Spills
936	51	795	41

Table 1. Data-set Size and Composition.

Since the optimization of the code was done by trial and error there was no need for a test set, the data-set was split into 80% test set and 20% validation set as we can see in the table below:

	Training Set	Validation Set
Percentage (%)	80%	20%
Non-Oil Spills	716	179
Oil Spills	29	12

Table 2. Training and Validation Data-Sets.

3. Methodology

After getting the processed data from "***Kaggle***", we re-uploaded it on "***GitHub***" and used the "***GitHub***" raw data link to read it in our code. Then we split the data and now we need to define our classifiers. As mentioned before we used a total of 4 different classifiers. We will discuss below the reason why we chose each as well as the final parameters used in the code for each classifier.

3.1. Random Forest Classifier

Random Forest Classifier (RFC) is known for its robustness when it comes to outlier variables. Add to that **RFC**rarely ever over-fits, it's efficiency is top notch and it is known to have one of the highest accuracy rates amongst supervised machine learning classifiers.

The parameters chosen for the **Random Forest Classifier** are the following:

[*]Equal contribution [1]University of Lausanne, Lausanne, Vaud, Switzerland. Correspondence to: Elias Al Alam <Elias.AlAlam@unil.ch>.

This study was made possible thanks to The Beucler Lab UNIL.

Random Forest Classifier	
n_estimators	3048
max_depth	2024
random_state	1

Table 3. Random Forest Classifier Parameters.

3.2. Gradient Boosting Classifier

Gradient Boosting Classifier (GBC) is an other supervised classifier, we chose it for its accuracy that rivals the random forest however **GBC** is very flexible and can be used on data that has not been pre-processed.
The final parameters chosen are in the table below:

Gradient Boosting Classifier	
n_estimators	2048
learning_rate	0.1

Table 4. Gradient Boosting Classifier Parameters.

3.3. Decision Tree Classifier

The Decision Tree Classifier (DTC) is known for it's speed however it is a greedy classifier where most of the time the solution it reaches it rarely the optimal one but it is a local maxima so in other words this classifier values speed over accuracy, however it can also handle irrelevant attributes and missing data with ease.
The final most optimal parameters chosen for this classifier can be found in the table below:

Decision Tree Classifier	
max_depth	864
random_state	1

Table 5. Decision Tree Classifier Parameters.

3.4. Gaussian Naïve Bayes Classifier

Gaussian Naïve Bayes Classifier (GNB) is the 4th and last classifier we used in our study. **GNB** is also a supervised classifier, we chose it for its simplicity, speed and the fact that it can handle continuous and discrete date. It can also give very good results without the need for a huge training data.
For the parameters, there is only one which is the variable smoothing. The final most optimal value used is ***var_smoothing = 0.000000015***

4. Results

After fixing the parameters mostly by trial and error, the best results we managed to reach are the following:

Rank	Classifier	Accuracy
1	Random Forest	96.3%
2	Gradient Boosting	95.7%
3	Gaussian Naïve Bayes	94.7%
4	Decision Tree	93.6%

Table 6. Accuracy Scores of All the Classifiers Ranked.

As we can see in table 6, random forest classifier ranked first in terms of accuracy and the decision tree classifier ranked last. During the trial and error and parameter change, the gradient boosting was always the same or very close to the random forest. To visualise the answers, we plotted a confusion matrix for each of these classifiers.

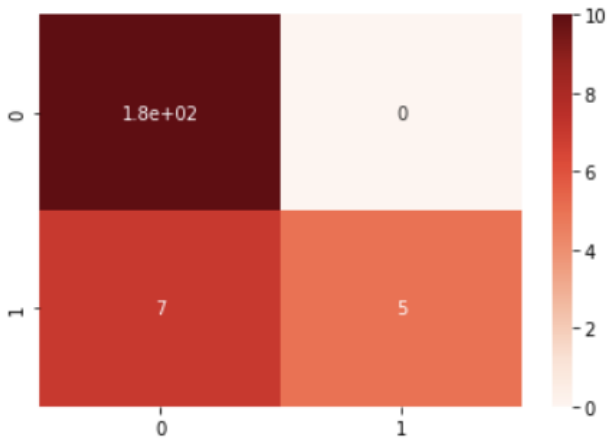

Figure 1. Confusion Matrix for the Random Forest Results.

As we can see, the random forest classifier managed to detect five out of 12 oil spills only but did not detect any false positives which is extremely good since that means we if this model tells us it detected an oil spill we can be 100% sure if we go to that spot we will find an oil spill.

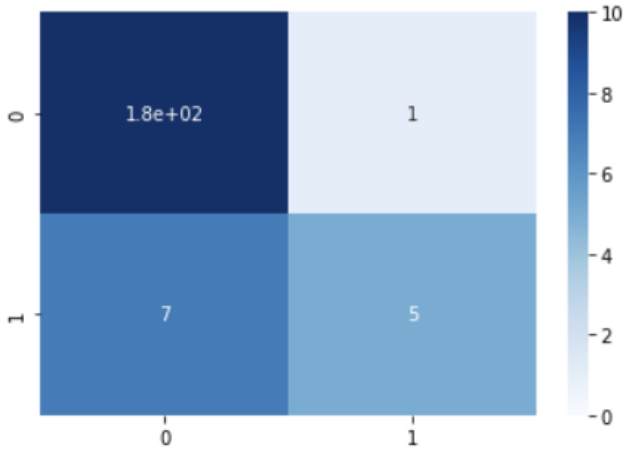

Figure 2. Confusion Matrix for the Gradient Boosting Results.

The gradient boosting classifier got almost always the same results as the RFC however it got 1 false positive which although may not seem like a lot it puts this classifier at a great disadvantage when compared to the random forest which has zero false positives.

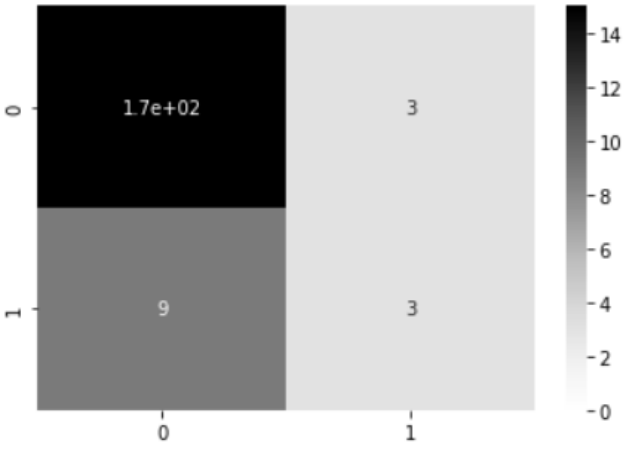

Figure 3. Confusion Matrix for the Decision Tree Results.

The decision tree was always the worst of all as we can see in ”Figure 3”, however when we decreased the test set to 70% instead of 80% it performed better than both the RFC and the GBC but it’s values were still not very good, so it is not that it performed better with lower test set, it is that the other classifiers performed worse while the decision tree was not effected by that decrease in the test set size.

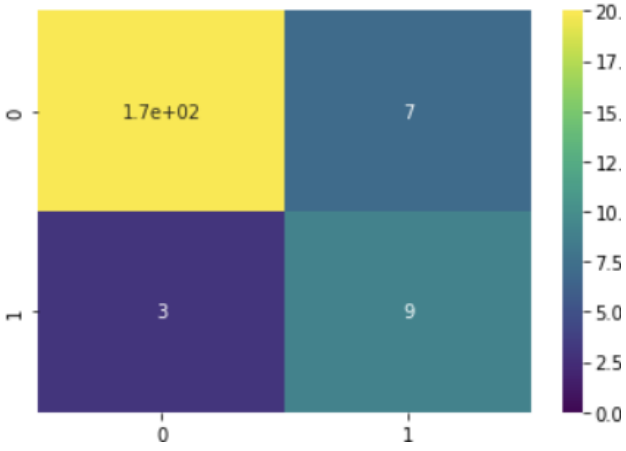

Figure 4. Confusion Matrix for the Gaussian Naïve Bayes Results.

The guassian naïve bayes classifier, as we can see in ”Figure 4” managed to detect the most oil spills out of all the classifiers, detecting nine out of 12 oil spills. This high number of oil spills detection came with a downside which is a larger number of false positives. The most we managed to detect with this classifier is ten oil spills out of twelve but that came with fifteen false positives.

5. Limitations

There are two main limitations for this study:

- Although the accuracy is high, this is mainly due to the high number of non-oil spills when compared to the number of oil-spills which means even if our classifier took everything as non-oil spill we would still have an accuracy greater than 80% which makes the accuracy score less meaningful. A quick solution for this would be to take part of the non-oil spill variables which makes the data-set more evenly distributed which makes the accuracy score more representative of the results we have.
- Since the images where pre-processed with a program or algorithm which we did not manage to find, this makes our data-set non reproducible. This can be solved by trying to get the original images and process them ourselves and then repeat the classification study on the new data-set we generated.

6. Conclusion

In conclusion, if we find that the time and cost to check the oil dumping sites is not high and we can afford it, i would go with **Gaussian Naïve Bayes Classification** since it managed to detect the most oil spills out of all four classifiers.

On the other hand, if we can't afford to check oil dumping sites that might turn out to not there, i would go with the **Random Forest Classifier** since, although it managed to detect only five out twelve oil spills, it did not detect any false positives which makes it the best classifier if we are on a tight budget.
It is also worth noting that in the code, we did a *shap explainer* at the end to try and see what are the main features with the highest impact on the results in order to redo the study and try and optimize the results even more, however the result did not seem promising and so we decided to keep it as reference but not optimise based on it.

Code and Data-Set Link

- Raw Code Link: Click Here.
- Jupiter Code with Results Link: Click Here.
- Data Set Link: Click Here.

Acknowledgements

I would like to thank **Dr. Tom Beucler** for being the most understanding and down to earth doctor I have ever met. I would also like to thank **Sir Milton Salvador Gomez Delgadillo** for being active on discord 24/7, always ready to help and always there when we needed him. Lastly, I would like to thank my peers for their feedback and kind review of my project which helped me greatly improve both my code and report.

Bibliography and Citations

- Banoula, M. (2022) Naive Bayes classifier - machine learning [updated]: Simplilearn, Simplilearn.com. Simplilearn. Available at: https://www.simplilearn.com/tutorials/machine-learning-tutorial/naive-bayes-classifier (Accessed: December 18, 2022).
- Berezovsky, O. (2020) The decision tree classifier – an overview, Logic20/20. Logic20/20 Consulting. Available at: https://www.logic2020.com/insight/tactical/decision-tree-classifieroverview#:~:text=Advantages%20of%20decision%20trees,deliver%20the%20high%20accuracy%20score. (Accessed: December 18, 2022).
- Kurama, V. (2021) Gradient boosting for classification, Paperspace Blog. Paperspace Blog. Available at: https://blog.paperspace.com/gradient-boosting-for-classification/ (Accessed: December 18, 2022).
- Malayvyas (2022) Oil_Spill with 97% accuracy, Kaggle. Kaggle. Available at: https://www.kaggle.com/code/malayvyas/oil-spill-with-97-accuracy (Accessed: December 18, 2022).
- Northam, J. (2020) The oil spill from Russian nickel mine is moving toward the Arctic Ocean, NPR. NPR. Available at: https://www.npr.org/2020/06/16/878852931/the-oil-spill-from-russian-nickel\-mine-is-moving-towards-the-arctic-ocean#:~:text=There%20is%20an%20environmental%20disaster,moving%20towards%20the%20\Arctic%20Ocean. (Accessed: December 18, 2022).
- Oil spills: A major Marine Ecosystem Threat (no date) National Oceanic and Atmospheric Administration. Available at: https://www.noaa.gov/explainers/oil-spills-major-marine-ecosystem-threat#:~:text=Oil%20spills%20that%20happen%20in,equipment%20breaking (Accessed: December 18, 2022).
- Rastogi, S. (2022) Oil spill classification, Kaggle. Available at: https://www.kaggle.com/datasets/sudhanshu2198/oil-spill-detection?resource=download (Accessed: December 18, 2022).
- Team, T.A.I. (2020) Why choose Random Forest and not decision trees, Towards AI. Available at: https://towardsai.net/p/machine-learning/why-choose-random-forest-and-not-\decision-trees(Accessed: December 18, 2022).