AG News Topic Classification

CSML 1010 - Winter 2020 - Group 20

Tony Lee, Viswesh Krishnamurthy

PROBLEM SELECTION & DEFINITION

- A text classification problem was chosen from the website https://datasets.quantumstat.com/. We chose the AG News corpus dataset
- The goal of this project is to develop a text classifier model that can accept a news 'headline' and 'content' of the news to classify the news article into one of the 4 following categories
  - World (coded as 1)
  - Sports (2)
  - Business (3)
  - Sci/Tech (4)

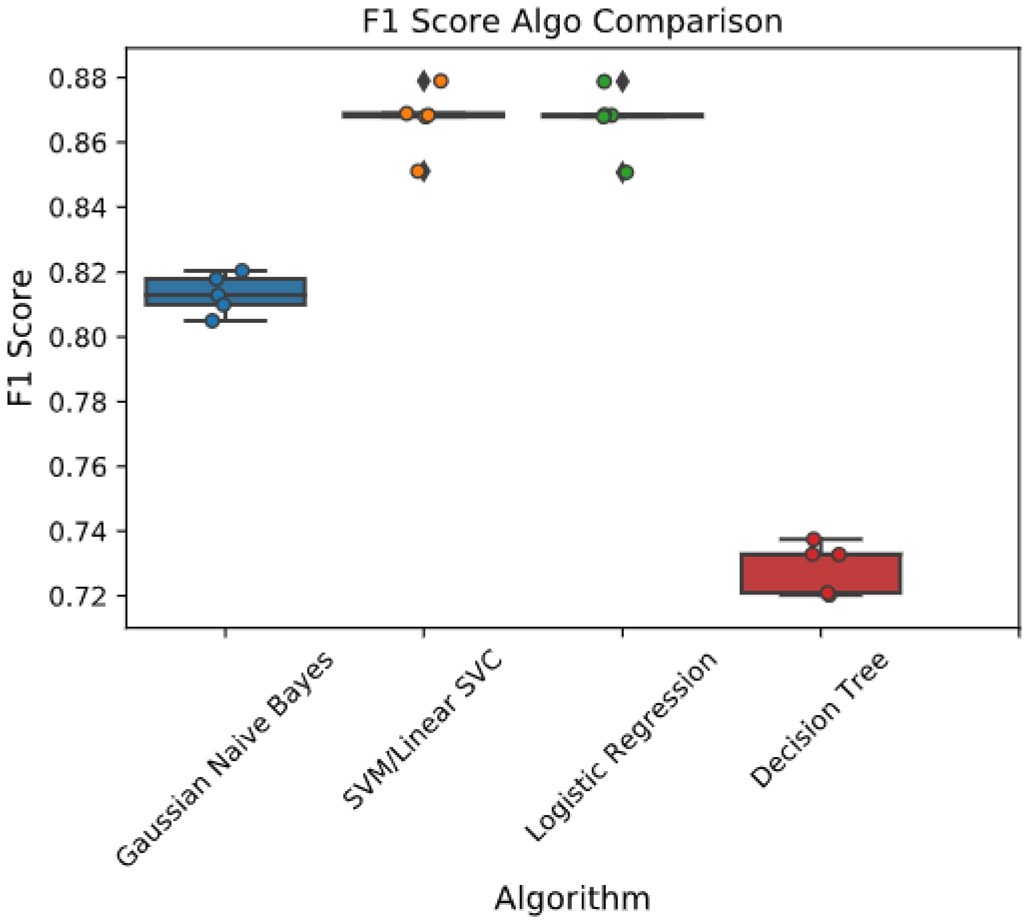
YORK U UNIVERSITÉ UNIVERSITY

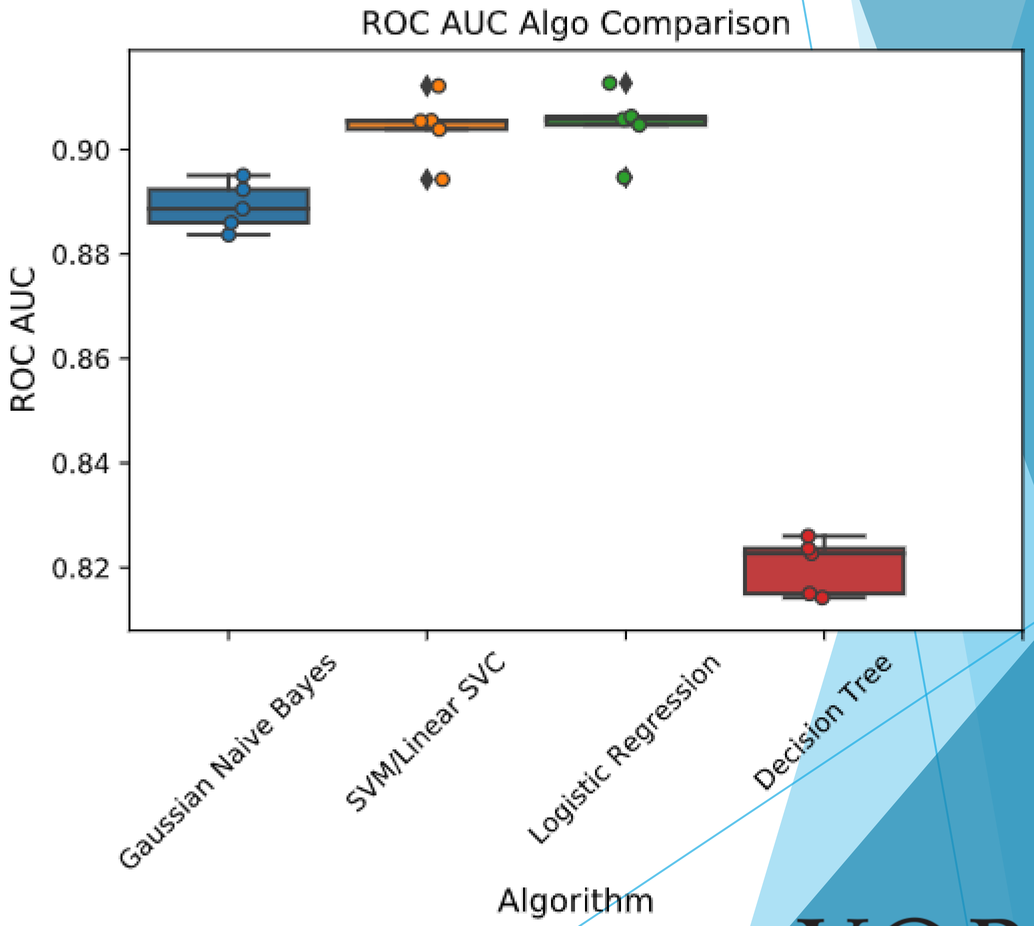
FINAL PROJECT PRESENTATION – 22nd May 2020

CROSS VALIDATION

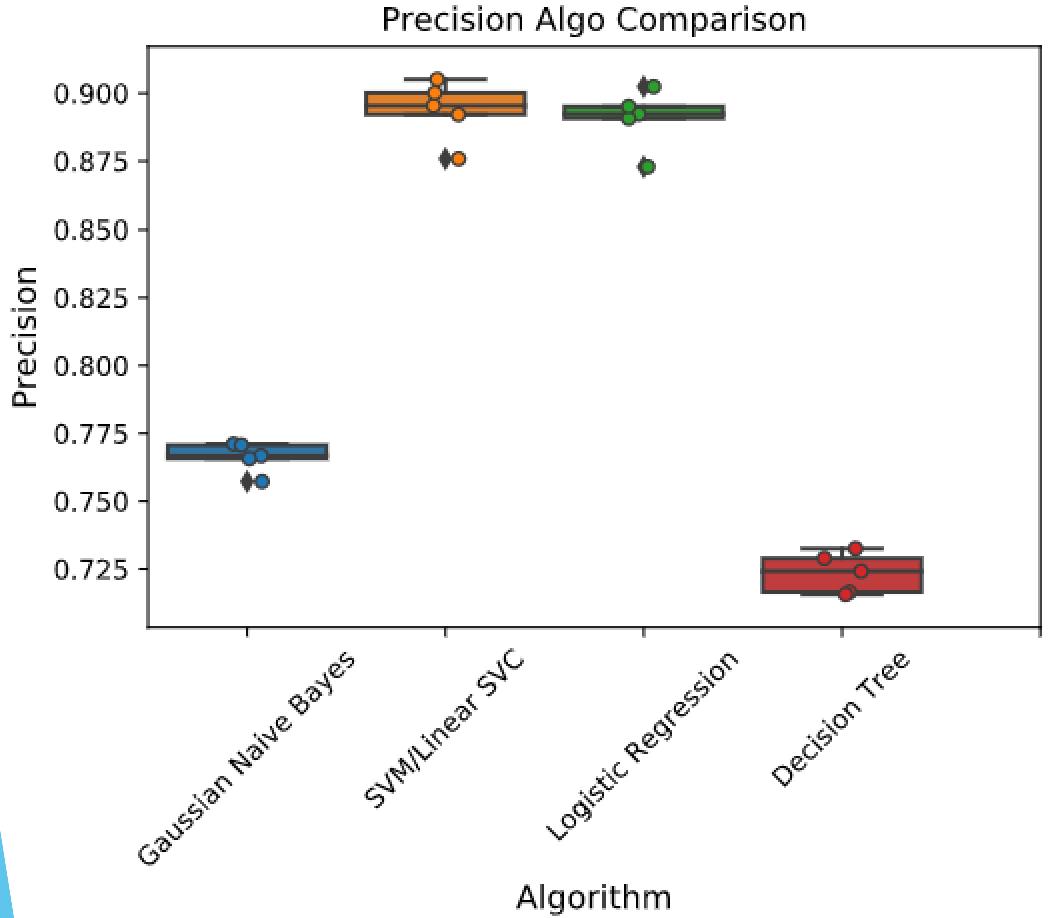
- We ran a 5 fold cross validation on the training dataset and re-ran all the models
- The models on the "solo" runs were trained on the 120,000 instance training set and tested on a separate 7600 instance test set
- The results of cross validation closely follow the previous "solo" runs
- Logistic Regression & SVM compare very closely in terms of the metrics. However, SVM was very resource intensive

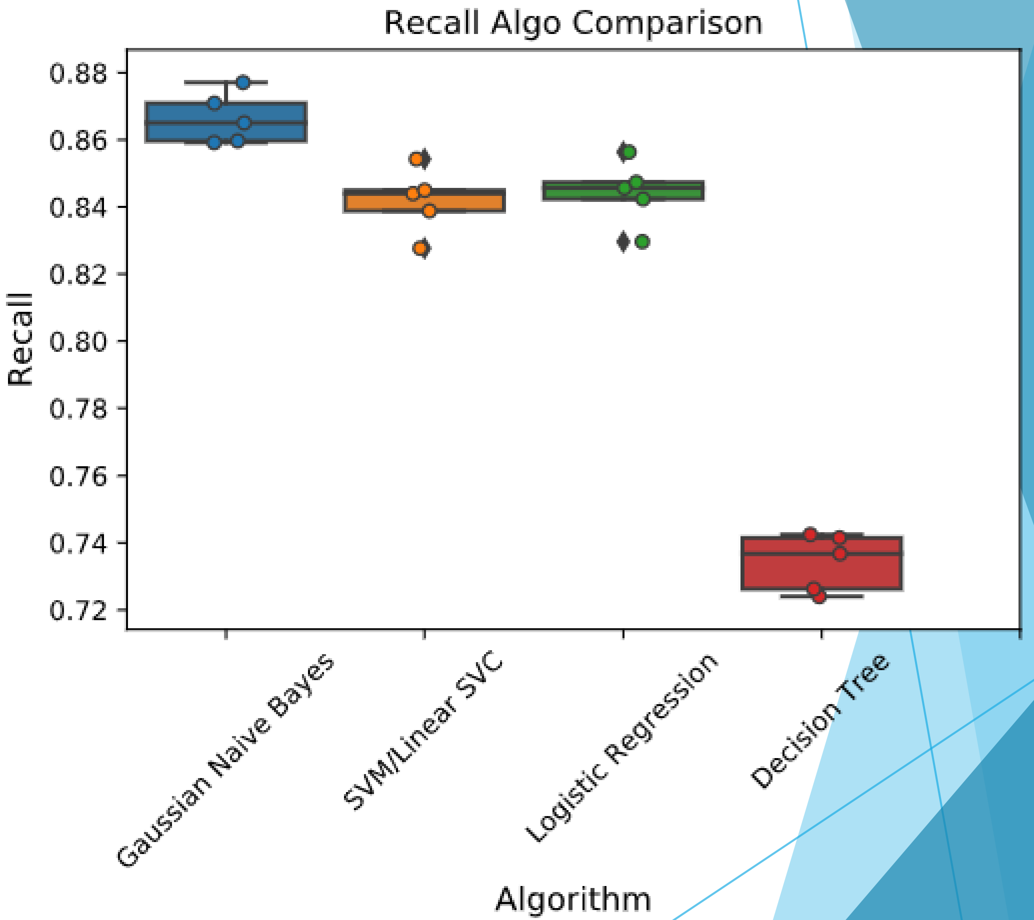
CROSS VALIDATION – COMPARISON OF RESULTS



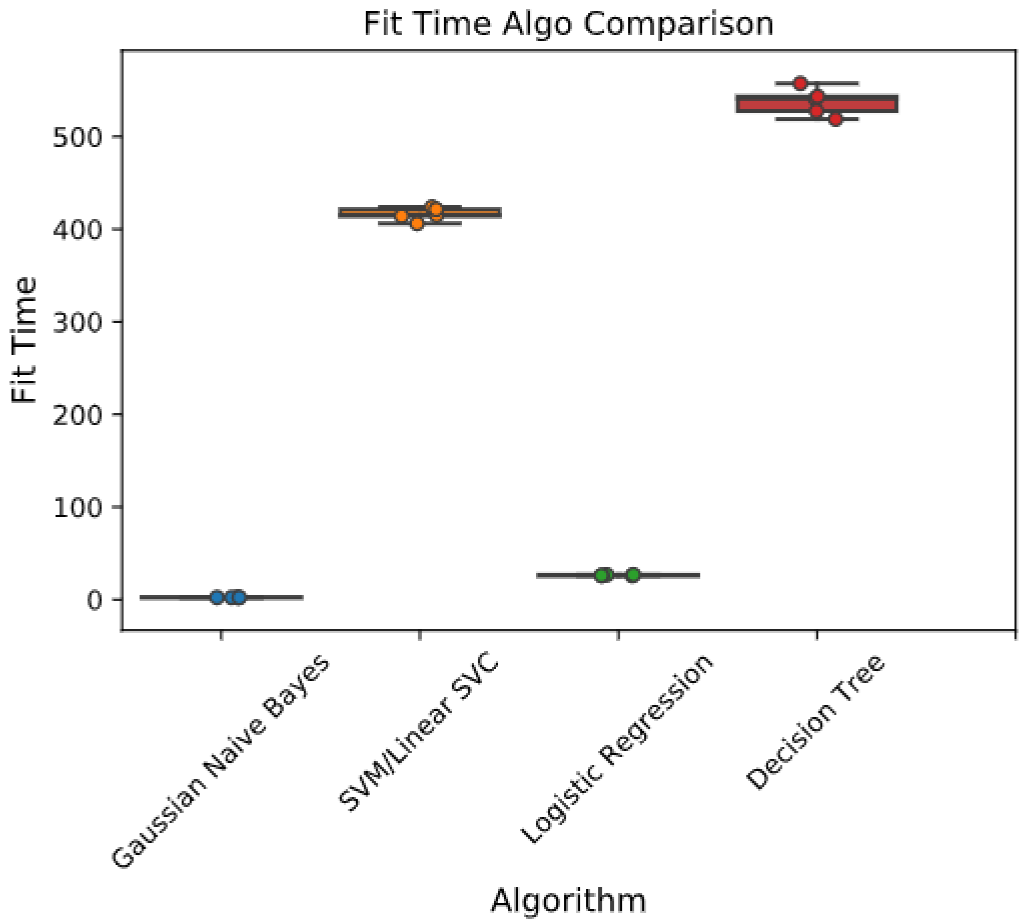


CROSS VALIDATION – COMPARISON OF RESULTS



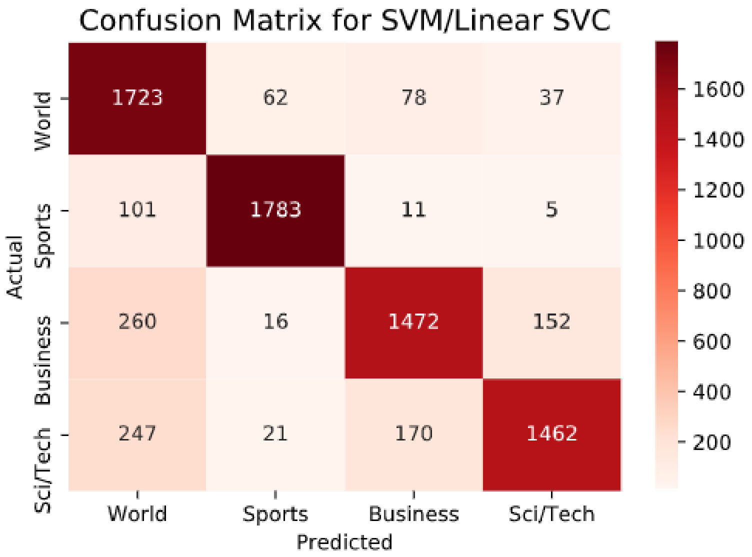


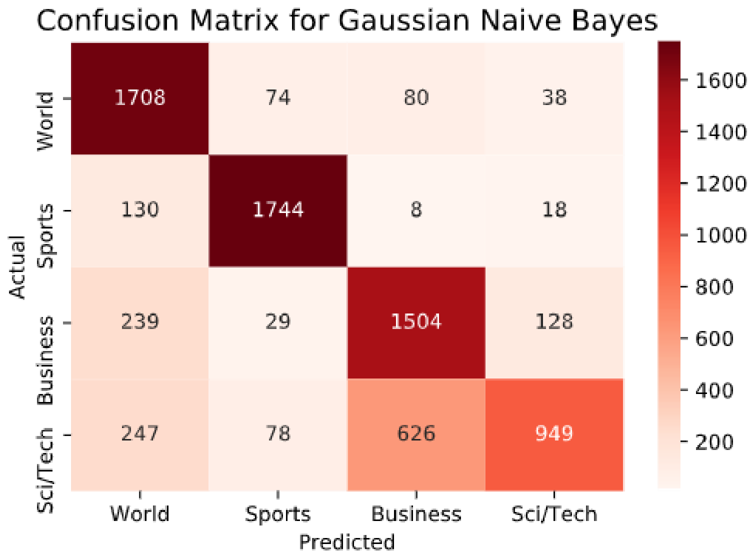
CROSS VALIDATION – COMPARISON OF RESULTS

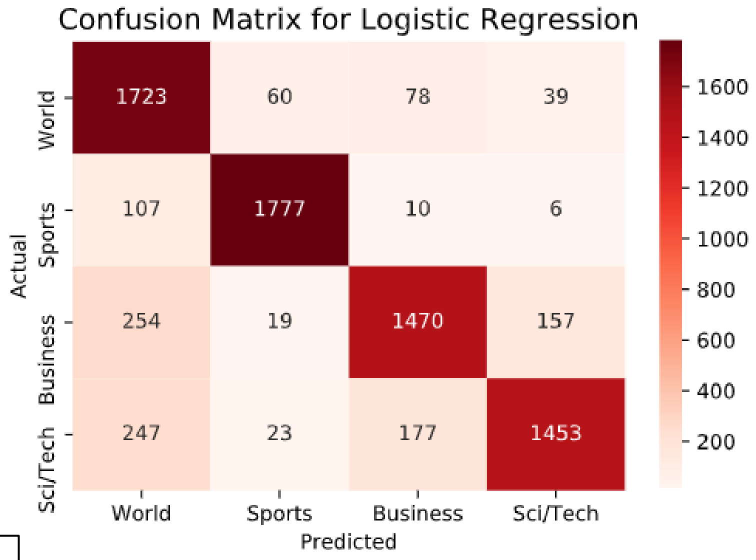


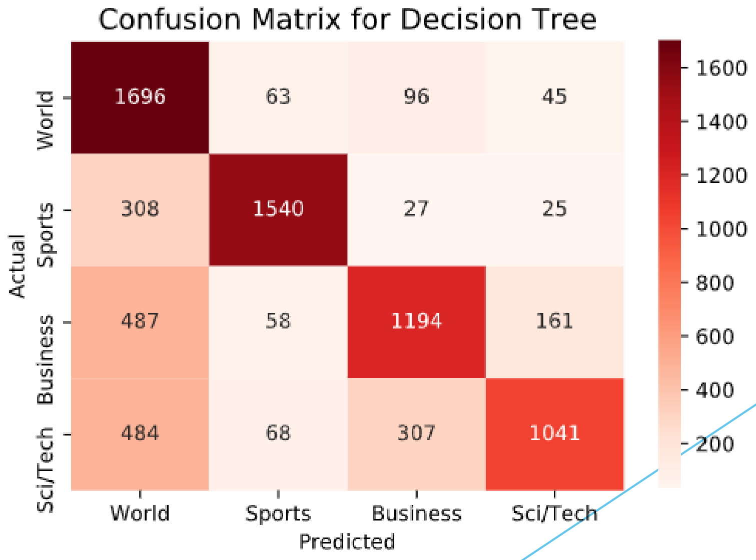
- Based on the previous metrics and comparison, Logistic Regression emerges as the model of choice

CROSS VALIDATION – COMPARISON OF RESULTS, CONFUSION MATRIX

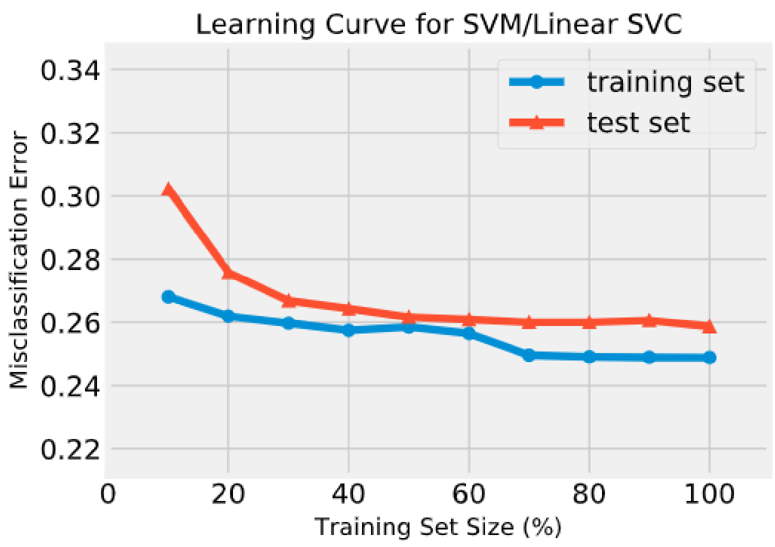


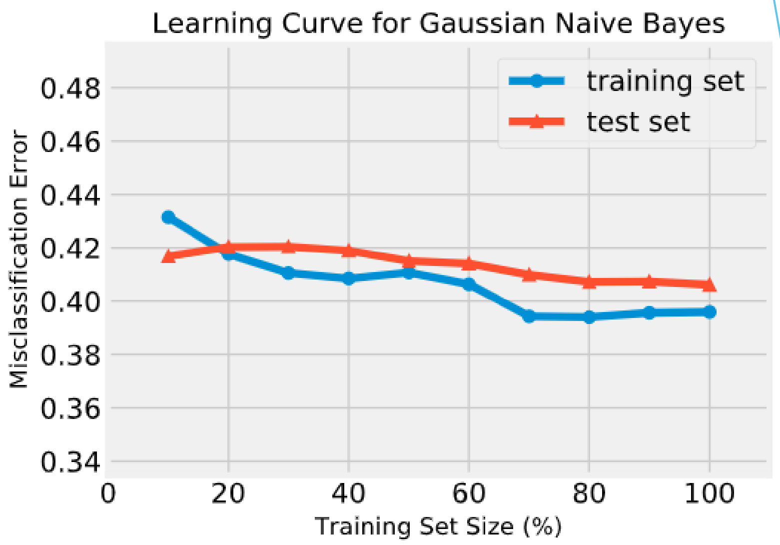


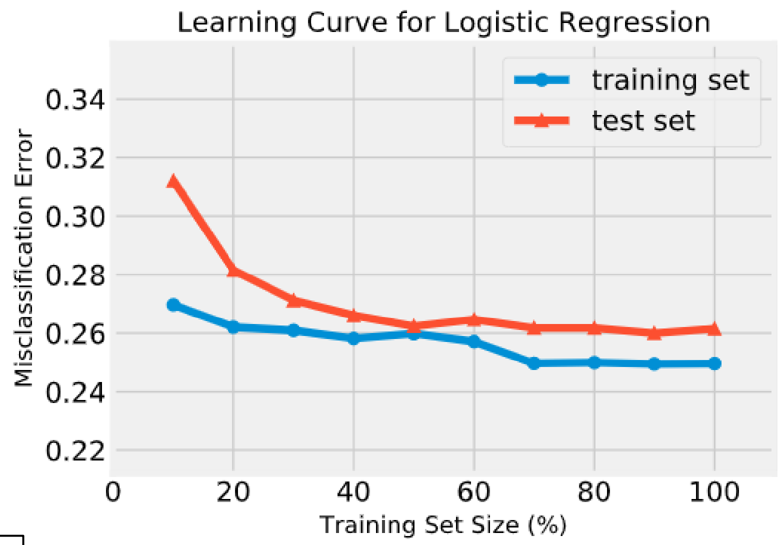


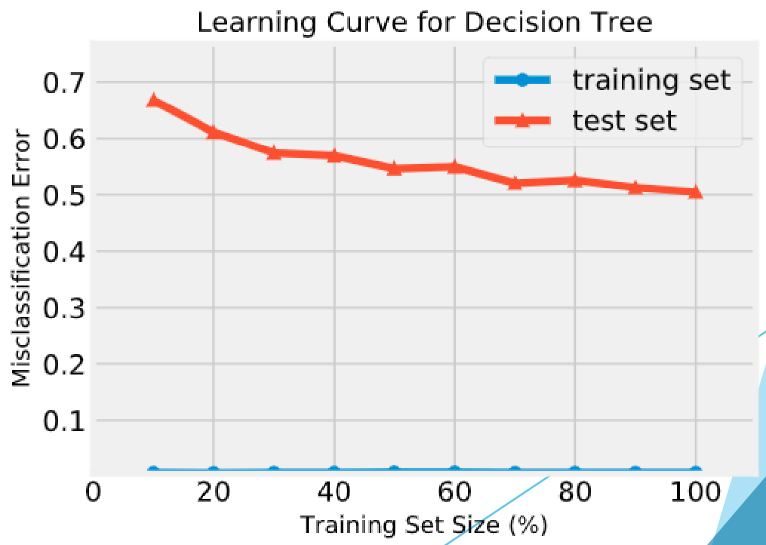


CROSS VALIDATION – COMPARISON OF RESULTS, LEARNING CURVE







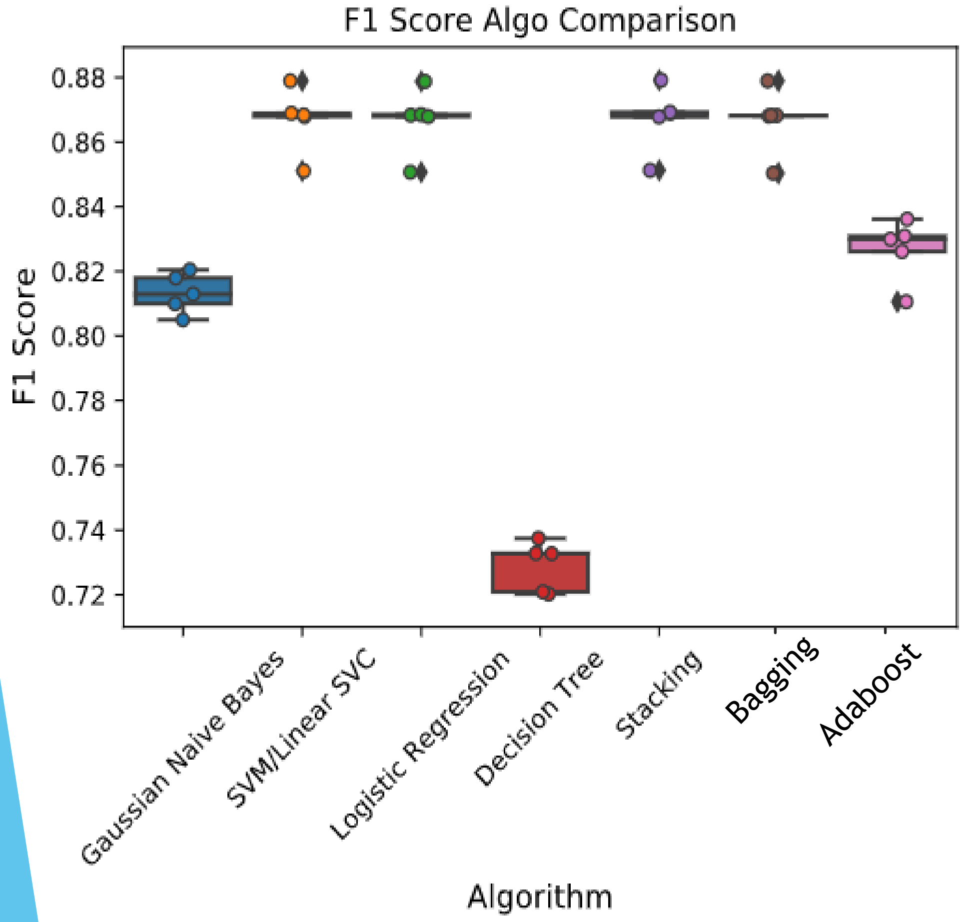


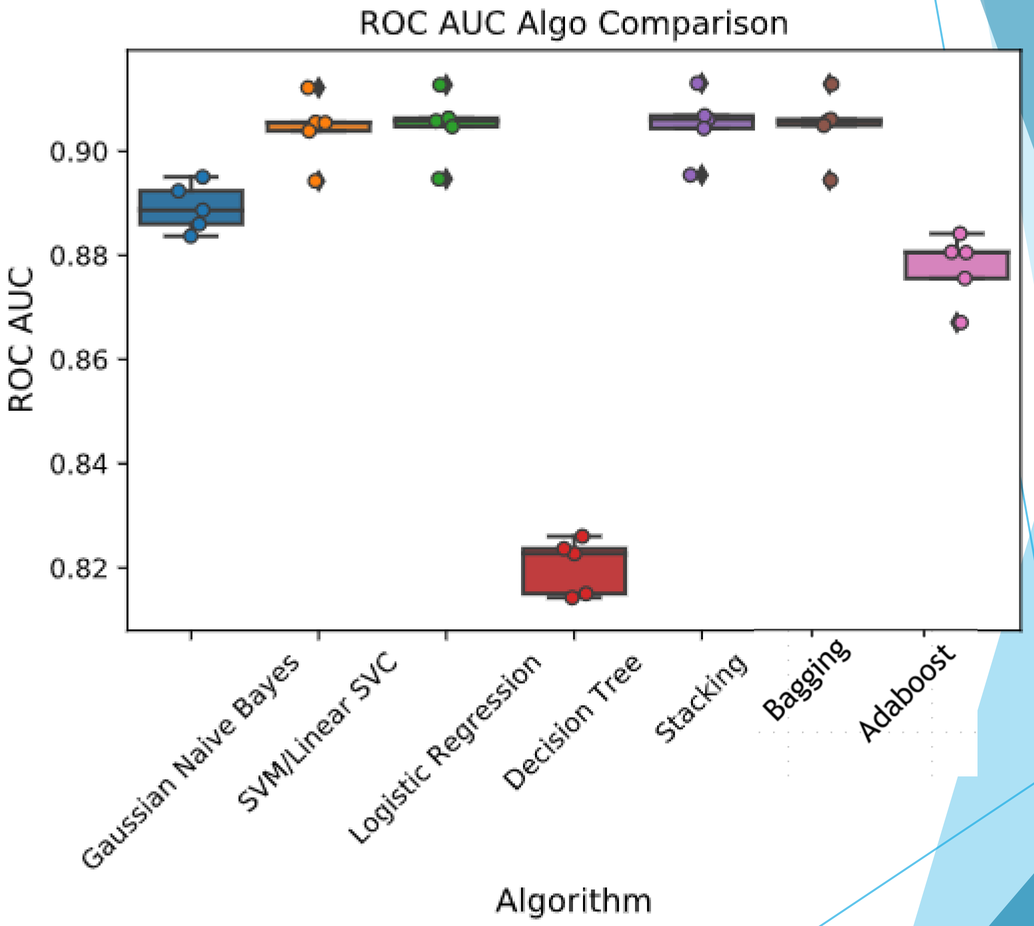
YORK
UNIVERSITÉ
UNIVERSITY

ENSEMBLE METHODS

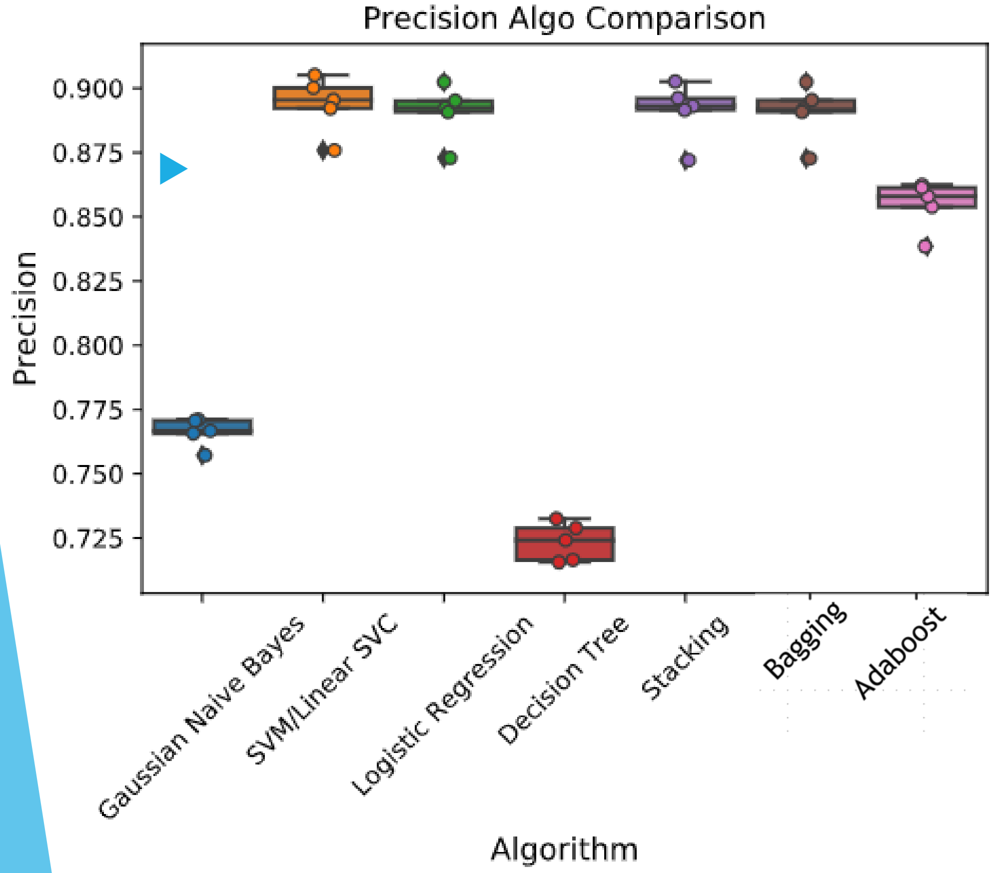
- The following ensemble methods were used
  - Bagging
    - The base estimator was chosen to be Logistic Regression
  - Stacking
    - The initial estimators in stacking were chosen to be Naïve bayes and SVM. The meta learner was Logistic Regression
  - Boosting
    - An Adaboost classifier was used for boosting in this iteration.

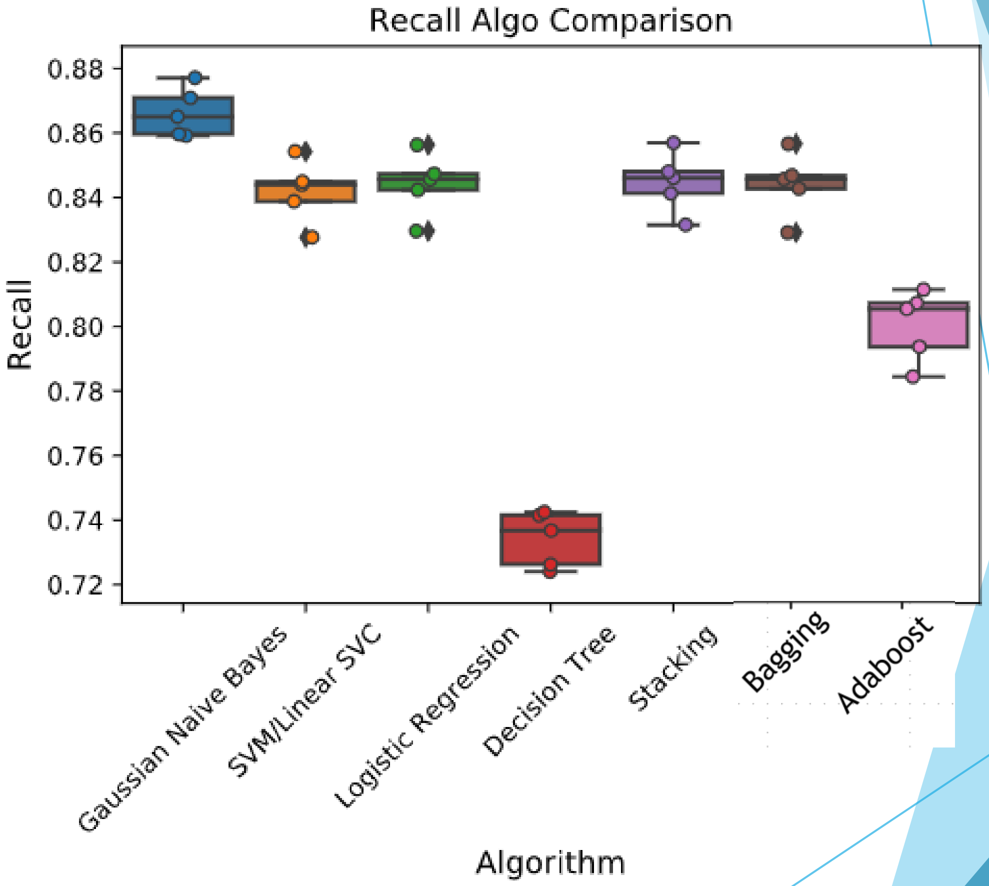
ENSEMBLE METHODS vs OTHERS



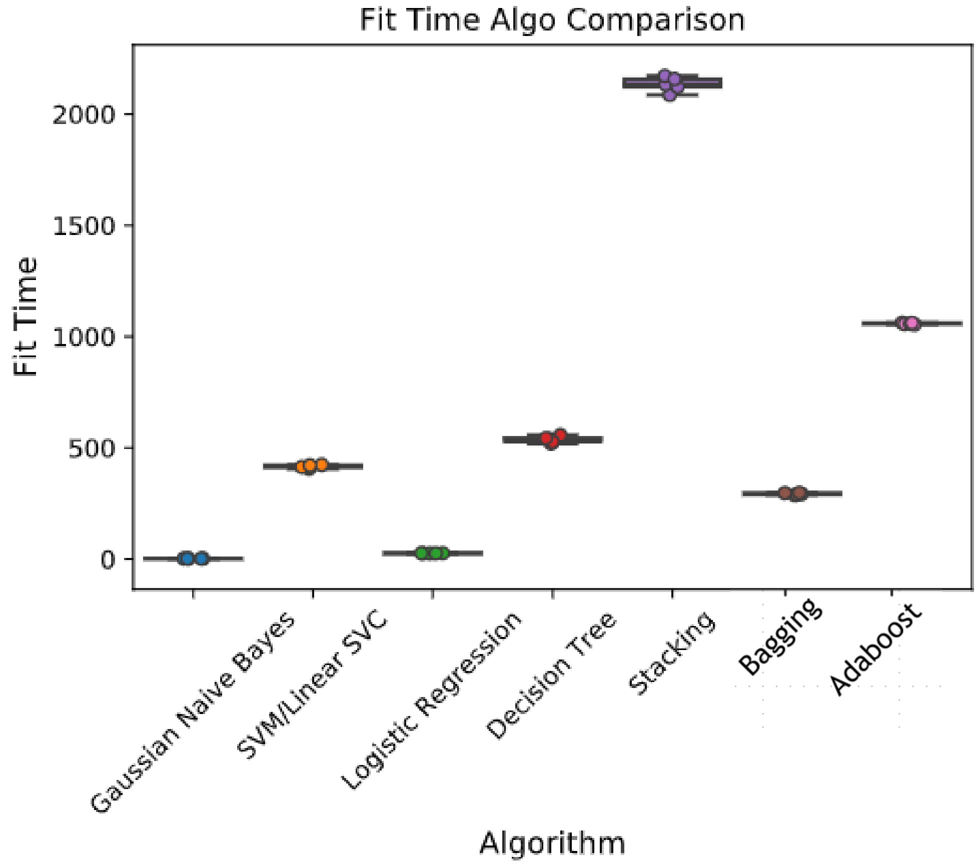


ENSEMBLE METHODS vs OTHERS





ENSEMBLE METHODS vs OTHERS



- Based on the metrics seen so far, it is clear that “Bagging” ensemble method is the algorithm of choice
- Stacking and Bagging perform almost similarly in terms of F1 Score, ROC AUC, Precision & Recall
- However, in terms of Fit time, stacking took almost 25 minutes to fit, while bagging took just over 3.5 mins

YORK UNIVERSITÉ UNIVERSITY

HYPERPARAMETER TUNING

- We performed hyperparameter tuning with Grid Search

- For Gaussian Naïve Bayes, ‘var_smoothing’ parameter was tuned

```
Best Score:   0.801674851179991
Best Params:  {'estimator__var_smoothing': 0.001}
```

- For Logistic Regression, ‘penalty’, ‘C’, ‘Solver’ hyperparameters were tuned

```
Best score: 0.8320745341017949
Best Params: {'estimator__C': 0.01, 'estimator__penalty': 'l2', 'estimator__s
olver': 'newton-cg'}
```

- For SVM, estimator penalty, tolerance, loss & C parameters were tuned

```
Best score: 0.8440514791910111
Best Params: {'estimator__C': 0.001, 'estimator__loss': 'squared_hinge', 'est
imator__penalty': 'l2', 'estimator__tol': 1e-08}
```

- For Decision trees, ‘splitter’, ‘min_samples_split’ were tuned

```
Best score: 0.6746835492233547
Best Params: {'estimator__min_samples_split': 9, 'estimator__splitter': 'bes
t'}
```

Clearly, the best params are pretty much the default ones.

The algorithms are already pretty smart about the defaults or can calculate them. Tuning these hyper-parameters might actually cause overfitting

YORK U UNIVERSITÉ UNIVERSITY

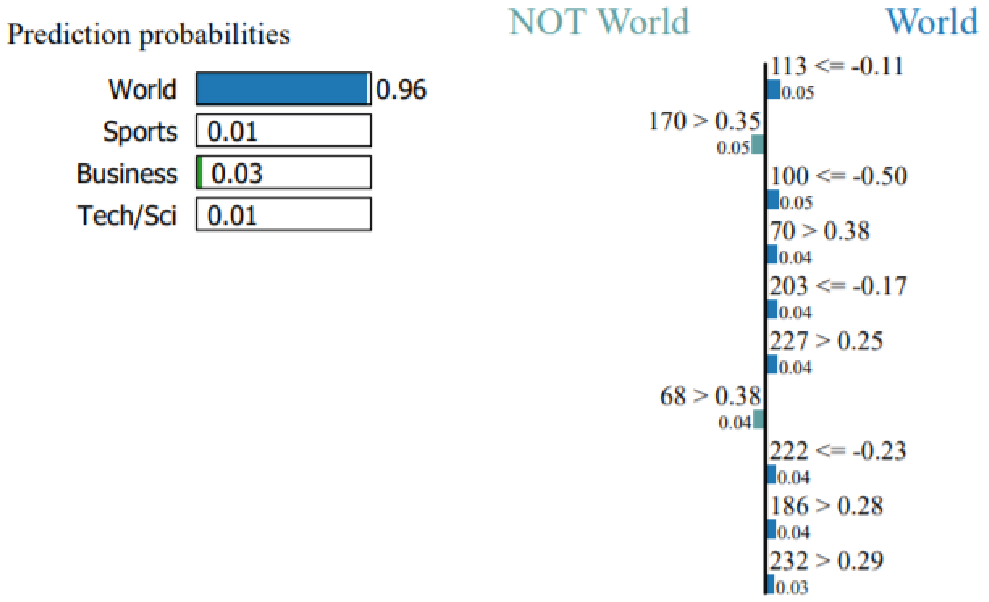
MODEL INTERPRETABILITY - LIME

- LIME stands for Locally Interpretable Model Agnostic Explanations

- Instantiate Explainer explains a specific instance in the dataset. The 34th document in this example

```
Document id: 34
Predicted class = World
True class = World
```

- A “text only” explainer can be visualized as seen below



RESULTS & DISCUSSION

- If this is deployed in production, the model of choice is Logistic Regression. SVM, Logistic Regression, Stacking & Bagging are very comparable in terms of F1 Score, ROC AUC, Precision & Recall. However, Logistic Regression is orders of magnitude faster with a very low fit time.

- Logistic Regression achieves, f1 Score of 92%

- Hyperparameter Tuning – The results of hyperparameter tuning suggest that default parameters are the best and therefore, tuning them further will result in overfitting

- Model Interpretability – Advanced word embeddings like Word2Vec inherently make the model "black box". Though LIME was used, it must be noted that in the "text only" explainer, the features shown are just numbers. They have gone through previous processing and can no longer be traced back to their original form. Simpler vectorization techniques like TF-IDF still retain the original features and can therefore allow LIME to explain them

YORK
UNIVERSITÉ
UNIVERSITY

NEXT STEPS

- Use other word embedding techniques like BERT, EIMO
- Use more advanced models such as CNN, Deep Learning
- Look at ways to deploy the model in a real world business scenario. For eg, a mobile news service that can get news from multiple sources and present them under different classes

REFERENCES

- References - Code sample sources disclaimer:
- # Code for this project is either directly from (with some modification),
- # or inspired by, but not limited to the following sources:
- # - Respective documentation and examples from each used API's doc/guide website
- # - Kelly Epley Naive Bayes:
- # https://towardsdatascience.com/naive-bayes-document-classification-in-python-e33ff50f937e
- # - MLWhiz's excellent blogs about text classification and NLP:
- # https://mlwhiz.com/blog/2018/12/17/text_classification/
- # https://mlwhiz.com/blog/2019/01/17/deeplearning_nlp_preprocess/
- # https://mlwhiz.com/blog/2019/02/08/deeplearning_nlp_conventional_methods/
- # https://www.kaggle.com/mlwhiz/conventional-methods-for-quora-classification/
- # - Christof Henkel preprocessing:
- # https://www.kaggle.com/christofhenkel/how-to-preprocessing-when-using-embeddings
- # - datanizing GmbH:
- # https://medium.com/@datanizing/modern-text-mining-with-python-part-1-of-5-introduction-cleaning-and-linguistics-647f9ec85b6a
- # - Datacamp wordcloud:
- # https://www.datacamp.com/community/tutorials/wordcloud-python
- # - Seaborn Pydata tutorials:
- # https://seaborn.pydata.org/introduction.html#intro-plot-customization
- # - Dipanjan S's tutorials:
- # https://github.com/dipanjanS
- # - Analytics Vidhya:
- # https://www.analyticsvidhya.com/blog/2018/04/a-comprehensive-guide-to-understand-and-implement-text-classification-in-python/
- # - Jason Brownlee's Feature Selection For Machine Learning in Python
- # https://machinelearningmastery.com/feature-selection-machine-learning-python/
- # - Susan Li's Multi-class text classification with Scikit-learn:
- # https://towardsdatascience.com/multi-class-text-classification-with-scikit-learn-12f1e60e0a9f
- # - Vadim Smolyakov Ensemble Learning to Improve Machine Learning Results:
- # https://blog.statsbot.co/ensemble-learning-d1dcd548e936
- # - Udacity course video on Youtube UD120:
- # https://www.youtube.com/watch?v=GdsLRKjjKLw
- # - Hyperparameter Tuning with Hyperopt
- # https://towardsdatascience.com/automated-machine-learning-hyperparameter-tuning-in-python-dfda59b72f8a
- # - Hyperparameter Tuning for Gaussian NB
- # https://www.quora.com/Can-the-prior-in-a-naive-Bayes-be-considered-a-hyperparameter-and-tuned-for-better-accuracy
- # - Hyperparameter Tuning for Decision Trees
- # https://towardsdatascience.com/how-to-tune-a-decision-tree-f03721801680
- # - Lime tutorial
- # https://marcotcr.github.io/lime/tutorials/Lime%20-%20multiclass.html

YORK UNIVERSITÉ UNIVERSITY