**Explainable Artificial Intelligence - Final Project Report**

Muhammad Waheed Sabir - *muhammadwaheed.sabir@phd.unipi.it*

Muhammed Imran - *muhammad.imran@phd.unipi.it*

Cosimo Palma - *cosimo.palma@phd.unipi.it*

*Scuola Normale Superiore di Pisa*

*Winter semester 2021-2022*

### **Introduction**

The present document reports our joint effort to apply some of the methods learnt during the course of Explainable Artificial Intelligence (from now on addressed as XAI). The dataset chosen for this purpose is a collection of **Yelp Reviews Dataset** [1].

We selected The project type A, whose required tasks are as follow:

* definition of a prediction/classification task;
* application of a standard data mining pipeline (data understanding, preparation, training and evaluation);
* on the machine learning model developed, explanations generation (by

exploiting several approaches using either methods presented during the hand-on lessons, or other state-of-the-art ones);

* Exploration, comparison and discussion of the obtained results (particularly, arguing on their usefulness for constructing a user-understandable statement of the decision-making process for a given classification task).

### **Data Preprocessing and Exploratory Data Analysis (EDA)**

For the classification task we opted for the Sentiment Analysis, with a simple binary outcome (positive or negative), which was inferred from the column “label” (containing the rating from 1 to 5) as in the following.

content label sentiment

0 My wife took me here on my birthday for breakf... 5 1

1 I have no idea why some people give bad reviews... 5 1

2 love the gyro plate. Rice is so good and I als... 4 1

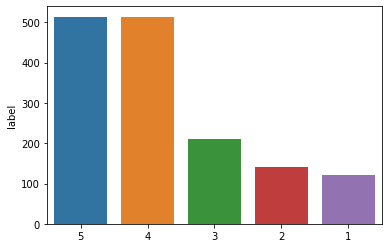
3 Rosie, Dakota, and I LOVE Chaparral Dog Park!!... 5 1

4 General Manager Scott Petello is a good egg!!!... 5 1

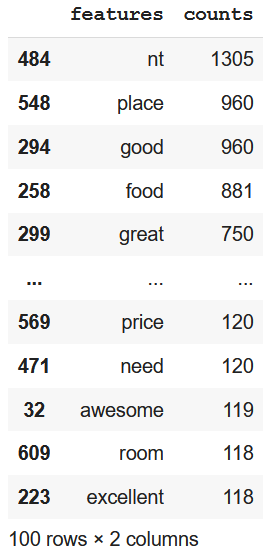
... ... ... ...

1496 Yum best donuts in Phoenix. 5 1

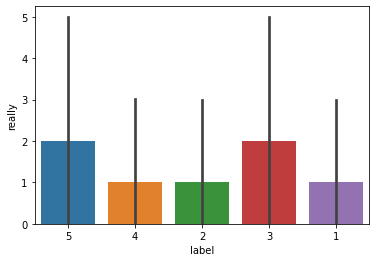
1497 Alright we are going to say that this place is... 5 1



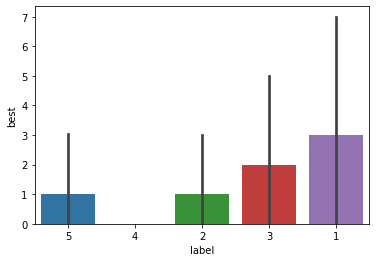
From this histogram as well we understand that the amount of positive reviews is far higher than the negative ones, which means that terms associated with a positive sentiment (1) will be more reliable, because based on more observations.



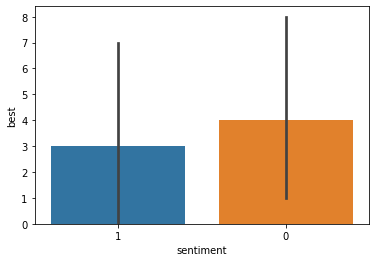
The function *CountVectorizer* from the sklearn library is used to create a type-dictionary of our dataset whereby each word is treated as a feature, i.e. it converts a collection of text documents to a matrix of token counts. This kind of processing is necessary to be able to assign a sentiment to each word, as well as feeding the models in the next part of the pipeline..



The following pictures display the distribution of labels across words (training dataset). The world “really” is indeed pretty neutral, presenting a homogenous distribution.



Surprisingly, the term “best”, usually associated with positive feelings, is mostly labeled as negative. This represents already a noticeable case to question the chosen heuristics to map words to sentiment..



The sentiment-based histogram for “best” presents only a slight difference, because we have chosen to consider the 3-stars reviews as positive as well. In the final discussion we will better consider if this attribution is wrong or not, and why.

### 

### **Training and Testing**

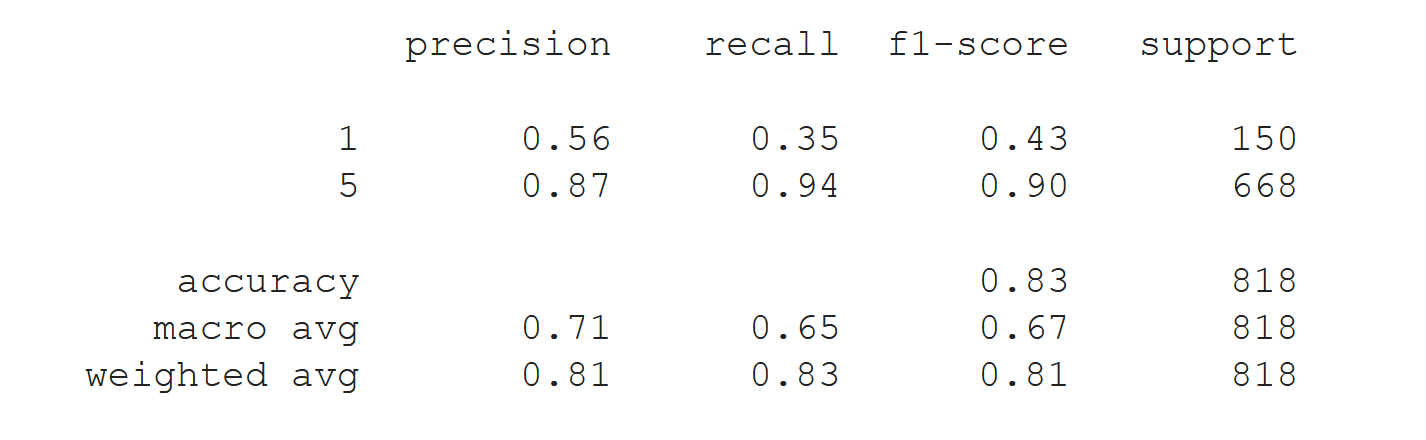
Before diving directly in evaluating a bunch of training models, let us briefly sketch a tabular overview of the used metrics.

**True Positives** (TP) - correctly predicted positive values  
**True Negatives** (TN) - correctly predicted negative values   
**False Positives** (FP) – not correctly predicted positive values  
**False Negatives** (FN) – not correctly predicted negative values

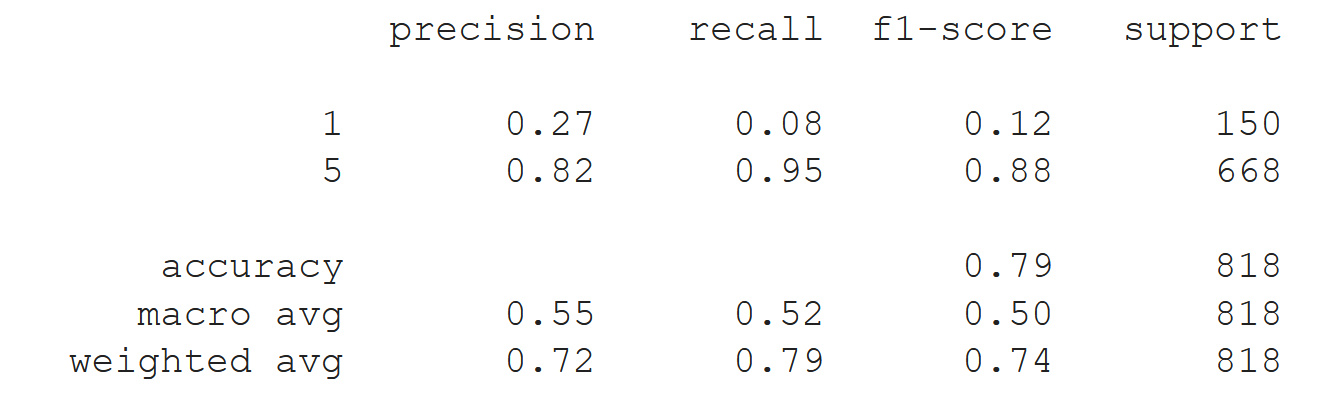
Accuracy, Precision, Recall and F1 score are directly derived from the above-defined features.

**Accuracy** = TP+TN/TP+FP+FN+TN  
**Precision** = TP/TP+FP  
**Recall** = TP/TP+FN  
**F1 Score** = 2\*(Recall \* Precision) / (Recall + Precision)  
**Macro average** = Calculate metrics for each label, and find their unweighted mean. This does not take label imbalance into account.  
**Weighted average** = Calculate metrics for each label, and find their average weighted by support (the number of true instances for each label). This alters ‘macro’ to account for label imbalance; it can result in an F-score that is not between precision and recall [18].  
The **support** is the number of occurrences of each class in y\_true [18].

#### ***Multinomial Naive Bayes***[20]

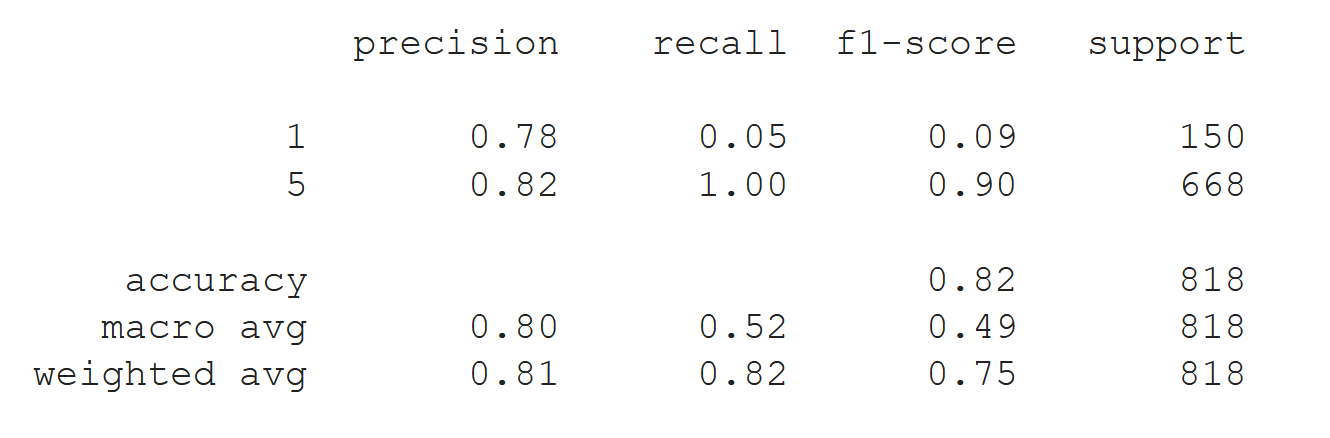


#### ***KNN-Classifier***[21]



#### ***Support Vector Machine***[22]

#### ***Random Forest Classifier***[23]



#### ***Bi-GRU-LSTM-CNN***[24]

accuracy: 0.8416666666666667

#### ***Bi-LSTM*** [25]

accuracy: 0.8758333333333334

Bi-LSTM is the model achieving the highest accuracy, for which reason it will be the one used for our explanations.

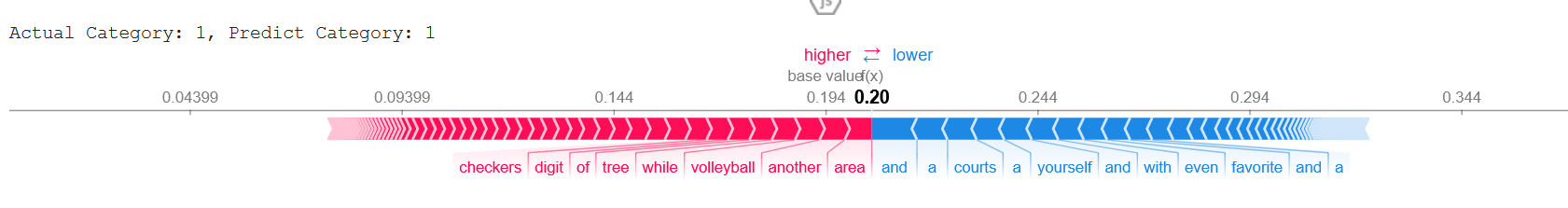
### **Explanation Methods**

Despite the high accuracy of machine learning algorithms, often their opaque nature makes it difficult even for the domain experts to understand and trust them. There is an ultimate need in the field of eXplainable Artificial Intelligence (XAI) to seek new solutions to fill the existing gap between accuracy and interpretability in order to improve the trust and transparency of AI-enabled systems.

#### **SHAP**

SHAP method provides consistent explanations of models internal mechanisms based on game theory [17]. The main objective of the SHAP is to highlight or explain the individual contribution of various features of any instance which helped the Machine Learning / Deep Learning model to predict a certain class. It can be explained using the example of a cricket match scorecard summary which highlights the individual contribution of every player in terms of scores. From that score card it can be easily seen which players performed well in the match. In the same way SHAP helps to understand the importance of individual features in different predictions of a model which are helpful to validate the models and further improve their performance**.**

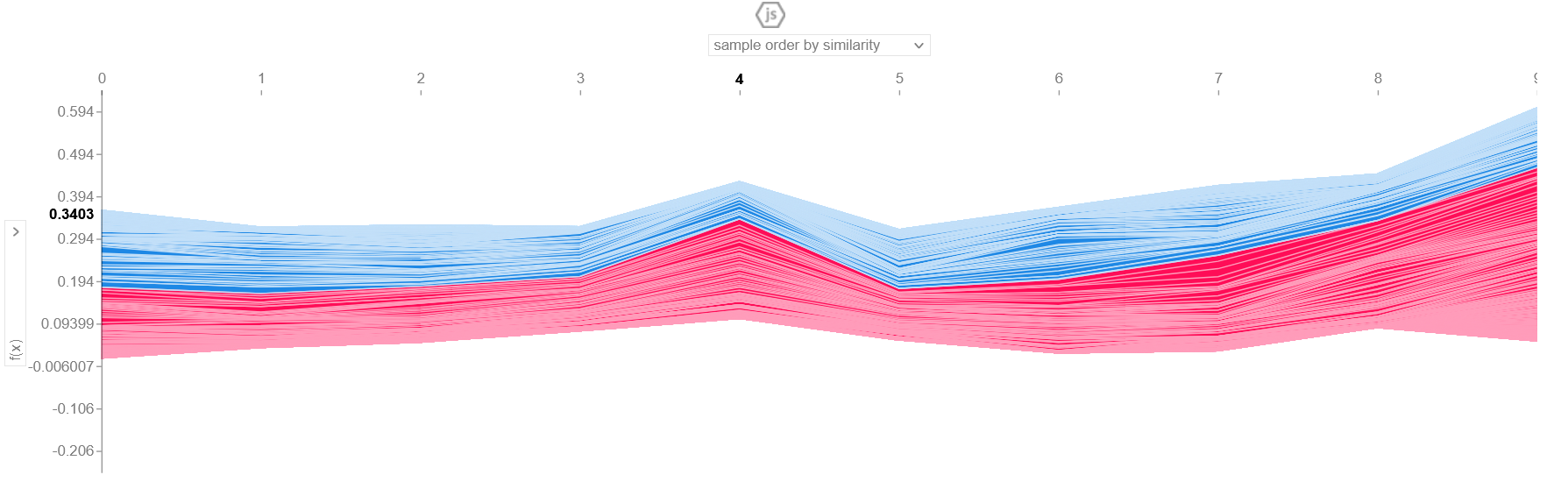
The following visualizations are produced with SHAP library which helps to make pretty visualizations of simple numeric measures to see which features were important to a model. This helps to make comparisons between features easily, and one can present the resulting graphs to non-technical audiences to explain the internal mechanism of the black box models.

****

Following is the explanation of the above diagram to explain the text sentiment analysis model trained on yelp reviews dataset,

* SHAP has highlighted the words in a review instance with blue and red colors where words highlighted with blue color represents that they have contributed positively to predict the class label and on contrary the words highlighted with red color shows that they have negatively contributed
* SHAP has highlighted attributes "court" , "favorite" etc to classify the review as positive sentiment
* This force plot of Kernel Explanation explains how different features pushed and pulled on the output to move it from the base\_value to the model output value or prediction. The prediction of this kernel explanation is the probability or confidence that the review belongs to the predicted category ("Positive" Class).

The following graph shows the importance of individual features in predicting the class label or sentient category. The graph contains the two colors on the map, the first one (blue) being the one for positive SHAP values, and the second one (pink) for the negative SHAP values.

****

#### **LIME**

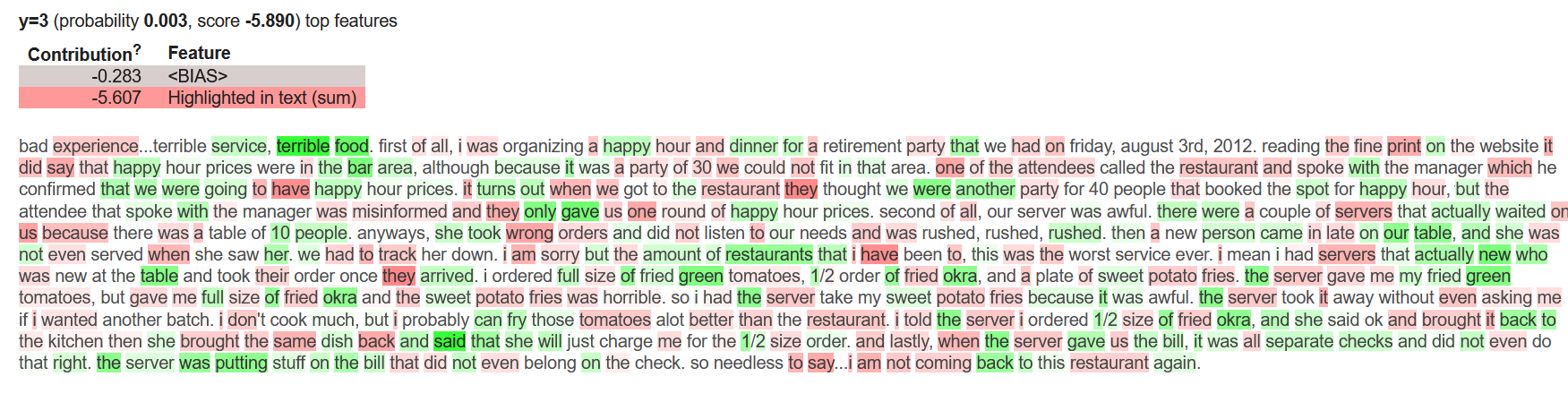
LIME is a model agnostic library to describe the importance or contribution of individual features of a model to predict a certain class. It is helpful to see the impact of perturbing on the input instances to pinpoint when the class label changes [15].

The following visualization shows the explanation of the sentiment classification of a random instance from the yelp review dataset. The words with negative sentiment are highlighted with “blue” color and words with positive sentiment are highlighted with “orange” color. From the diagram it can be seen that “0.12” or 12% is the rate of certainty of negative sentiments whereas the positive sentiment is labeled with a “0.88”, i.e. 88% precision. The degree of precision is also reflected from the color´s intensity.

**ELI5**

ELI5 is a python package which provides the ability to understand the internal mechanism of the ML models [3]. It can be used to visualize the various parameters of a model as well as to exploit or understand the individual predictions of a model based on the significance of different features. It provides a convenient way to debug the decisions made by the model.

The following ELI5 based explanation visualization highlights the positive sentiment words with green and negative sentiment words with red colors. It can be seen that the overall sum of the negative words is “5.607” or 56.07% as this review is classified as a negative sentiment review.

****

### **Explanations’ comparison metrics and final considerations**

The used explanation methods focus on different visualizations and heuristics. Nevertheless there are different parameters which should be taken in consideration to validate them. Two of these are fidelity and computation time [4]. Fidelity aims at evaluating how well the explanation model is at simulating black-box decisions. This one is hence available only for methods which construct a surrogate model, as in the case of LIME [10].In the context of feature importance based explanations [5], we can evaluate the explanations also by *faithfulness* and *monotonicity*: the former aims to validate if the relevance scores imply true importance: “higher importance values are expected for attributes that greatly influence the final prediction. The faithfulness method incrementally removes each of the attributes deemed important by the explanation. At each removal, the effect on the performance is evaluated. These values are then employed to compute the overall correlation between feature importance and model performance. These metrics correspond to a value between -1 and 1: the higher the value, the better the faithfulness of the explanation”. [10]

The latter, on the other hand, “evaluates the effect of the explanation by querying the black box model, by incrementally adding each attribute in order of increasing importance. We expect that the black-box performance increases by adding more and more features, thereby resulting in monotonically increasing model performance”.[10]

To these two metrics, which *quantitatively* assess the accuracy of the explanations, we can add a wide range of *quantitative* ones, like *comprehensibility* (how much effort is needed for correct human interpretation)*, succinctness* (how concise and compact is the explanation?)*, actionability* (what can one do with the explanation?)*, reusability* (could the explanation be personalized?) *and completeness* (is the explanation complete, partial, restricted?) and *stability* (does the explanation change, if the same input is given several times?)*.* Although difficult to quantify, these metrics are not subjective.

In the paragraph “Data exploration and preprocessing” we have already pointed out that the “human” classification criteria may be not totally correct. Therein, the word best was classified as slightly negative, although it shall not be the case. There are few words that are “generally positive” and “best” seems to be one of those. It can be labeled as negative only if it happened to appear in sarcastic reviews. This kind of inconvenience could be possibly avoided by enhancing the volume of the dataset, which in our case was heavily shrunk to avoid computation issues in the Colab file. Otherwise, the sentiment analysis could be tuned on the sentence- instead of the word-level. Moreover, all the considered methods seem to put more stress on the result itself, on how the model worked to reach it: the main interest of the user (possibly an AI-outsider) is to understand how the result has been reached, usually to evaluate their reliability, but not always. Animations would be a remarkable tool to this degree of insight, which would harmonize ideally with the application of attention layers[4]. This would really avoid the comprehensibility-completeness dilemma, giving the user the possibility to understand model and results without excessive oversimplification or obscurity.

### **Consulted Literature and Online Resources**

[1]https://www.kaggle.com/yelp-dataset/yelp-dataset

[2] https://github.com/mayank100sharma/Sentiment-Analysis-on-Yelp-Reviews

[3] https://eli5.readthedocs.io/en/latest/tutorials/black-box-text-classifiers.html

[4] F. Bodria & al., *Explainability Methods for Natural Language Processing: Applications to Sentiment Analysis*, <http://ceur-ws.org/Vol-2646/18-paper.pdf>

[5] M. Sareela, *Comparison of feature importance measures as explanations for classification models,* <https://link.springer.com/article/10.1007/s42452-021-04148-9>

[6] https://medium.com/@kalia\_65609/interpreting-an-nlp-model-with-lime-and-shap-834ccfa124e4

[8] https://www.analyticsvidhya.com/blog/2021/08/text-preprocessing-techniques-for-performing-sentiment-analysis/

[9] Zhou, J.; Gandomi, A.H.; Chen, F.; Holzinger, A.; *Evaluating the Quality of Machine Learning Explanations: A Survey on Methods and Metrics*. Electronics 2021, 10, 593.

https://doi.org/10.3390/electronics10050593

[10] https://colab.research.google.com/github/francescanaretto/XAI-course\_2021/blob/main/Tabular/metrics-evaluation.ipynb

[15] https://github.com/marcotcr/lime

[17] SHAP, https://github.com/slundberg/shap#citations

[18] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision\_recall\_fscore\_support.html

[20] Multinomial-Naive-Bayes https://www.upgrad.com/blog/multinomial-naive-bayes-explained/

[21] KNN Classifier https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning

[22] Support Vector Machines https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

[23] Random forest classifier https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/

[24]*Bi-GRU-LSTM-CNN* https://medium.com/@keynekassapa13/weighted-bi-gru-cnn-ab6cdbeda77b

[25] Bi-LSTM https://analyticsindiamag.com/complete-guide-to-bidirectional-lstm-with-python-codes/