Final Project Report

Machine learning and AI

for SARS-CoV-2

Edoardo Fratantonio

Garett Morris

INDEX

PROJECT AIMS

The main aim of the study was to apply machine learning to predict ligand binding affinity for the SARS-CoV-2, especially to the main protease (Mpro). Amongst the coronaviral targets investigated in the past, the main protease (Mpro, 3CLpro, nsp5) generated a lot of interest, particularly after the initial SARS-CoV outbreak in the early 2000s.

The **"Main Protease"** has been identified as an attractive antiviral target because it plays a significant role in breaking Cov-encoded polyproteins that facilitate the assembly of replication-transcription machinery [1]. Mpro digests the polyprotein at 11 distinct locations, beginning with autolytic cleavage of this enzyme from “pp1a” and “pp1ab”. Furthermore, Mpro lacks a human counterpart and is substantially preserved across all CoVs. These features make it an appealing drug target against CoVs. Following the COVID-19 pandemic, the crystal structure of SARS-CoV-2 Mpro was soon realised, facilitating its pharmacological study and inhibitor development [2].

This is where machine learning comes in; it will be used to predict how strong the binding affinity between the ligand and the Mpro is; this will ease the job for scientists to determine which ligand would function more efficiently as an inhibitor for halting the biological process of SARS-CoV-2.

METHODOLOGIES

Machine learning is widely used nowadays, for tackling many problems in our society and it is expanding exponentially across other fields, such as, chemistry, biology, physics, neuroscience and so forth. Machine learning is a subset of Artificial Intelligence, which can be divided in three different main categories:

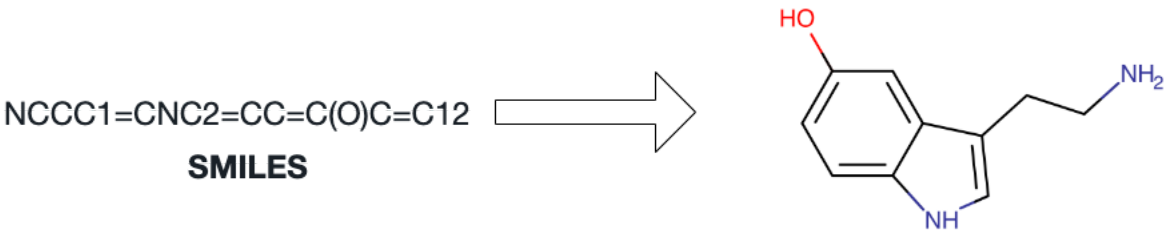
- **Supervised Learning:** Where we have “X” and “Y”, “X” being the Data and “Y” being the Labels, and the model, meaning the environment that the training is based on, learning from the Data and predicting “Y”.

- **Unsupervised Learning:** Where we only have X, and the model will try to learn the patterns of the data in order to be able to predict “Y”.

- **Reinforcement Learning:** Neither “X” nor “Y” are given, it essentially is the science of decision making. What I mean by this is that an Artificial Intelligence given a certain environment, will try to make decisions and will understand if those decisions are correct through rewards.

The one used for this project is the Supervised Learning, as for this study both “X” , the compounds, and “Y”, the IC50 values. Even though “X” is where the creativity comes in, any kind of features can be used for “X”, and for this project the SMILES were used.

SMILES (Simplified Molecular Input Line Entry System) is a chemical notation that allows a user to represent a chemical structure in a way that can be understood by the computer [3]. This allows to transform any molecule into a string, which will represent the molecule in question and extract as many features as possible.



Fig[1.0]
[on the left there is the SMILES, which represent the String on the right "SEROTONIN"]

In this study the Covid-19 Moonshot dataset from “PostEra” was used. The dataset had the SMILES, which in total were 2063 compounds, and both rapidfire and fluorescent IC50 values, in micromolar.

The IC50 value represents the concentration at which a drug inhibits the biological process of 50 percent. The smaller the number is, the stronger it binds. However, for this project, the values were in micromolar, therefore they had to be changed first in molar and then into the pIC50 value, which is simply the negative log of base 10 of the IC50 value, when converted to molar.

The reason pIC50 is preferred is because, first of all, the nature of potency values is logarithmic, for example the dose-response curves, they are sigmoidal when you plot them in logarithmic space. Also because it is an easy-read form. Furthermore, it helped to instantiate a classification threshold, which in this case was 6, therefore any pIC50 value greater than “6.0” would have been labelled as “1”, otherwise “0”.

After converting the values to pIC50, the SMILES had to be featurized so that the computer could understand them. Featurization is the process by which compounds are analysed and all of their properties are extracted and translated into 0's and 1's that a machine can comprehend.

Compounds can be featurized in many ways, the one chosen for this study is fingerprint, more specifically, **”Morgan Fingerprint”** (Extended-Connectivity Fingerprints (ECFP)) and **“MACCS Keys Fingerprint”.**

- **Morgan Fingerprint:**

The "Extended Connectivity Fingerprints" Featurizer has been reimplemented (ECFP). In essence, it navigates each atom of the molecule and obtains all possible paths through the atom in question, with a specific radius inserted by the user, radius 2 was chosen for this study (ECFP4). Then, for each unique path, a number with a maximum based on bit number is recorded; the larger the radius, the greater the fragments encoded. For example, radius 2 contains all paths found in radius 1 as well as new ones. Then there's the factor of the number of bits, which in this case is 2048. This depends on the dataset, but the standard starting point is 1024 bits.The higher the bit number the more unique the fingerprint is.

- **MACCS Keys Fingerprint:**

MACCS Keys fingerprints are a 166-bit 2D structure that are commonly used to quantify molecular similarity. Each bit can represent more than $9.3 \cdot 10^{49}$ distinct fingerprint vectors, as the bits are either on (1) or off (0).

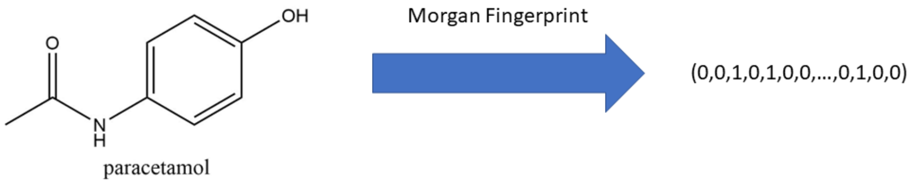


Fig [1.1] On the left we have the paracetamol molecule, which is then featurized using **Morgan Fingeprint**. On the right we can see how it is represented after the featurization.

After featurizing the data, the dataset was split into Training(75%) and Test(25%), using the Scaffold splitter method.

There are different methods for splitting the dataset, in this study both “Random Splitter” and “Scaffold Splitter” were used, however as the results will show, the

Scaffold Splitter performed better, and indeed it was the one taken in consideration for this study.

The “**Scaffold Splitter”** is based on the Bemis-Murcko scaffold representation, which identifies rings, linkers,and atomic properties such as atom type and bond order in a dataset of molecules. Then split the groups by the number of molecules in decreasing order [6]. While, on the other hand, the “**Random Splitter”** will simply split the data randomly.

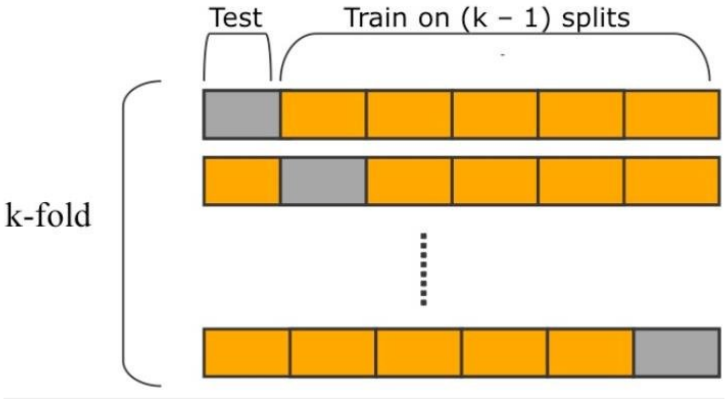
The environment chosen for this project was Colab, using Python as the development tool, with the following libraries:

- RDKit
- DeepChem
- Scikit-Learn
- Numpy
- Matplot

The libraries just mentioned were all useful for the development of the Classification software, and also for the graphs which would help to visualise the performance of the model.

In order to assess the performance of the models and understand which hyperparameters would have been the most efficient, the cross validation is what has been used. Cross Validation is a technique widely used in machine learning, especially when the hyperparameters have to be picked. Essentially, it will take the dataset passed as an input, in this case will be the training split, and then will be split into “K” mini-datasets, with “K” being the integer passed from the developer as the quantity of the mini-dataset, usually the **“10 fold cross-validation”** is the most used, in fact, is the one used for this study.

The dataset will be shuffled first, and then divided into "K" mini-datasets, where "K-1" will be used as "training" for the model, while the last piece will be utilised as a validation or test set, and repeated the same action for K times. Thus, this will test K distinct models; at the conclusion of the process, an array of K values will describe the model's performance and accuracy; an average will be taken, and the final number will indicate how well or poorly the model performed with the relative hyperparameters.



[Fig 1.2] Example of Cross Validation, where it shows that it will be split in K parts and the validation set will be K-1

In order to assess the results, different kinds of metrics have been used. The **F1 score**, the **ROC-AUC score** and the **confusion matrix.**

- **F1 score:**

The F1 score is composed from two different formulas, the **Precision** and the **Recall.** Essentially, the difference between these two is that, **precision** is more about accuracy, so how accurate is the model rather than **recall** being more towards reliability, therefore how reliable is the model.Then, once we have these two, the F1 score can be worked out, and that number will describe how reliable and accurate our model is.

$$\text{Precision}= \frac{True_{positive}}{True_{positive} + False_{Posiitive}}$$

$$\text{Recall} = \frac{True_{positive}}{True_{positive} + True_{negative}}$$

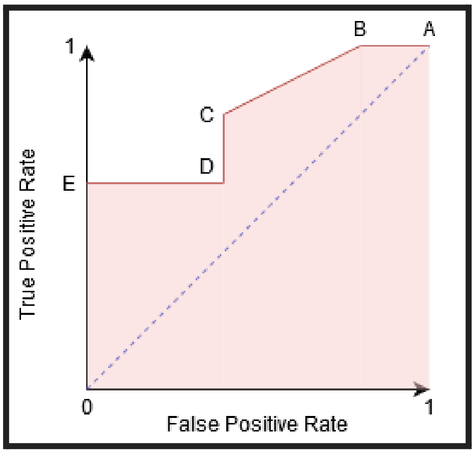
$$F1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

[Fig 1.6] F1 score formula, where it describes the Precision and Recall formula.

- **ROC-AUC score:**

Area Under the Receiver Operating Characteristic Curve (ROC AUC) from the predictions that the model made.

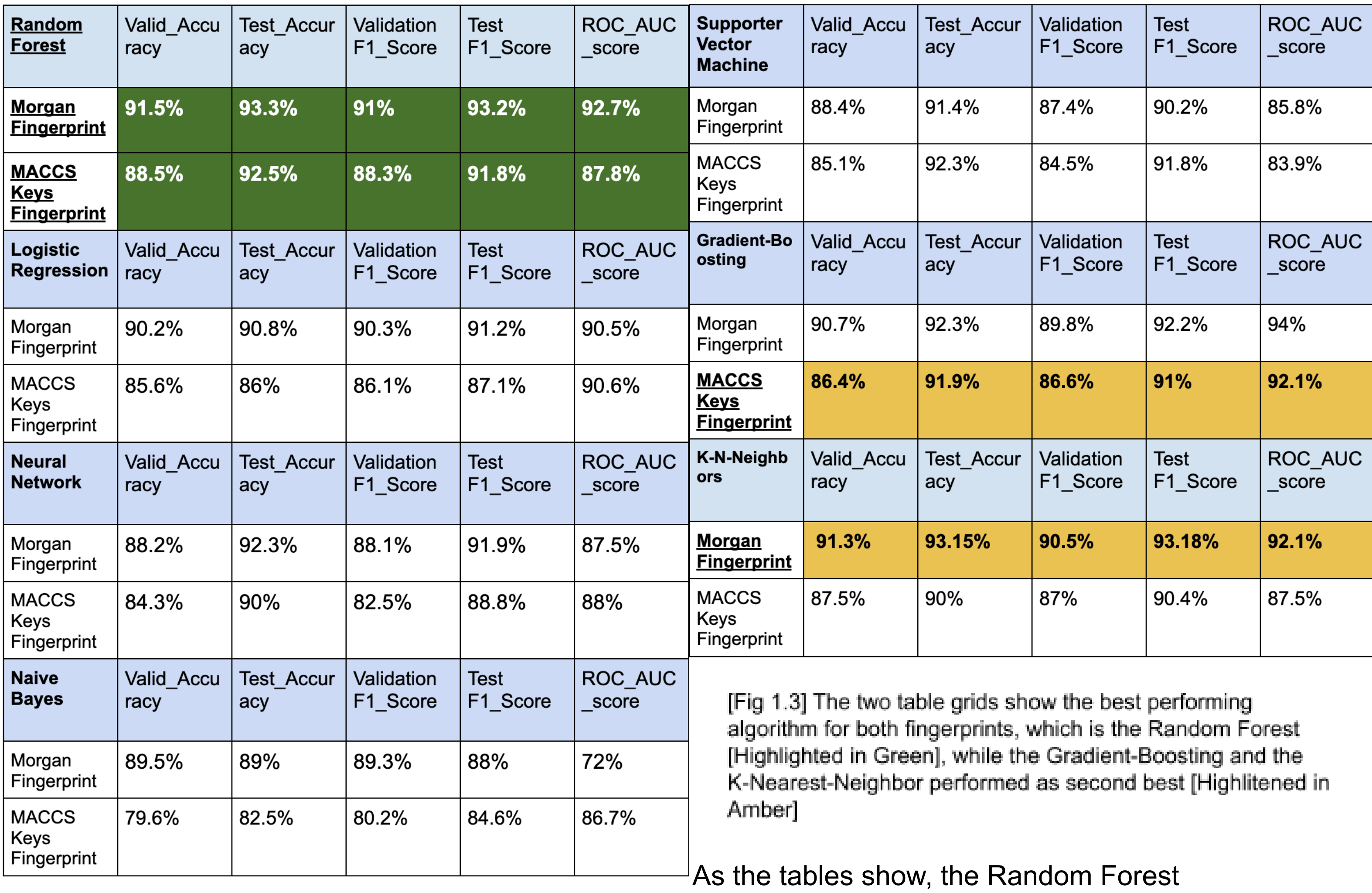
As Fig 1.7 illustrates, on the Y axes there is the "**True positive Rate"** and on the X axes the "**False positive Rate".** The AUC score will describe how well the model can distinguish between the labels, and therefore make predictions. The higher the score is, the better the performance of the model at understanding the difference between positive and negative labels.



[Fig 1.7] Illustration of the ROC-AUC score

RESULTS

As for the results of the project, in six weeks I have managed to experiment seven different algorithms [Fig 1.3] for both “**Morgan Fingerprint**” and **“MACCS Keys Fingerprint”**.



Random Forest	Valid_Accuracy	Test_Accuracy	Validation F1_Score	Test F1_Score	ROC_AUC _score
Morgan Fingerprint	**91.5%**	**93.3%**	**91%**	**93.2%**	**92.7%**
MACCS Keys Fingerprint	**88.5%**	**92.5%**	**88.3%**	**91.8%**	**87.8%**
Logistic Regression	Valid_Accuracy	Test_Accuracy	Validation F1_Score	Test F1_Score	ROC_AUC _score
Morgan Fingerprint	90.2%	90.8%	90.3%	91.2%	90.5%
MACCS Keys Fingerprint	85.6%	86%	86.1%	87.1%	90.6%
Neural Network	Valid_Accuracy	Test_Accuracy	Validation F1_Score	Test F1_Score	ROC_AUC _score
Morgan Fingerprint	88.2%	92.3%	88.1%	91.9%	87.5%
MACCS Keys Fingerprint	84.3%	90%	82.5%	88.8%	88%
Naive Bayes	Valid_Accuracy	Test_Accuracy	Validation F1_Score	Test F1_Score	ROC_AUC _score
Morgan Fingerprint	89.5%	89%	89.3%	88%	72%
MACCS Keys Fingerprint	79.6%	82.5%	80.2%	84.6%	86.7%

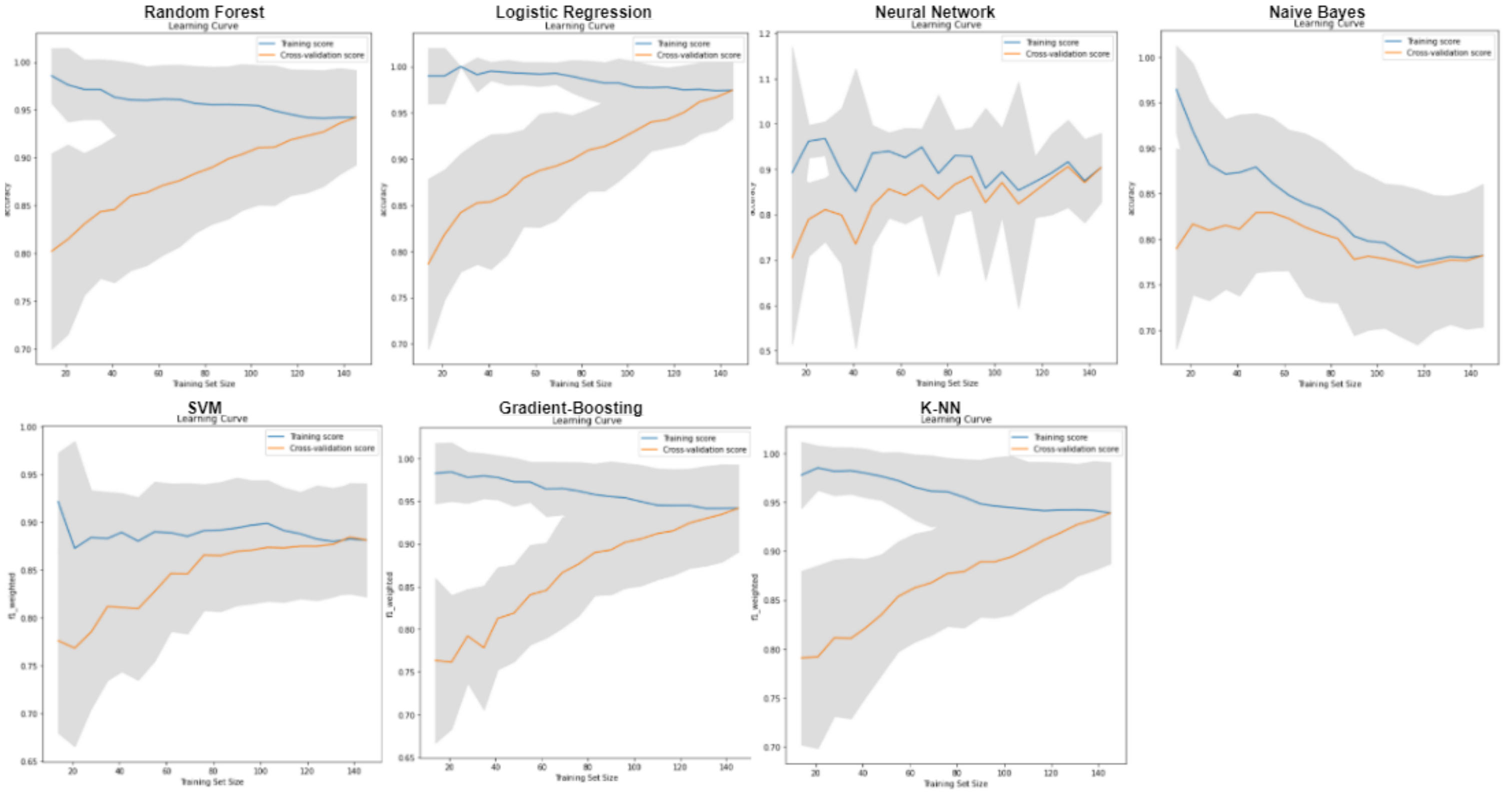
Supporter Vector Machine	Valid_Accuracy	Test_Accuracy	Validation F1_Score	Test F1_Score	ROC_AUC _score
Morgan Fingerprint	88.4%	91.4%	87.4%	90.2%	85.8%
MACCS Keys Fingerprint	85.1%	92.3%	84.5%	91.8%	83.9%
Gradient-Boosting	Valid_Accuracy	Test_Accuracy	Validation F1_Score	Test F1_Score	ROC_AUC _score
Morgan Fingerprint	90.7%	92.3%	89.8%	92.2%	94%
MACCS Keys Fingerprint	**86.4%**	**91.9%**	**86.6%**	**91%**	**92.1%**
K-N-Neighbors	Valid_Accuracy	Test_Accuracy	Validation F1_Score	Test F1_Score	ROC_AUC _score
Morgan Fingerprint	**91.3%**	**93.15%**	**90.5%**	**93.18%**	**92.1%**
MACCS Keys Fingerprint	87.5%	90%	87%	90.4%	87.5%

[Fig 1.3] The two table grids show the best performing algorithm for both fingerprints, which is the Random Forest [Highlighted in Green], while the Gradient-Boosting and the K-Nearest-Neighbor performed as second best [Highlitened in Amber]

As the tables show, the Random Forest (RF) was the one to perform best out of all the others, achieving **“91.5%”** for Validation Accuracy, **“93.3%”** for **Test Accuracy, “91%”** for **Validation F1-score, “93.2%”** for **Test F1-score,** while **“92.7%”** for **ROC-AUC score.** The test score is higher than the validation score, as the validation accuracy was taken from the cross-validation, therefore it was an average out of all the K fold, while the test accuracy was taken from the model after being fitted, meaning after training, in order to be able to make predictions.

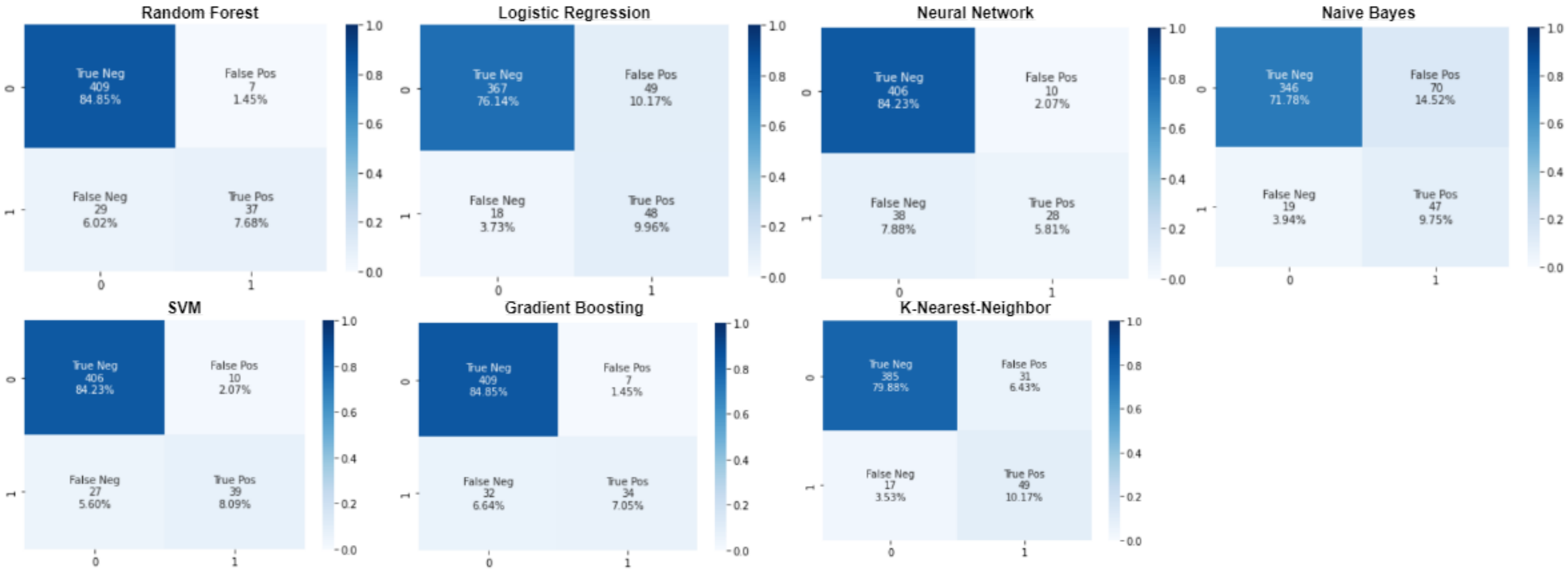
The best parameters chosen for the RF are the default taken from “Scikit-Learn”, with some adjustments.

Number of estimators = 300, criterion= ‘gini’, max_depth= 13, random_state= None.



[Fig 1.4] Showing all the algorithms learning curve from cross validation

The graphs [Fig 1.4] and confusion matrices [Fig1.5] describe how the models performed. Most of them performed really well, such as Random Forest, Logistic Regression, Gradient-Boosting and K-NN, while on the other hand the Neural Network, Naive Bayes and SVM, did not perform as well.



[Fig1.5] Confusion matrix of all the algorithms

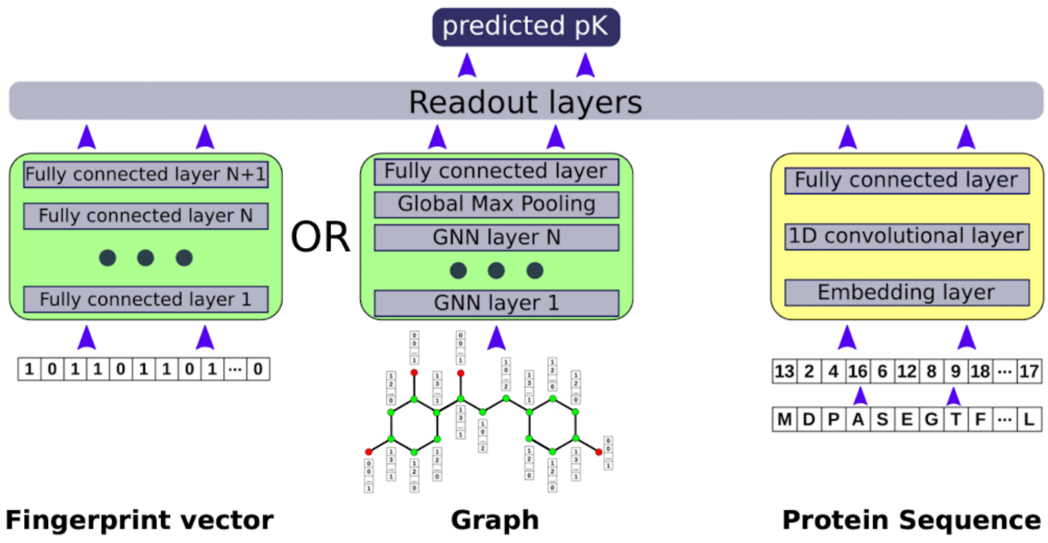
In contrast, the confusion matrices [Fig 1.5] describe the relationship between **"True Negative," "False Positive," "False Negative," and "True Positive"**. This will demonstrate how accurate the model is at predicting labels. All of the values are important, but the "True Negative" and "True Positive" values are critical in understanding how often the model is correct; thus, the higher the value, the better, depending on the labels of the dataset. Nevertheless, "False Negatives" and "False Positives" are extremely crucial; the aim is to keep the number as low as possible; for example, if the model predicts that a molecule cannot bind to a ligand when it actually can, or vice versa, the model is inaccurate and therefore useless.

The “**True Negatives”** is much higher compared to the “**True Positive”** because the dataset itself had more “zero labels” than the “one labels”, it was a 7:3 split, therefore, the model predicted what it had as labels. Furthermore, for the test set, which was a total of 466, the “zero labels” were 410, and 56 for the “one labels”.

FURTHER STUDIES

Further studies would be needed for this project, also because six weeks were not enough to experiment the variety of ways to experiment with this problem. One of the ways which would be very interesting experimenting with is the **PLIGs** (Protein-Ligand Interaction Graphs) [7].

PLIGs are a simple method for representing atom-atom contacts in 3D protein-ligand complexes as GNN note features. It featurizes an atom node in the molecular graph by describing each atom's properties as well as all atom-atom contacts made with protein atoms within a certain distance.



[Fig 1.8] [7] Representation of PLIGs

One of the most important properties of a small molecule drug in early stage preclinical drug discovery is its binding affinity for the appropriate protein target. A high binding affinity is absolutely essential for a drug's overall efficacy. PLIGs can directly combine 3D interactions into atom node molecular graphs, each ligand atom's intermolecular contacts in the 3D protein-ligand complex are encoded.

If combined with the AE score (Atomic Environment Vectors), another fascinating algorithm for learning protein-ligand binding affinity, would be extremely interesting. This is subject to further investigation because it would be very appealing to see what kind of results it would achieve; however, more time would be necessary to experiment as extensively as possible in order to identify one of the best methods for dealing with binding affinity.

CONCLUSIONS

This study provided insight into the prediction of ligand binding affinity for SARS-CoV-2. As a result, the algorithms tried produced interesting findings. Unfortunately, the dataset was not exhaustive, which means that because of the threshold instantiated, the "zero labels" were significantly larger than the "one labels," and some of the compounds did not have the IC50 value, and thus could not be included in the study. Furthermore, the Fluorescent was chosen for this experiment, even though the Rapidfire was also investigated, but the dataset was too limited and the results could not be generalised. I am looking forward to more research and seeing new and improved methodologies that will undoubtedly make an impact in the field of cheminformatics.

Bibliography

The code of this project can be found:
https://github.com/edofrata/ML_AI_SARS-CoV-2_Oxford.git

[1] Cui, Wen, et al. “Frontiers | Recent Progress in the Drug Development Targeting SARS-CoV-2 Main Protease as Treatment for COVID-19.” *Frontiers*, www.frontiersin.org, 1 Jan. 2001, https://www.frontiersin.org/articles/10.3389/fmolb.2020.616341/full.

[2] Cui, Wen, et al. “Frontiers | Recent Progress in the Drug Development Targeting SARS-CoV-2 Main Protease as Treatment for COVID-19.” *Frontiers*, www.frontiersin.org, 1 Jan. 2001, https://www.frontiersin.org/articles/10.3389/fmolb.2020.616341/full#h3.

[3]“SMILES Tutorial | Research | US EPA.” *SMILES Tutorial | Research | US EPA*, archive.epa.gov, https://archive.epa.gov/med/med_archive_03/web/html/smiles.html#:~:text=What%20is%20SMILES%3F,learn%20a%20handful%20of%20rules.

[4] “What Is pIC50? - Collaborative Drug Discovery Inc. (CDD).” *Collaborative Drug Discovery Inc. (CDD)*, 31 July 2018, www.collaborativedrug.com/what-is-pic50-2.

[5] “How to choose bits and radius during circular fingerprint calculation in RDKit?” https://www.researchgate.net/post/How_to_choose_bits_and_radius_during_circular_fingerprint_calculation_in_RDKit

[Fig [1.1]] Laksh. “A Practical Introduction to the Use of Molecular Fingerprints in Drug Discovery | by Laksh | Towards Data Science.” *Medium*, 24 Aug. 2020, towardsdatascience.com/a-practical-introduction-to-the-use-of-molecular-fingerprints-in-drug-discovery-7f15021be2b1

[6]“Splitters &mdash; Deepchem 2.6.1.Dev Documentation.” *Splitters &mdash; Deepchem 2.6.1.Dev Documentation*, deepchem.readthedocs.io/en/latest/api_reference/splitters.html

[Fig 1.2] “Cross Validation“ https://www.researchgate.net/figure/K-fold-cross-validation-In-addition-we-outline-an-overview-of-the-different-metrics-used_fig2_326866871

[7] Moesser, Marc A., et al. “Protein-Ligand Interaction Graphs: Learning from Ligand-Shaped 3D Interaction Graphs to Improve Binding Affinity Prediction | bioRxiv.” *bioRxiv*, www.biorxiv.org, 7 Mar. 2022, https://www.biorxiv.org/content/10.1101/2022.03.04.483012v1?rss=1.