0) ABSTRACT

The identification of Drug – Target interactions (DTIs) is a one of the most crucial steps in drug development and discovery. It can lead researchers to develop novel drug compounds for the already existing drugs. Identifying these Interactions is very tedious, expensive and requires a lot of wet lab experiments; A humongous number of these interactions still remain undiscovered. Many methods exist which use the properties of the sequence and the ligands to predict these interactions; In this method I incorporate the properties of the binding site along with these properties. I created a new dataset which was built from the complex files present in the Protein Data Bank. Finally, I used various Machine Learning algorithms and our accuracy peaked at 84.4% and got a ROC value of 0.918.

1) INTRODUCTION

Firstly, pharmaceutical discovery and development is a very complicated and costly research with a lot of limitations. Ashburn and Thor provided an alternative method to accelerate drug discovery, which is repurposing/ repositioning an older drug. To elaborate more, it takes roughly 1-1.5 billion dollars on an average and at least 10-15 years to get the drug approved by the FDA and bring it to the market. Whereas drug repurposing or drug repositioning is relatively inexpensive as all the formalities to approve a drug have already been taken place and the drug is ready to be bought into the market. For example – Viagara was used to treat hypertension and angina; one of its side effects was that it induced penile erection and now after tweaking the drug a little, it is being used to cure erectile dysfunction.

Secondly, many of the interactions between drug compounds and pharmacological targets are unknown. Discovering these interactions plays a crucial role in developing new targets for existing drugs or discovering novel drug candidates for current targets.

2) METHODS

In this section I will go through how a general DTI model presented works and what things I did differently.

2.1) Dataset General:

The most widely used Dataset is referred to as the "Gold Standard Dataset" and has four types of protein targets which are: GPCR, Nuclear Receptor, ion channel and enzyme. It was released by Yaminishi et al. All of the interactions in this dataset are based on DrugBank, BRENDA, KEGG BRITE and SuperTarget Database.

Dataset	**Drugs**	**Proteins**	**Positive Interactions**	**Possible Negative Interactions**
Nuclear Receptor	54	26	90	1314
GPCR	223	95	635	20550
Ion Channel	210	204	1476	41346
Enzyme	445	664	2926	292554

2.2) Dataset Mine:

To evaluate the proposed DTI Model, I used the protein-ligand complex files and the FASTA files in the PDB; The protein structure, binding site and the ligand were extracted from these files. First, all the PDB ID's under the "Source Organism" sections were gathered. Secondly, all the Complex files and the FASTA files for these ID's were downloaded and saved, there were a total of 96,000 unique ID's. Thirdly, an algorithm (Discussed below) was used to extract the binding sites and the corresponding ligand present there.

2.3) Calculating Possible Negative Interactions:

There are 158,000 total binding sites, but the number of unique binding sites is equal to the number of Unique Positive interactions which is 55681.

- Let there be “L” ligands and “B” unique binding sites
- On an average, each protein has 3 unique ligands binding to it.
- For a negative interaction, I randomly select a ligand which is not a binder to the protein.
- On an average, each binding site can have L – 3 negative interactions.
- So, for B binding sites, there can be (L-3)^B possible negative interactions.

Ligands	**Binding Sites**	**Unique Positive Interactions**	**Possible Negative Interactions**
29822	158,000	55681	29819^55681

2.4) Filtering out Protein-Ligand complex files:

A File was loaded as a RdKit molecule, if the molecule could not be sanitized, it was neglected.

2.5) Extracting Binding Site:

A protein-ligand complex file was read using RdKit and all the atoms were segregated into ATOMS (Corresponding to the residues/ atoms of the protein) and HETATM (Corresponding to the residues/ atoms of the ligand/ water molecule). All the water molecules under HETATM were removed. Then the coordinates, residue ID and Chain of each ligand attached were extracted. All the ATOMS present within 4A of the ligand coordinates were said to be the atoms of the binding site. After these atoms were extracted, the entire residue which these atoms belonged to were considered to be the primary binding site residues.

The residues present under REMARK 800 (labelled as AC1, AC2 and so on) for each ligand were considered to be the secondary residues for the binding site. A union of the primary and secondary were taken to form the final binding site.

Once the binding site was constructed, it was checked if it could be sanitized; If it could not be, it was neglected.

2.6) Blacklisting Ligands:

There are some ligands which are very small and others which occur very frequently; In order to remove these ligands, a minimum threshold of 17 atoms was enforced on each ligand. So, if the number of atoms in a ligand were less than 17, it was blacklisted and not considered.

This is very important, as if I considered ligands which were occurring very frequently, the model would be biased to the properties of such ligands and always predict them to be binders.

Ligands	**Binding Sites**	**Unique Positive Interactions**	**Possible Negative Interactions**
29146	73358	30269	29142^30269

2.7) Calculating Features:

The model requires 4 different types of features to be calculated: Ligand, Binding Site, Sequence and Volume.

2.7.1) Ligand:

The ligand features/ properties were calculated using RdKit and by using the library PyDPI (which was outdated and most of the code had to be modified). Example of a few features are: Crippen Log P, Number of Hydrogen Bond Donors/ Acceptors, maximum positive/ negative charge etc.

2.7.2) Binding Site:

These features were the same as the Ligand features but calculated for the binding site instead.

2.7.3) Sequence:

The entire sequence was used to calculate the features/ properties of the protein (entire protein). Each Functional group in a Sequence has different properties; and using these properties of each functional group, the properties like Hydrophobicity, Electronegativity, Di-peptide Composition etc were calculated.

Ex: Calculate Hydrophobicity of Sequence: AAARRRNNNDDDARAA

- There are 6 A’s, 4R’s, 3 N’s and 3 D’s

- Hydrophobicity of each group:
    - A: 0.02
    - R: -0.42
    - N: -0.77
    - D: -1.04
- F(A) = Hydrophobicity of A * occurrence of A
- Overall = (F(A) + F(R) + F(N) + F(N) + F(D)) / length of sequence
- Overall = (6*0.02 + 4*(-0.42) + 3*(-0.77) + 3*(-1.04)) / (6+4+3+3)

2.7.4) Volume:

The volume feature is the volume of the binding site, which is calculated using PyVol. In order to calculate the volume, we need the Residue ID and the PDB ID of the Ligand attached to the Binding Site (which are noted down while extracting the binding site).

In total there were 1012 Features.

2.8) Representing and Merging Features:

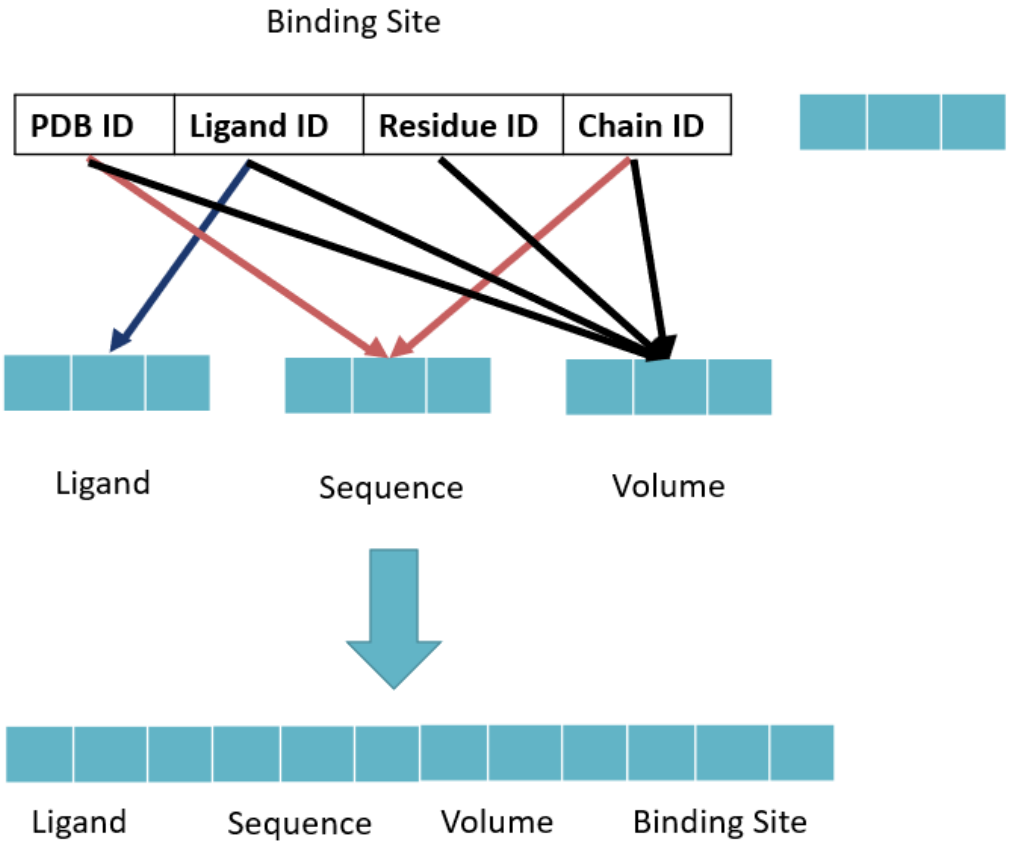
2.8.1) Representing Features:

Each feature was calculated independently (each feature calculated was a numerical value) and stored as a hash map to ensure very fast retrieval. The keys for the hash map were:

- Binding Sites: a tuple which contained the PDB ID, Ligand ID, chain ID and the Residue ID.
- Sequence: tuple of sequence ID and chain ID.
- Ligand: it was the Ligand ID
- Volume: it was the same as the keys for Binding Site.

2.8.2) Merging Features:

The Binding Site Map key contains information regarding every other key and also that the total number of positive interactions is equal to the number of Binding Sites; I looped over all the keys of Binding Site, used the keys to get all the other features and merged them. Demonstrated below:



The above picture demonstrates the merging of features of one binding site. Similar merging was done for each binding site to generate the positive data.

After generating the positive data, all the redundant rows were removed leaving me with 30,000 unique positive data-points.

2.9) Generating Negative Data:

As discussed above, negative data was generated by random selection. To generate this data, the columns containing the ligand features were separated out of the positive dataset. The ligand features were shuffled row-wise and then appended back to the remaining dataset.

This technique was used over other techniques as it was 10-20 times faster than other techniques of sampling.

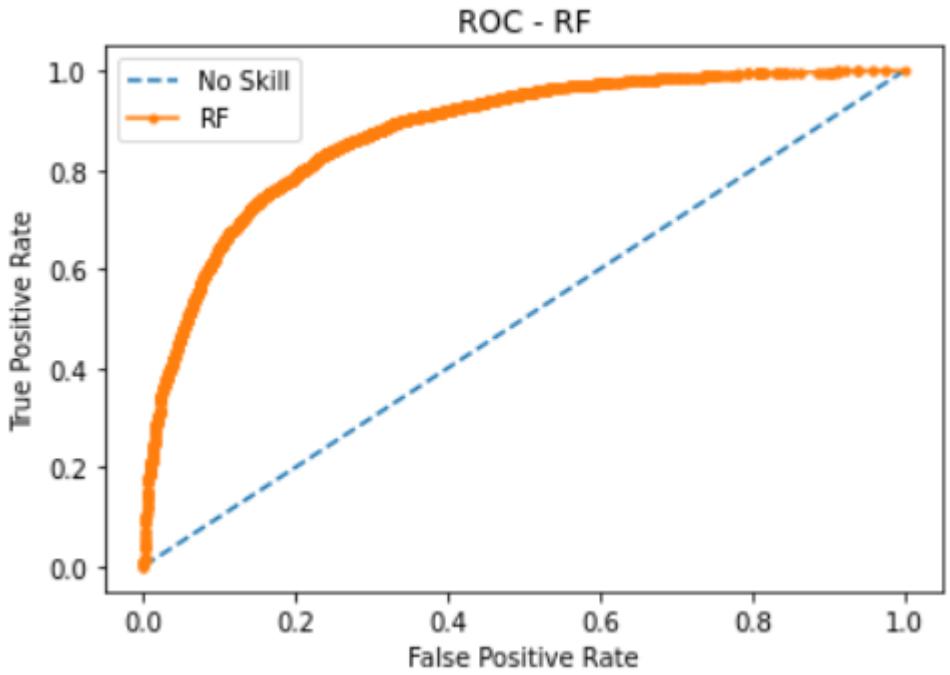
3) DRUG TARGET INTERACTION CLASSIFICATION RESULTS

Considering the complexity of DTI classification in terms of pattern recognition I opted to implement different classification models to determine which provides the best predictive performance. The predictive models used in this study were implemented using scikit-learnand XgBoost, a Python package to perform data mining, data analysis and machine learning tasks. I implemented three different classification models Random Forest, SVM and XgBoost. In the end, I had an option to use either of the classifiers or use an ensemble of them to get a better result.

Each classifier was trained with the positive and negative data as discussed above, their hyperparameters were tuned and were trained using 10 fold Cross Validation.

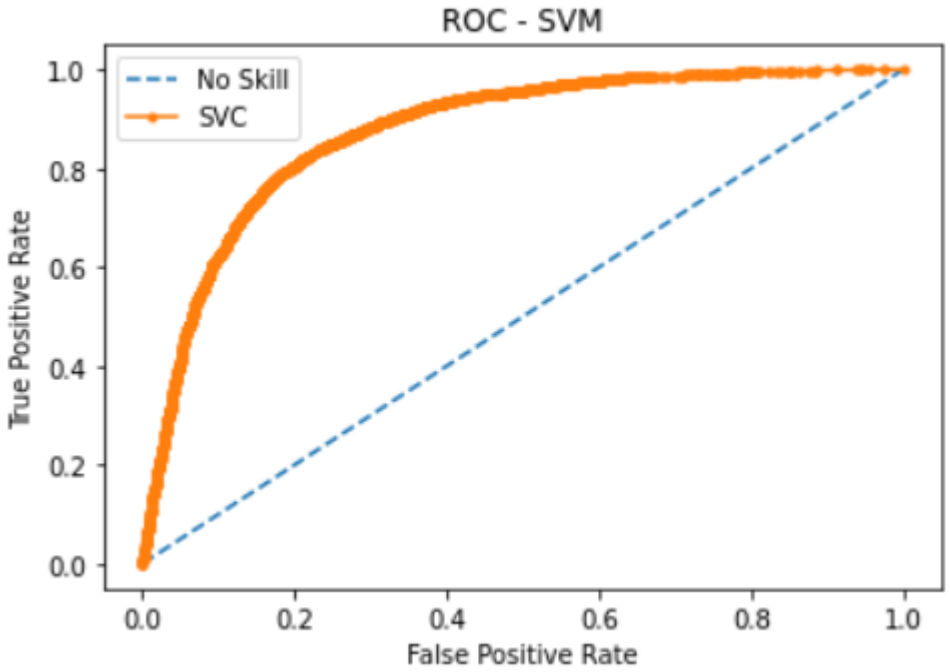
3.1) Predictive model Evaluation:

3.1.1) Random Forest:



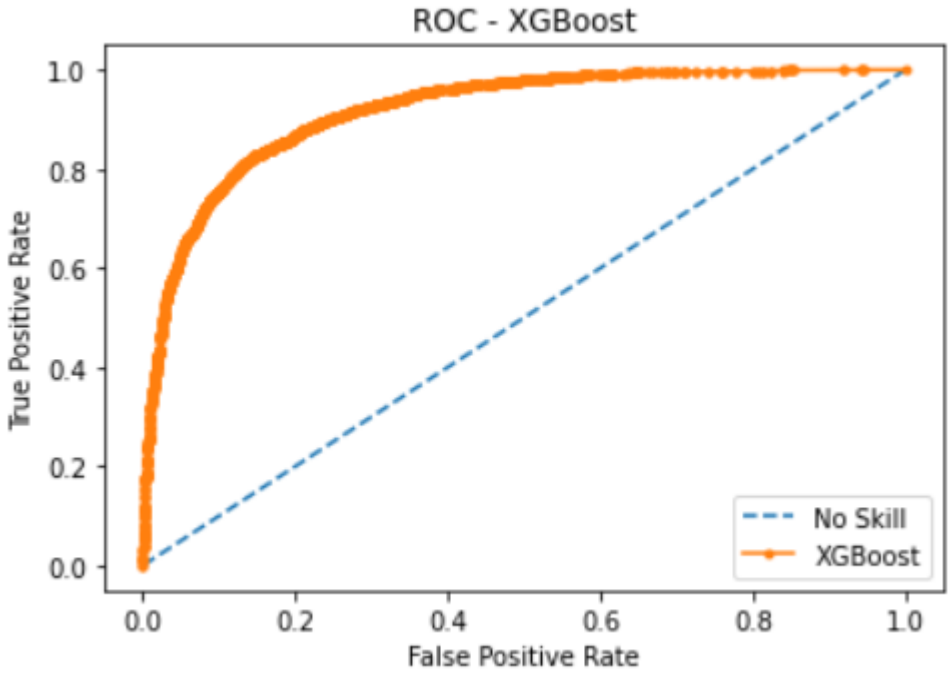
- 10 Fold CV Accuracy:78.57%
- AUC:0.877\

3.1.2) SVM



- 10 Fold CV Accuracy: 80.68%
- AUC:0.875

3.1.3) XgBoost

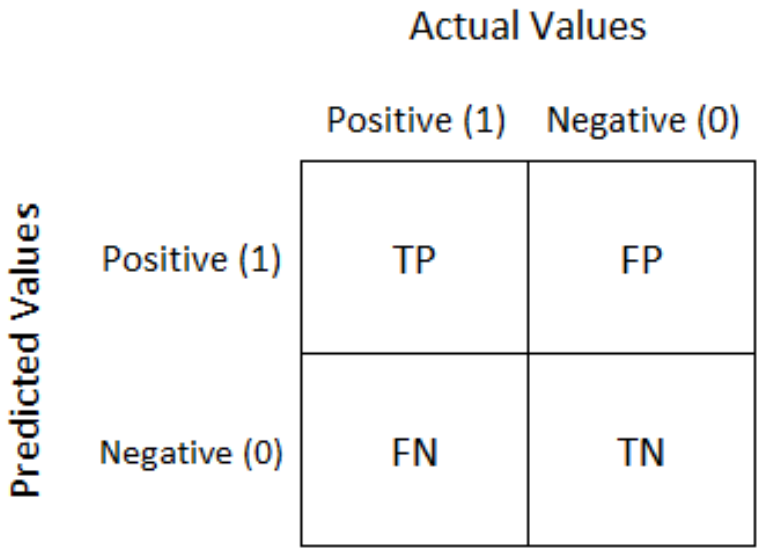


- 10 Fold CV Accuracy: 84.4%
- AUC:0.918

4) CASE STUDY: NEGATIVE DATA

Negative data plays a huge part in the model; the amount of negative data generated alone can affect the model a lot. Most of the researchers use 10 times the negative data. In this study, I will show you the effects of different sizes of negative data using the confusion matrix.

Note: All tests were done on Random Forest



Confusion Matrix

0.5 times negative data:

Predicted///Actual	1	0
1	7074	2653
0	55	895

I can clearly see that the model is biased towards positive data and predicts most of the data to be positive. But if it predicts a molecule to be negative, it does so with high accuracy.

Would be used if False Negatives mattered a lot.

1 times negative data:

Predicted///Actual	1	0
1	5687	1411
0	1433	5705

I can clearly see that the model is not biased towards positive or the negative data and predicts.

Would be used when both False Negatives and Positives matter equally.

3 times negative data:

Predicted///Actual	1	0
1	2455	4626
0	76	21315

I can clearly see that the model is biased towards negative data and predicts most of the data to be negative. But if it predicts a molecule to be negative, it does so with high accuracy.

Would be used if False Positives mattered a lot.

For my application, as both False and True Negatives play a huge role. Hence, I went with 1 times negative data.

5) CASE STUDY: COVID 19

Using the model I built, I scanned the entire database of approved drugs and predicted 248 of them to be a possible cure of COVID-19.

I apologize that I cannot disclose more information about this except for the fact that me and other researchers in the team had very similar results.

6) CASE STUDY: HDAC

Histone deacetylase is among the most promising therapeutic targets for cancer treatment and is well researched in the field of medicine. There are 14 approved cures/inhibitors of HDAC. Out of the 14, my model correctly identified 8 of them from the database of drugs, hence showcasing how powerful the model I build really is.

7) CONCLUSION

I developed a predictive model from scratch which would predict Drug-Target Interaction using various Machine Learning algorithms like Random Forest, SVM and XgBoost and achieved incredible ROC values and accuracy by performing 10 fold cross-validation. This significant ROC is achieved because of the use of an extra set of features which are the binding site features. It demonstrates that the findings of my proposed model will enrich the future research, especially in the drug-target interaction area. I also studied various sizes of negative data, came to the conclusion that the size of it depends on your application and made predictions regarding the cure for COVID-19; hopefully, a cure is found soon.