Final Report

Edit New page

Bradley, Elaine Vivian edited this page 5 minutes ago · 14 revisions

Final Report:

Introduction & Background:

The binding affinity of drug molecules with target proteins is crucial in pharmaceutical development. Traditional methods, such as high-throughput screening through wet-lab experiments are costly and limited when processing large sets of drug candidates. Machine learning models that accurately predict drug-target interactions (DTI) offer a valuable alternative.

Recent work includes a model that combines protein and drug feature selection through Incremental Wrapper Subset Selection with replacement method, and Rotation Forest classifier, achieving an accuracy of up to 98.12% [1]. However, it faces challenges with high-dimensional data and class imbalances and was tested on a narrow dataset.

We will use the publicly accessible BindingDB dataset, containing more than 20,000 experimentally determined binding affinities of protein-ligand complexes. It includes data from 110 protein targets and over 11,000 small molecule drug compounds [2]. The dataset is at http://www.bindingdb.org.

Problem Definition:

Pharmaceutical development involves understanding how drugs will interact with intended target proteins. Current processes for understanding DTI require extensive, costly testing, which can lead to the delay in release of medication to market as well as consumer costs. Implementation of a prediction model for DTI, will reduce time and financial costs of pharmaceutical developments, allowing for faster, more cost-effective development.

Methods:

Data Preprocessing:

The drug-target binding affinity dataset we used was retrieved from BindingDB_2020 [2]. This dataset, downloaded from Kaggle, is organized by affinity measurement type (Ki, Kd, IC50, and EC50) and formatted to include only five most relevant columns: drug_id, target_id, smiles, target_seq, and affinity. They also underwent several preprocessing steps: multi-chained proteins were excluded, as were entries without a Uniprot ID or an affinity level. For duplicate drug-target pairs, only the entry with the higher affinity label was kept. The affinity label was transformed to a float, using the formula 9-log(affinity).

For our analysis, we chose EC50 as the binding affinity metric because it has balanced affinity distribution and a relatively large number of observations. EC50 represents the concentration needed to achieve 50% of the drug's maximum effect in a biological system. Lower EC50 values are typically preferable when evaluating drug potency, as they indicate the drug is effective at a lower concentration. The EC50 dataset contains 163,745 drug-protein pairs, covering 119,715 drugs and 1,345 proteins.

To convert the SMILES string of each drug into a molecular object, we used the 'MolFromSmiles' function from the RDKit package [3]. This molecular object is then used to generate a binary fingerprint vector that represents the molecule's structural features. The 'GetMorganFingerprintAsBitVect' function creates a 2048-bit binary vector, where each bit represents the presence or absence of specific substructures or atom environments in the molecule.

The protein sequences were also converted into amino acid compositions using the 'sequence_to_composition' function. This function divides the count of each amino acid by the sequence length to get its relative abundance, generating a feature vector based on the 20 amino acids. Finally, these fingerprint and composition features are concatenated, forming a combined feature array that represents both the drug's SMILES string and the target's sequence composition.

Principal Component Analysis (PCA):

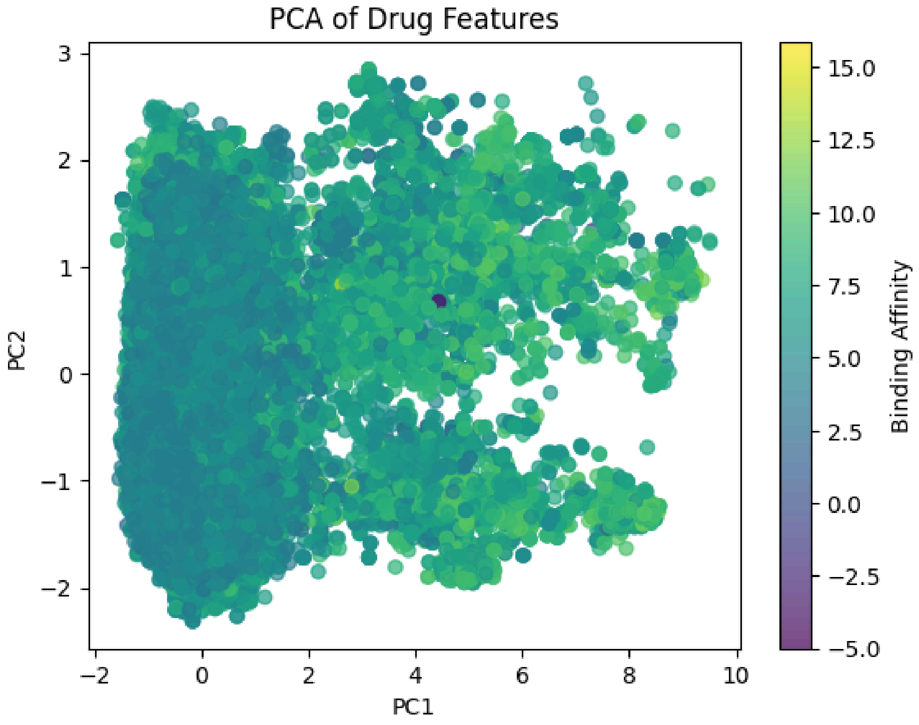
To reduce the dimensionality of our feature space and improve computational efficiency, we applied Principal Component Analysis (PCA) to both drug and target features. PCA identifies principal components (PCs) that capture the most significant variance in the dataset, effectively reducing noise and redundancy in the features, making the data more manageable for the model.

For drug features, which were represented as 2048-bit binary Morgan fingerprints, we retained 1,000 PCs that collectively explained 90.45% of the variance in the drug feature space. Despite this high cumulative variance, each individual component contributed only a small amount, with PC1 accounting for ~3-4% of the variance and PC2 for ~1-2%. This pattern indicates that variance is distributed across many components, emphasizing the high dimensionality and sparsity of the original fingerprint data.

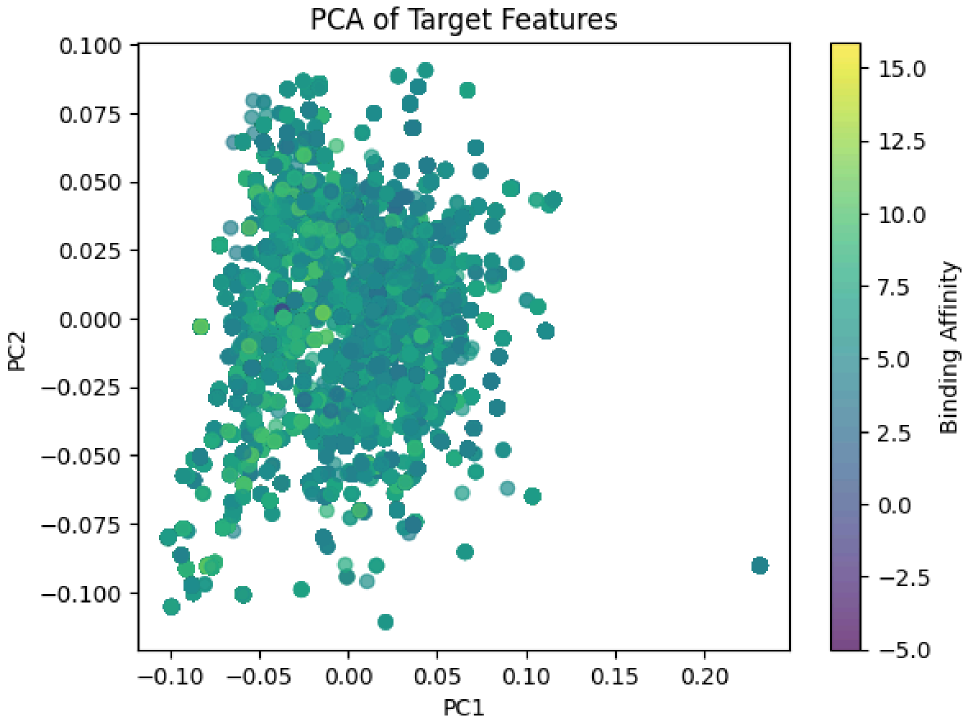
+ Add a custom sidebar

Clone this wiki locally

https://github.gatech.edu/el



For the target protein features, we retained 15 principal components. These 15 PCs explained 96.65% of the total variance, with PC1 contributing ~3.2% and PC2 ~2.0%. Similar to the drug features, the variance in target features is spread across multiple components, highlighting the complexity of protein sequences.



While PCA reduces computational complexity and storage requirements, it might discard some critical information that the model requires for accurate predictions. Using the PCA-transformed features with our neural network model resulted in a test loss of 0.96, compared to 0.59 when using the original, untransformed features. The lower predictive performance shows that in complex datasets such as drug-target binding affinity, even small contributions from higher-dimensional features may encode critical information or subtle patterns that are essential for accurate predictions. Therefore, careful consideration of the number of components and alternative dimensionality reduction techniques is crucial to balance efficiency and predictive accuracy.

ML Algorithm:

Supervised Model - Random Forest Regressor:

The machine learning component of our project begins with model selection and training, where we utilize a Random Forest Regressor as our primary model. Random Forest is advantageous due to its capacity to handle high-dimensional data, manage non-linear relationships, and resist overfitting—a common challenge in predicting complex drug-target binding affinities. The model leverages both drug and protein sequence features extracted from the dataset, which we split into training and testing subsets in an 80-20 ratio. This split ensures that our model is tested on unseen data, providing a robust measure of its generalizability and performance.

After the feature extraction and data splitting, we standardize all features through standard scaling. This scaling step is essential because it normalizes the ranges of different features, ensuring that they contribute equally during model training. It also reduces the potential influence of outliers on the final model, which is critical for maintaining predictive accuracy across diverse drug-target pairs.

To further improve our model's performance, we apply GridSearchCV for hyperparameter tuning. This approach allows us to optimize key parameters, including the number of decision trees in the forest (n_estimators) and the maximum depth of each tree (max_depth). By fine-tuning these parameters, we aim to minimize the model's mean squared error (MSE) and achieve a high R-squared (R^2) score, ensuring the best configuration for accurately predicting binding affinity.

Finally, we evaluate the model using Mean Squared Error (MSE), R-squared (R^2), and Mean Absolute Error (MAE) metrics. Each of these metrics offers unique insights: MSE reflects the model's prediction accuracy by providing an average of squared errors; R^2 shows how well the model explains the variability in binding affinities; and MAE captures the average absolute differences, giving us a view of the typical prediction error. Through this comprehensive evaluation, we can identify the best model configuration to accurately predict drug-target binding affinity within the high-dimensional and complex dataset. This concludes our first approach to training a supervised model for this task.

Another approach we used to train the Random Forest Regressor model was to utilize a single feature as the input: the concatenation of the embedded version of the SMILES sequence of the ligand with the decomposed amino acid composition of the target sequence.

We used the AllChem.GetMorganFingerprintAsBitVect function from the RDKit library to convert the SMILES sequence into a 2048-bit vector, representing an embedded version of the ligand's structure. For the target sequence, we created a function that counts the occurrences of each amino acid in its sequence and converts this information into a numerical vector.

These two vectors (the SMILES fingerprint and amino acid composition) were horizontally stacked using np.hstack() to form a single feature vector. This combined feature was then used as input to train our Random Forest model. We employed an 80-20 train-test split and experimented with different hyperparameters, such as n_estimators and max_depth. Due to computational limitations in Google Colab, we couldn't use grid search for hyperparameter tuning this approach, as it was resource-intensive and led to timeouts. Instead, we explored different combinations manually. This approach resulted in an MSE as low as 0.514. Further improvements could be made by exploring additional hyperparameters or incorporating features from our previous methods into this approach.

Supervised Model - Neural Network:

For our second supervised model, we chose to train a neural network on our data to predict the drug-target affinities. For the initial representation of the data we chose the same method of using the RDKit library to convert the SMILES sequence into a 2048-bit vector, and for the target sequence, we used the previously created function that counts the occurrences of each amino acid in its sequence and converts this information into a numerical vector. We then converted these inputs into torch tensors to make the inputs compatible with the network designed in pytorch and used the Dataloader library to create dataloaders of batch size 32 for the network. Previously for the random forest regressor we stacked the two vectors horizontally to feed into the model as an input, however the input size for this had large variations in it. Since for neural networks we need to provide inputs of a fixed size, we explored padding this stacked vector. This however resulted in very very sparse vectors that consisted of upto 96% zeros. The resulting straightforward network also did not perform well at all since it was learning patterns in this noise that was dominant in the inputs instead of the actual data. To combat this we decided to not stack these vectors and instead feed the two input vectors into separate networks with their respective shapes and combine their outputs by feeding them into a final network. We used the following network architecture for the neural network:

`AffinityNN(

(drug_fc): Sequential(

```
  (0): Linear(in_features=2048, out_features=128, bias=True)

  (1): ReLU()

  (2): Dropout(p=0.3, inplace=False)
```

)

(target_fc): Sequential(

```
(0): Linear(in_features=20, out_features=64, bias=True)

(1): ReLU()

(2): Dropout(p=0.3, inplace=False)
```

)

(fc_combined): Sequential(

```
(0): Linear(in_features=192, out_features=64, bias=True)

(1): ReLU()

(2): Dropout(p=0.3, inplace=False)

(3): Linear(in_features=64, out_features=1, bias=True)
```

)

)`

As we can see above, the drugs and target are passed through a single linear / fully connected layer with a ReLU activation. A dropout layer with probability of 0.3 was added in order to reduce overfitting and reduce reliance on particular neurons. The output features for each of these networks were set at 128 and 64 respectively and the combined 192 features were fed into a combined network where the were passed through a single linear layer, ReLU activation and dropout layer before being passed through the final linear layer that predicts the affinity. This approach ended up working out well and the model was able to learn the patterns in data with the MSE loss lowering upto 0.4565 before it started overfitting. This ultimately surpassed the performance of the random forest regressor however it was also very quick and easy to overfit.

Since there wasn't too much improvement in performance before overfitting, we believe that we might need more data / features or perhaps more advanced feature engineering combined with domain knowledge to achieve a breakthrough in lowering the MSE loss further.

Unsupervised Model - KMeans:

For our unsupervised model, we chose to utilize a KMeans algorithm to cluster our dataset, which consists of the SMILES Morgan Fingerprint of each drug molecule and the Amino Acid composition of the target protein for each drug-target pair. Since the KMeans algorithm is an unsupervised model, we do not utilize the drug-target affinity as part of the input parameters for training the model; however, we are able to utilize the drug-target affinity for evaluating the effectiveness of the clustering.

The features that our KMeans model utilized for training were the SMILES Morgan Fingerprint of the drug molecule, and the Amino Acid composition of the target protein. The Morgan Fingerprint is a 2048 binary bit vector which is an encoded representation of which chemical structures are present in the drug molecule under consideration. This means that each drug molecular structure is represented by 2048 features. The Amino Acid composition is an array, representing the percentage of each amino acid in the target protein sequence. There are 20 different amino acids, so the Amino Acid composition is represented by 20 features. Since the Morgan Fingerprint and the Amino Acid composition were the only inputs into our model, the model had 2068 features to consider for its clustering.

Utilizing a KMeans model with high-dimensionality has several risks due to the "curse of dimensionality" and higher risk for including noisy features that don't benefit the model [4]. The curse of dimensionality is explained as the negative impact that high-dimensional data can have on machine learning algorithms [4]. In order to improve the performance of our KMeans model, we eliminated noisy data in two ways.

The first method was through data preprocessing, where we converted the SMILES data and Target Protein Sequence into the Morgan Fingerprint bit array and Amino Acid composition array. After converting the data into a format that our model can understand, we removed any rows of data with "NaN" data entries and we did not include data where the Morgan Fingerprint could not be created from the SMILES input. After the dataset was cleaned of noisy data, we standardized the cleaned feature matrix by utilizing the StandardScaler() function from the sklearn library. By standardizing our data, we ensure that all features are able to contribute to the clustering of our data.

The second method to reduce the impact that high-dimensional data has on our KMeans model, is to implement Principal Component Analysis so that the KMeans algorithm only utilizes the most impactful features when it comes to clustering. By eliminating features that contribute meaningful information, we reduce the noise of the dataset. This allows the KMeans algorithm to prioritize clustering of information that is the most meaningful, ideally resulting in better clustering that aligns with the known drug-target affinity for each drug and target protein combination.

In order to compare the performance of each KMeans clustering model, we ran the KMeans algorithm on our dataset while varying the number of features as well as the number of clusters the algorithm utilized. The KMeans model was executed with the following number of features:

- No PCA applied, full dimensions - 2068 features
- PCA applied - 30 features
- PCA applied - 50 features
- PCA applied - 100 features

Each of the number of features were executed with the following number of clusters:

- 5 clusters
- 10 clusters
- 15 clusters
- 20 clusters
- 40 clusters

- 80 clusters
- 200 clusters
- 500 clusters
- 1000 clusters
- 1500 clusters

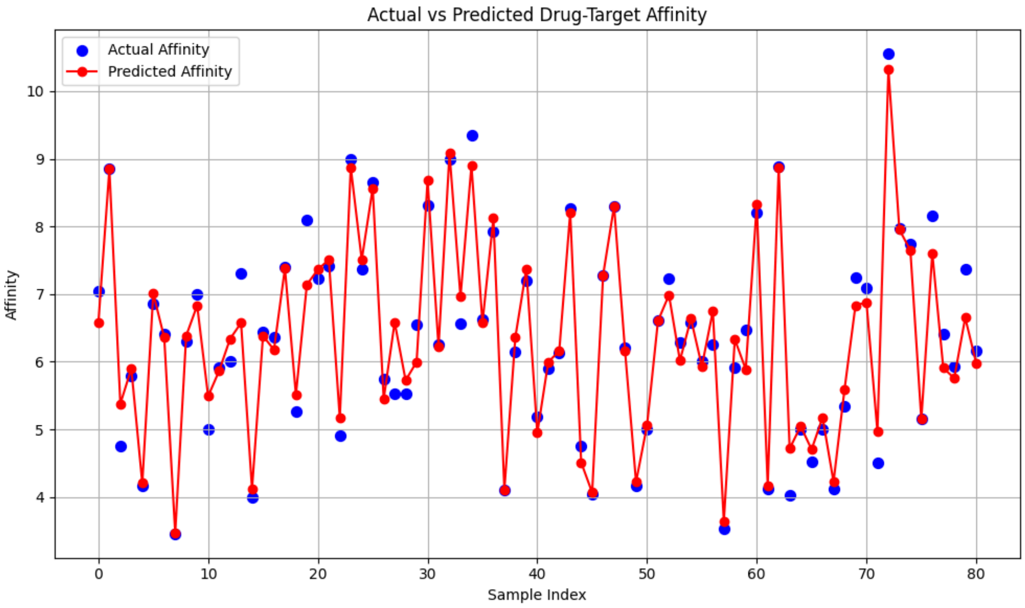
Based on our feature number and cluster number combinations, we executed the KMeans algorithm with 40 different combinations of input parameters. The inclusion of the full feature dataset with 2068 features allows us to have a baseline execution with no dimensionality reduction to compare our dimensionally reduced results with.

Results and Discussion:

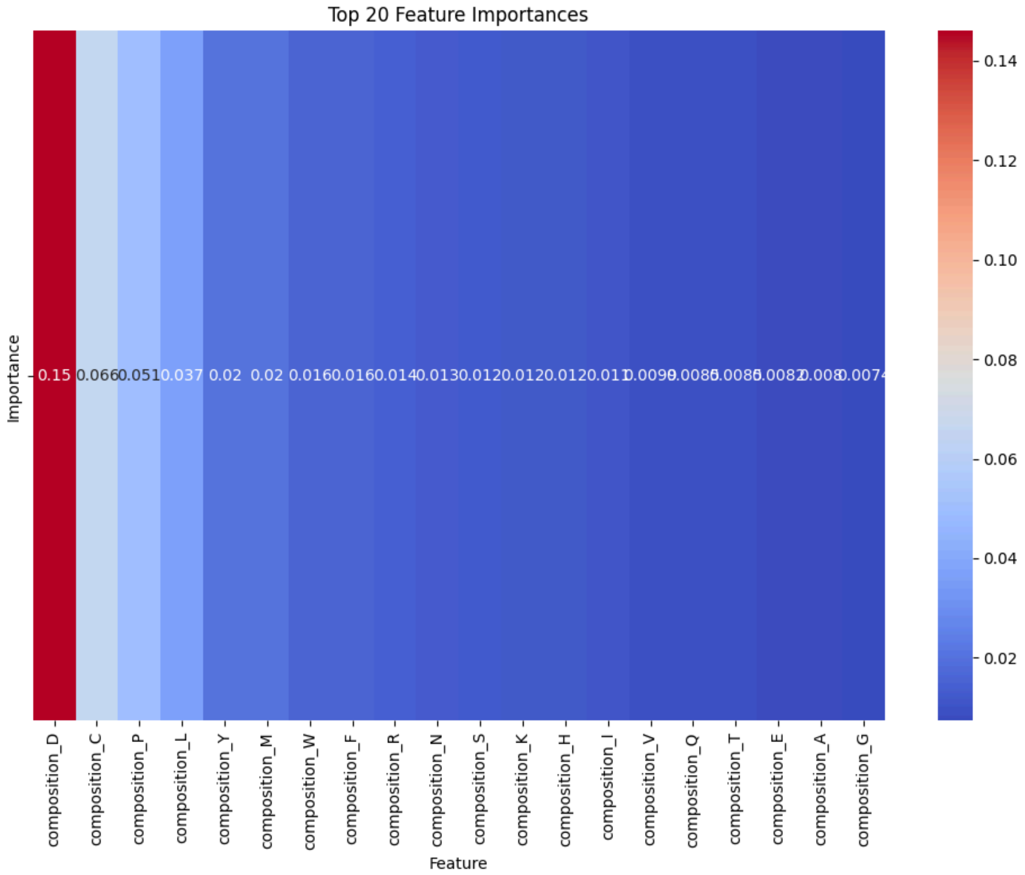
Visualizations:

Visualizations for Random Forest Regressor:

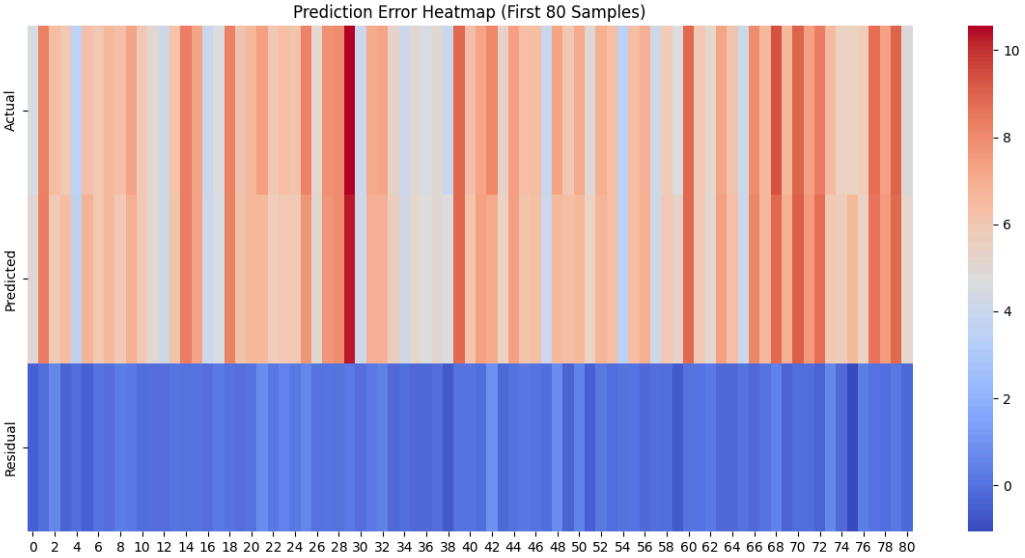
In order to visualize the results of our Random Forest Regressor, we decided to demonstrate the performance of our model through a visual comparison of the actual vs predicted drug-target affinity, a display of the top 20 features of our dataset with their importance to our model, a prediction error heatmap for the first 80 samples in our dataset, and the comparison of the MSE loss with the depth of our Random Forest.



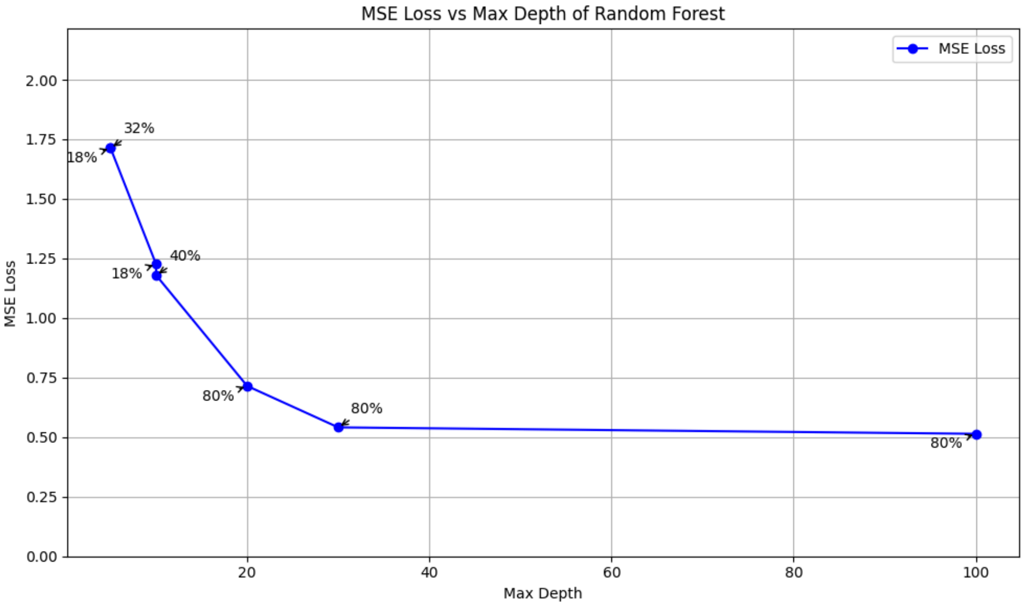
When visualizing the actual vs predicted drug-target affinity, we selected 0.05% of our dataset, resulting in a sample size of 80 predicted data points and their associated true values. In the visualization of the actual and predicted affinity, the predicted affinity has been overlaid onto the actual affinity so that the accuracy of our model for each individual datapoint can be better understood. The further the actual and predicted data points are separated on the graph, the more inaccurate our prediction is.



For the visualization of the importance of each feature, the feature importance was first calculated from our dataset using the .feature_importances_ attribute of the Random Forest Regressor model from scikitlearn, which is calculated as the average reduction of Mean Squared Error (MSE) for the entire forest. Our heatmap of the importance of the top 20 features reveals that the most important features in our model are the compositions of the 20 amino acids that make up the target protein sequence. This means that the composition of the target protein had the most significant impact on the prediction of the target affinity for each drug molecule. As seen in the visualization, the composition of amino acid D had the most impact in the reduction of the MSE for our predictions.



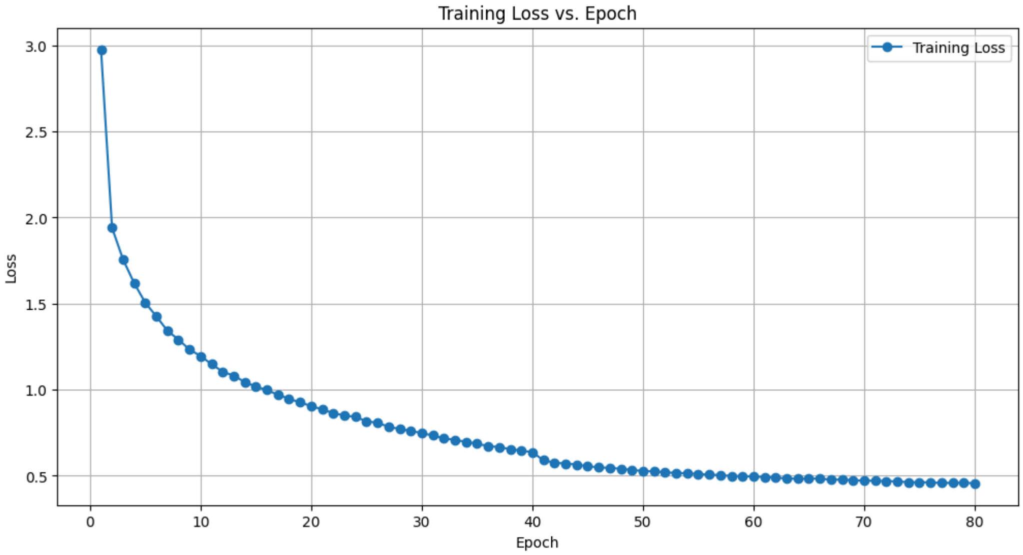
The prediction error heatmap provides a visual representation of the values of the predicted affinity values, the actual affinity values, and the residuals for the first 80 samples. In this instance, the residual represents the difference in the predicted and actual affinity values. For our visualization, a dark blue residual value indicates a negative residual value, which means that our model over-predicted the affinity value. A light blue residual indicates a positive residual value, which means that our model under-predicted the affinity value. We have included the visual representation of predicted and actual values as another means of reflecting the residual. When the predicted and actual affinity values are the same, the color of the predicted and actual will also be the same, indicating an accurate prediction and a residual of 0. The data range of our heatmap varies from approximately -1.5 to 10.5, in order to accommodate both the residual values and the affinity values for each drug-target protein interaction.



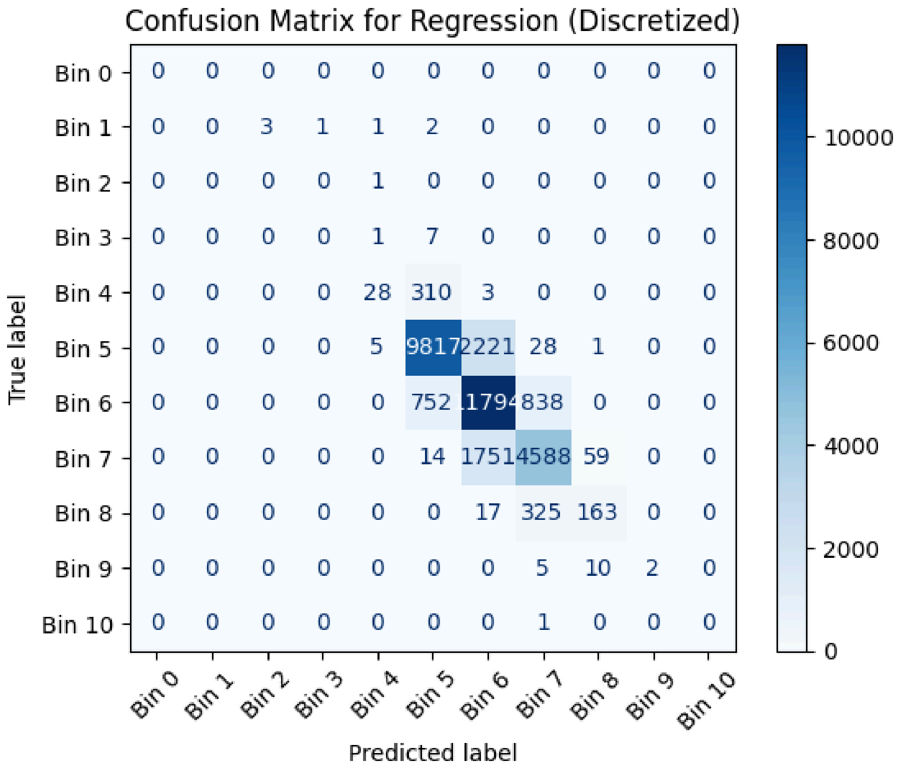
For our final visualization, we graphed the MSE Loss associated with the percentage of the dataset utilized for training and the depth of the Random Forest Regressor. The values for the MSE Loss are on the y-axis, the depth of our model is on the x-axis, and the percentage of the dataset utilized for training is marked as the label for the datapoint on the graph. For our model results utilizing 18% and 32% of our dataset to train with, and a forest depth of 5, it can be seen that our MSE Loss was almost the exact same, indicating that the size of our training dataset did not have an impact on the model performance at a depth of 5. For our model results utilizing 18% and 40% of our dataset to train with, and a forest depth of 10, it can be seen that we had a slightly lower MSE with a larger training dataset size, indicating that we saw a slightly better performance at a deeper depth with a larger training dataset. For our training dataset size at depths of 20, 30, and an unlimited depth (graphed as a depth of 100 for visualization), we utilized a larger percentage, 80% of our dataset, to train with. We saw significant improvement for our MSE with a depth of 20 and 30, but there was minimal improvement between a depth of 30 and the unlimited depth. This would indicate that the ideal depth to train our model at would be a maximum depth of 30.

Visualizations for Neural Network:

The visualizations of the neural network’s performance provide valuable insights into its training and prediction behavior. To analyze the training process, we plotted the loss versus epoch graph, which shows a steady decrease in training loss from approximately 3.0 to below 0.5 over 80 epochs. This indicates effective learning and convergence as the model optimizes its parameters.

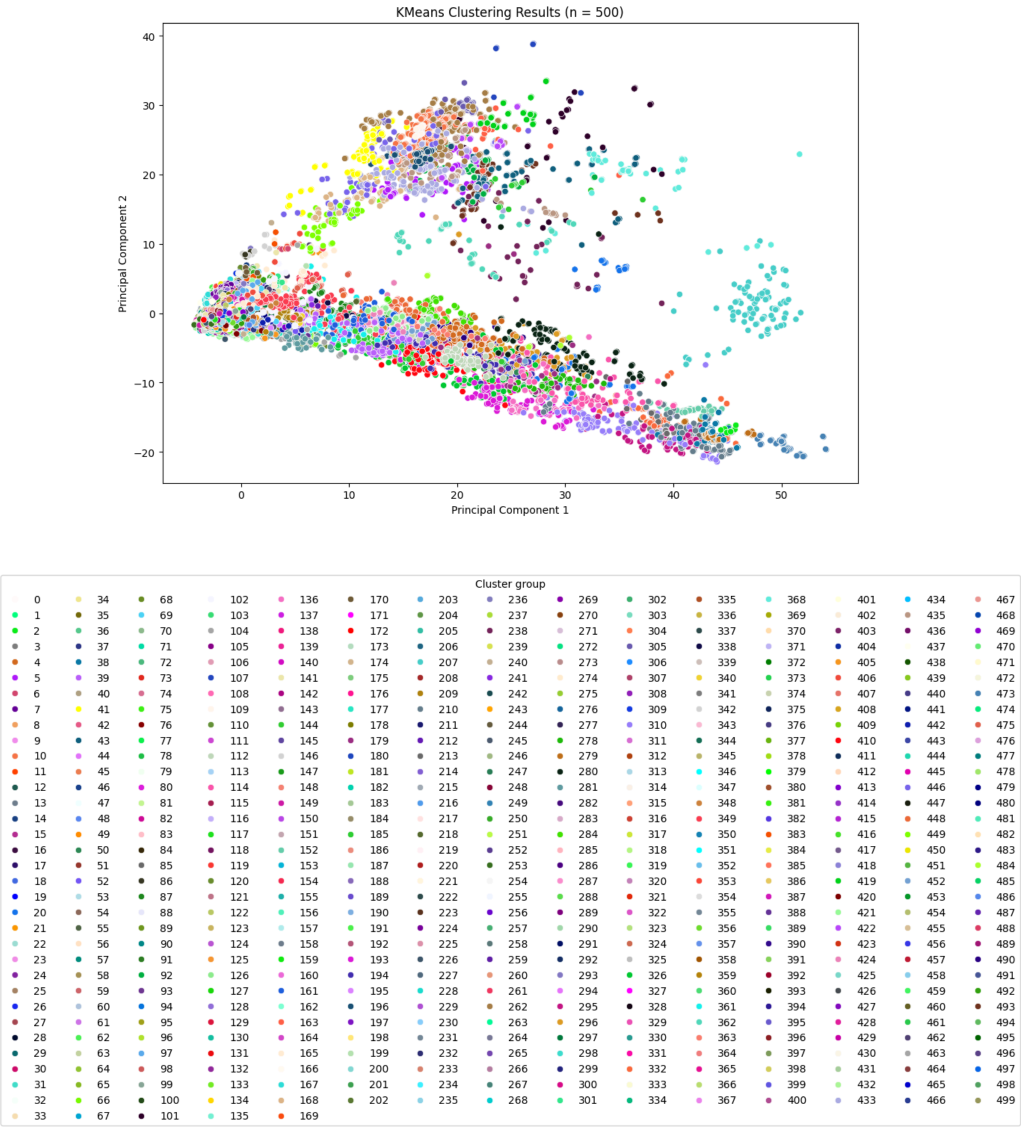


Additionally, the confusion matrix for regression, highlights the model’s prediction accuracy. A strong concentration along the diagonal in the central bins suggests that the model performs well for mid-range values, with the highest density in bins 5 and 6. However, there are notable misclassifications into neighboring bins, reflecting slight over- or under-predictions. The sparsely populated edge bins (e.g., bins 0, 1, 9, and 10) suggest challenges in predicting extreme values, likely due to data scarcity in those ranges. These results indicate that while the model is well-trained for central values, further refinement is needed to handle outliers and reduce neighboring bin errors effectively. The visualizations clearly demonstrate both the strengths and areas for improvement in the model.

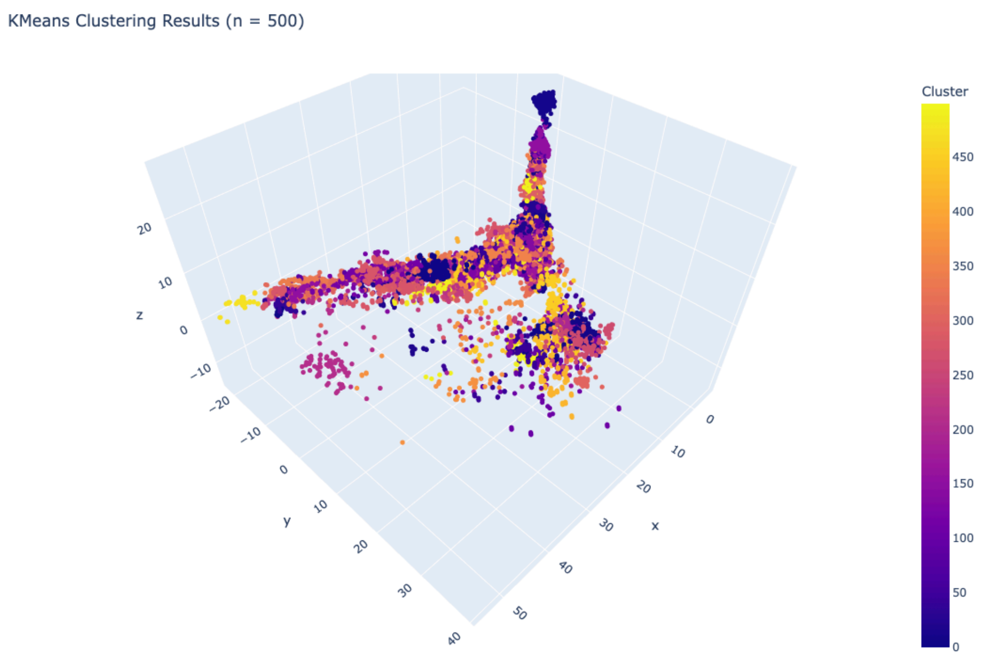


Visualizations for KMeans:

In order to visualize the results of our KMeans, we plotted both 2-dimensional and 3-dimensional scatterplots of our dataset with their cluster label identified via color. In order to plot the high-dimensional data in a 2-dimensional and 3-dimensional space, we utilized PCA to determine the top 2 and top 3 principal features for the 2-dimensional scatterplot and 3-dimensional scatterplot respectively.



The 2-dimensional visual representation of our results can be difficult to analyze due to the dense clustering of our dataset, which results in several data points being graphed on top of each other. In addition, when the number of clusters is large, such as the visualization above with 500 clusters, it becomes difficult to interpret which cluster is which.



By plotting our results in a 3-dimensional space, we're able to distinguish the individual data points from each other, which gives us a better indication of the effectiveness of the clustering from our KMeans algorithm. There is still a bit of a visual limitation when it comes to assigning different colors to each cluster that are visually distinct. If there are two clusters next to each other with a similar color, it may appear as one larger cluster rather than two separate clusters.

Quantitative Metrics:

The quantitative metrics utilized to evaluate the performance of our Random Forest Regressor model include Mean Squared Error (MSE), R-squared (R^2), and Mean Absolute Error (MAE). The primary metric that we focused on for optimization for this portion of our project was MSE. There are several reasons that we focused on evaluating MSE over R-squared or MAE for this round of development due to MSE's sensitivity to outliers and the interpretability of MSE.

MSE is a metric that is sensitive to outliers in a dataset due to the fact that the equation for MSE involves squaring the errors between the predicted and actual drug-target affinity values. When a larger error is squared, we have an exponential increase in the error value that gets included in the average performance of our model, thus increasing the overall value of MSE which would indicate a poor performance of our model. If we trained our model with a dataset that contained significant outliers, it would have a detrimental impact on our model's ability to predict the drug-target affinity for our test data and for any new drug-target interactions presented to our model. In order to guarantee that our model is not overfitted to noisy or abnormal data, we wanted to gauge the performance based on a metric that would reflect this.

Due to the fact that Random Forest Regression models utilize MSE to guide the training of the model, this means that MSE is a great metric to evaluate the overall performance of the model as well. The goal of the Random Forest Regressor is to split the overall dataset amongst a “forest of trees” with various bootstrapped subsets of the overall data. Each decision tree then splits the data at each node of the tree prioritizing different features of the dataset, and evaluating the feature performance through minimization of MSE before the node and after the node in each child. When optimizing our Random Forest Regression model, we varied the percentage of the dataset used to train, as well as the maximum depth of the forest. To gauge the performance of optimizing these parameters, a decrease in the MSE of our trained model would indicate better parameters for optimization.

The quantitative metric utilized to evaluate the overall effectiveness of our KMeans models was the Weighted Average Standard Deviation of the drug-target affinity per cluster in the model. Each cluster utilizes the standard deviation of the affinity values in the cluster to determine how well-fit the data points are in the cluster. A cluster that is well-fitted for predicting the drug-target protein affinity will have a lower standard deviation indicating minimal variance in the cluster. It’s important to take the weighted average of each cluster’s standard deviation, rather than just the average of each cluster’s standard deviation to account for the distribution of data. If we just took the average standard deviation, the result could be skewed by tightly fitted clusters with few data points in the cluster while a significant portion of the data may have ended up in only one or two of the clusters with a high standard deviation.

Analysis of the 3+ Algorithms/Models:

Analysis of the Random Forest Model:

We used quantitative metrics described above and feature importance to examine the performance of our Random Forest Regressor model in predicting drug-target affinity. Using MSE as our primary metric, we observed that models with greater maximum tree depth generally improved performance, with MSE decreasing as the model complexity increased. Specifically, when model depth increased from 5 to 10, MSE dropped significantly from ~1.713 to ~1.180-1.226. This improvement continued as we tested greater depths, with an MSE reaching as low as 0.514 at an unrestricted depth. This suggests that the model benefited from greater complexity, although the diminishing decrease in MSE’s drop indicates a potential trade-off between model depth and overfitting.

In terms of feature importance, the amino acid composition within the target protein sequence proved to be crucial, with the amino acid D having the greatest effect on reducing MSE. The findings indicate the central role protein structure could have in influencing drug-target interactions.

The analysis of residuals revealed some patterns in prediction errors, with certain interactions showing consistent over or under -predictions. This suggests while the model was able to capture the data overall and many key relationships, it may require further tuning to address specific prediction biases that we currently see. Overall, the Random Forest Regressor demonstrated strong predictive capability, effectively utilizing protein composition features and improving with increased model complexity and larger training datasets.

Analysis of the Neural Network Model:

The neural network model demonstrated strong performance in predicting drug-target binding affinities, achieving a minimum MSE of 0.4565 before overfitting began to emerge. This surpassed the best MSE achieved by the Random Forest Regressor, highlighting its ability to learn complex patterns in the data. The architecture leveraged separate pathways for drug and target features, ensuring that the distinct characteristics of these inputs were effectively processed and combined. By avoiding the sparsity issues associated with horizontally stacked inputs and employing dropout layers with ReLU activation, the model reduced overfitting and improved generalization. However, the neural network showed a tendency to overfit rapidly, suggesting a need for either additional regularization or a larger dataset to further improve performance. Despite this limitation, the network successfully captured intricate relationships in the data, demonstrating its potential for tasks involving high-dimensional, non-linear interactions.

Analysis of the KMeans Model:

Since the KMeans model is unsupervised, we did not provide the drug-target affinity values as inputs to the model. For our dataset, the affinity values are essentially our dataset labels and can be utilized to evaluate the effectiveness of our KMeans clustering. In order to evaluate the performance of our clusters, we calculated Weighted Average Standard Deviation (WASD) for the affinity values per model. When the standard deviation value is low, that means that the affinity values in the clusters are close to each other with minimal variance. Ideally, we would like to see a standard deviation of 1 or lower to indicate good performance of our clustering model.

As would be expected, when comparing the performance of each model with varying feature number and cluster number combinations, we saw better weighted average standard deviation values as the number of clusters increased. After our data preprocessing was completed, we were left with 163,745 drug-protein combinations, and the maximum number of clusters we utilized was 1500 clusters, which is 0.92% the size of the cleaned dataset. Theoretically, we could continue to increase the number of clusters to improve the overall performance of the KMeans algorithm. A KMeans model with 163,745 data points and 163,745 clusters would have a standard deviation of zero, but that would be extremely overfitted and wouldn’t provide any valuable insight or predictions. Instead, we want to balance increasing the number of clusters to improve our WASD while minimizing the number of clusters as much as possible.

Looking at the results of the KMeans models, the KMeans model with the best performance was the model with 100 features and 1,500 clusters, which yielded a WASD of 1.1414. Our model with 30 features and 1,500 clusters yielded a WASD of 1.1663, and our model with 2068 features and 1,500 clusters yielded a WASD of 1.1690. The fact that the model with 100 features performed better than the model with only 30 features implies that the top 100 features don’t contain very much noise and provide value to the clustering of the data points. We can confirm that some of the features in the original dataset of 2068 features do contain some level of noise considering the model with 100 features performed better than the model with 2068 features. This reflects that either extreme of too many features or too few features can have a negative impact on the performance of the KMeans model.

Overall, KMeans may not be the best model to utilize for predicting drug-target protein affinity scores since the Morgan Fingerprint and Amino Acid compositions contain a large number of features. Although we can reduce the number of features, in order to address the “curse of dimensionality”, the amount of dimensions we can reduce while maintaining good performance of our model is limited.

Comparison of the 3+ Algorithms/Models:

Both the Random Forest Regressor and the Neural Network models effectively leveraged the provided features, but they exhibited distinct strengths and limitations. The Random Forest Regressor excelled in handling structured and interpretable features, such as amino acid composition, and showed significant improvements with increased model depth, achieving an MSE of 0.514. It also provided insights into feature importance, emphasizing the critical role of certain amino acids in protein-drug interactions. In contrast, the Neural Network surpassed the Random Forest in predictive accuracy, with a lower MSE of 0.4565, by effectively learning complex, non-linear relationships in the data. However, it was more prone to overfitting, highlighting its dependency on robust regularization techniques and sufficient training data.

The KMeans model, being unsupervised, utilized a different metric for performance than our two supervised models, which was Weighted Average Standard Deviation of the drug-target affinities rather than MSE. This difference in quantitative metrics makes it difficult to compare the performance of the KMeans model directly to the Random Forest Regressor and the Neural Network model. Due to the non-linearity of the dataset, as noted in the positive performance of the Neural Network model, the KMeans model struggled with clustering the high-dimensional data. One solution for non-linear data would be to increase the dimensions of the data to discover linear relationships; however, our dataset already had a large amount of features, which is not beneficial to a KMeans model. One of the few ways that the KMeans model saw improvement in its performance was increasing the cluster size to just under 1% of the size of the overall data set, at 1500 clusters. However, increasing the number of clusters would make KMeans prone to overfitting, similar to the Neural Network model. In contrast to a Neural Network, there are fewer methods to prevent overfitting while improving accuracy of the KMeans model.

Overall, it appears that the KMeans algorithm failed to identify clustering patterns that aligned with the drug-target affinity with such high-dimensional data despite using feature reduction with Principal Component Analysis. While the Random Forest model offered better interpretability and stability, the Neural Network demonstrated superior capacity for capturing intricate patterns, making it better suited for tasks involving high-dimensional data. Combining the strengths of both the Random Forest and Neural Network models, such as using feature importance from Random Forest for feature engineering in the Neural Network, could further enhance predictive performance.

Next Steps:

Building on the current analyses, our next steps will focus on enhancing model performance and exploring new methodologies to address the observed limitations. For the supervised models, we will refine the Random Forest and Neural Network approaches by incorporating advanced feature engineering and data augmentation techniques. Specifically, we aim to use the feature importance from the Random Forest model to improve input representations for the Neural Network, potentially integrating additional chemical and biological features beyond the Morgan Fingerprint and amino acid composition. To mitigate overfitting in the Neural Network, we will explore advanced regularization techniques, such as L2 regularization and early stopping, and experiment with more complex architectures, such as attention mechanisms or graph-based networks, to better capture the chemical and biological relationships in the dataset.

For the unsupervised models, we will further investigate clustering methodologies to improve the interpretability and robustness of KMeans results. Given the limitations of KMeans in handling non-globular clusters, we will experiment with density-based clustering algorithms such as DBSCAN or hierarchical clustering to capture more complex relationships within the dataset. Additionally, we plan to evaluate the impact of alternative dimensionality reduction techniques, such as t-SNE or UMAP, on clustering quality. These methods may reveal hidden structures in the data that PCA might have overlooked. Finally, we will integrate supervised and unsupervised approaches, using clustering results to precondition the supervised models or using supervised predictions to validate the clusters, to gain deeper insights into drug-target interactions and further improve prediction accuracy.

References:

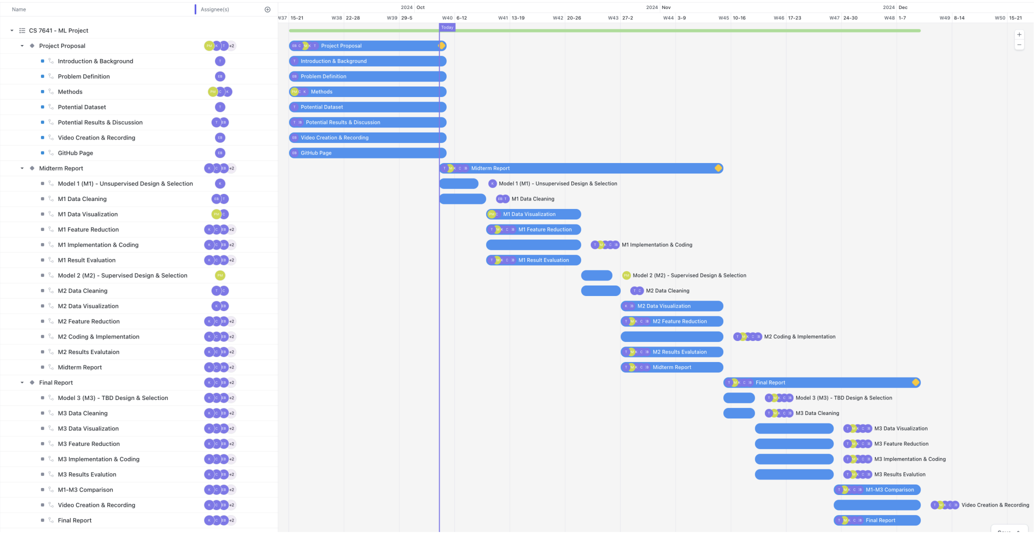
[1] Abbasi Mesrabadi, H., Faez, K. & Pirgazi, J. Drug–target interaction prediction based on protein features, using wrapper feature selection. Sci Rep 13, 3594 (2023). https://doi.org/10.1038/s41598-023-30026-y

[2] Tiqing Liu, Yuhmei Lin, Xin Wen, Robert N. Jorissen, Michael K. Gilson, BindingDB: a web-accessible database of experimentally determined protein–ligand binding affinities, Nucleic Acids Research, Volume 35, Issue suppl_1, 1 January 2007, Pages D198–D201, https://doi.org/10.1093/nar/gkl999

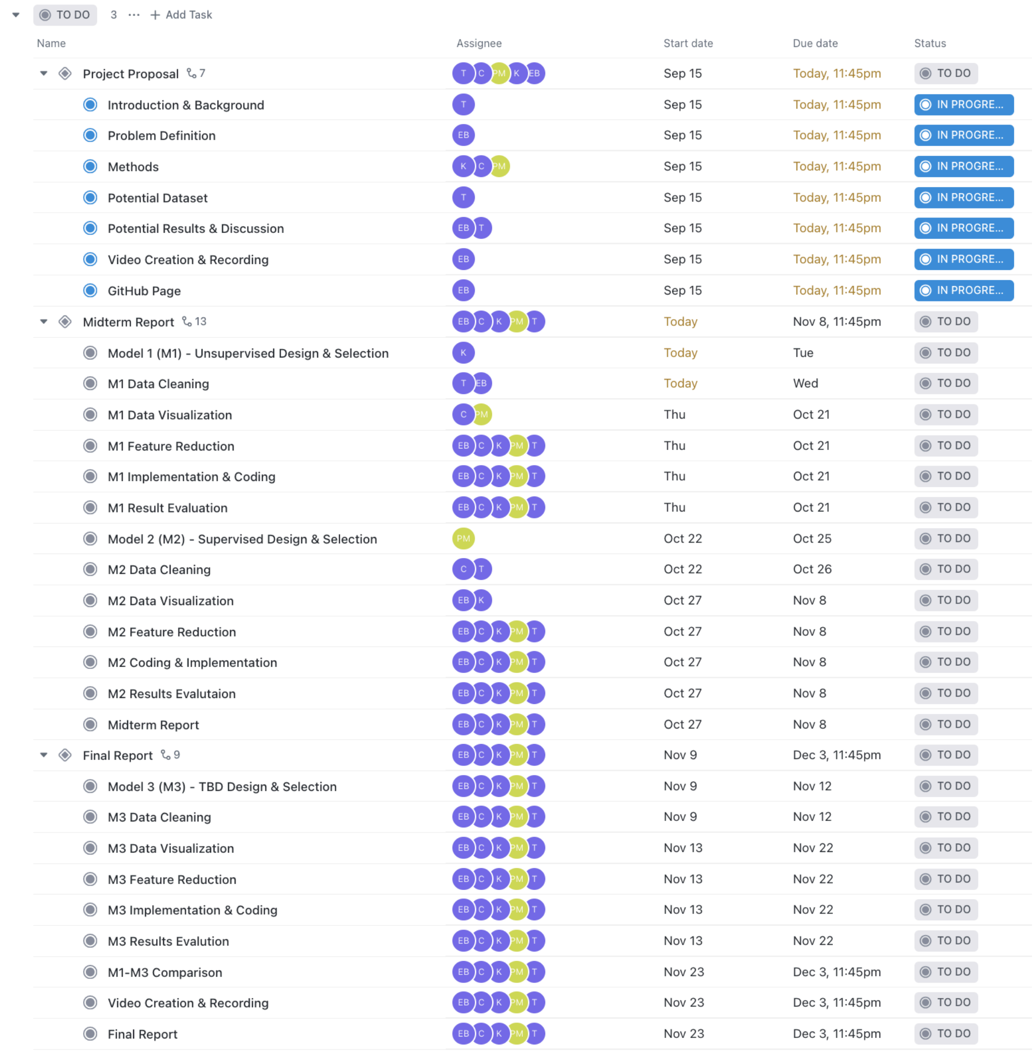
[3] Landrum, G. (2020). RDKit: Open-source cheminformatics. Retrieved from https://www.rdkit.org

[4] Curse of dimensionality in Machine Learning. GeeksforGeeks. (2024, April 3). https://www.geeksforgeeks.org/curse-of-dimensionality-in-machine-learning/

Gantt Chart:



List View of Gantt Chart:



Name	Assignee	Start date	Due date	Status
Project Proposal 7	T C PM K EB	Sep 15	Today, 11:45pm	TO DO
Introduction & Background	T	Sep 15	Today, 11:45pm	IN PROGRE...
Problem Definition	EB	Sep 15	Today, 11:45pm	IN PROGRE...
Methods	K C PM	Sep 15	Today, 11:45pm	IN PROGRE...
Potential Dataset	T	Sep 15	Today, 11:45pm	IN PROGRE...
Potential Results & Discussion	EB T	Sep 15	Today, 11:45pm	IN PROGRE...
Video Creation & Recording	EB	Sep 15	Today, 11:45pm	IN PROGRE...
GitHub Page	EB	Sep 15	Today, 11:45pm	IN PROGRE...
Midterm Report 13	EB C K PM T	Today	Nov 8, 11:45pm	TO DO
Model 1 (M1) - Unsupervised Design & Selection	K	Today	Tue	TO DO
M1 Data Cleaning	T EB	Today	Wed	TO DO
M1 Data Visualization	C PM	Thu	Oct 21	TO DO
M1 Feature Reduction	EB C K PM T	Thu	Oct 21	TO DO
M1 Implementation & Coding	EB C K PM T	Thu	Oct 21	TO DO
M1 Result Evaluation	EB C K PM T	Thu	Oct 21	TO DO
Model 2 (M2) - Supervised Design & Selection	PM	Oct 22	Oct 25	TO DO
M2 Data Cleaning	C T	Oct 22	Oct 26	TO DO
M2 Data Visualization	EB K	Oct 27	Nov 8	TO DO
M2 Feature Reduction	EB C K PM T	Oct 27	Nov 8	TO DO
M2 Coding & Implementation	EB C K PM T	Oct 27	Nov 8	TO DO
M2 Results Evalutaion	EB C K PM T	Oct 27	Nov 8	TO DO
Midterm Report	EB C K PM T	Oct 27	Nov 8	TO DO
Final Report 9	EB C K PM T	Nov 9	Dec 3, 11:45pm	TO DO
Model 3 (M3) - TBD Design & Selection	EB C K PM T	Nov 9	Nov 12	TO DO
M3 Data Cleaning	EB C K PM T	Nov 9	Nov 12	TO DO
M3 Data Visualization	EB C K PM T	Nov 13	Nov 22	TO DO
M3 Feature Reduction	EB C K PM T	Nov 13	Nov 22	TO DO
M3 Implementation & Coding	EB C K PM T	Nov 13	Nov 22	TO DO
M3 Results Evalution	EB C K PM T	Nov 13	Nov 22	TO DO
M1-M3 Comparison	EB C K PM T	Nov 23	Dec 3, 11:45pm	TO DO
Video Creation & Recording	EB C K PM T	Nov 23	Dec 3, 11:45pm	TO DO
Final Report	EB C K PM T	Nov 23	Dec 3, 11:45pm	TO DO

Excel View of Gantt Chart: CS 7641 - Group 32 - Project Gantt Chart.xlsx

Contribution Table:

Name	Proposal Contributions
Elaine Bradley	* Upload of files to GitHub * File explanation included in README.md * Authored the Unsupervised_Learning_Model_1_KMeans.ipynb file, which implemented the KMeans algorithm and the visualizations for the KMeans algorithm * Wrote the Unsupervised Model - KMeans section found under the ML Algorithm section of the Methods part of the Final Report * Wrote the Visualizations for Random Forest Regressor, Visualizations for KMeans, Quantitative Metrics, and Analysis of the KMeans Model sections found under the Results and Discussion part of the Final Report * Contributed to the Comparison of the 3+ Algorithms/Models under the Results and Discussion part of the Final Report
Katherine (Katie) Ferguson	* Researched and set up unsupervised learning mode in proposal * Contributed to Expected Results and Project outcomes * Helped edit final proposal and and reports * Wrote the Analysis of the 1+ Algorithm/Model and Next Steps sections found under the Results and Discussion part of the Midterm/Final Report * Created Powerpoint and Recorded Final Video
Chenyu Gu	* Researched and found the unsupervised learning models * Literature Review * Authored the Model_Drug Target Binding Affinity.ipynb file, which was one variation of the Random Forest Regression model * Authored the PCA & Visualize - Supervised Learning Model 2 Neural Network.ipynb, which implemented the visualization of Neural Network model * Contributed to parts about Random Forest Regressor ML Algorithm section found under the Methods part of the report * Wrote the visualization of NN model, analysis of the NN model, and comparison between NN and RF under the results part of the report * Wrote the Next Steps part of the report
Pratham Mehta	* Researched and found the supervised learning and deep learning based models * Literature Review * Authored the Supervised_Learning_Model_1_Random_Forest_Regression.ipynb file, which was one variation of the Random Forest Regression model * Authored the Supervised_Learning_Model_2_Neural_Network.ipynb file, which implemented the Neural Network to predict drug-target affinity * Implemented the visualizations of the Random Forest Regression model * Contributed to parts about Random Forest Regressor ML Algorithm section found under the Methods part of the report * Wrote the parts about the neural network under the ML algorithm section found under the Methods part of the report

Name	Proposal Contributions
Tamarin Tandra	* Literature Review * Authored the Pre-processing_Drug Target Binding Affinity.ipynb file, which was utilized to clean, stabilize, and normalize the dataset used in our ML models * Authored the PCA - Supervised_Learning_Model_2_Neural_Network.ipynb file, which implemented PCA on the drug and target features and compared losses. * Wrote the Introduction & Background, Data Preprocessing, and Principal Component Analysis (PCA) section found under the Methods part of the Final Report * Set up Teams meetings for the team