Semester Project - Drug Discovery for Lung Cancer using Data Mining Techniques

Athulya Anand, Diksha Adke, Atharva Karnik, Sricharraan Ramaswamy[1]

Abstract

Lung cancer remains a leading cause of cancer-related deaths worldwide, with small cell lung cancer (SCLC) being the most prevalent type. While conventional treatment modalities such as chemotherapy, radiation therapy, and surgery have demonstrated efficacy, their associated adverse effects have prompted the exploration of alternative lung cancer therapies, including novel drugs. Notably, the aberrant activation of the Epidermal Growth Factor Receptor (EGFR) protein is a crucial driver of lung cancer. In this project, we aim to evaluate the performance of various regression algorithms, namely SVM Regressor, KNN Regressor, Random Forest Regressor, XGboost Regressor, and PCA Regressor, in predicting the efficacy of a drug in inhibiting the proliferation of cancer cells harboring EGFR protein defects.

Keywords

Lung cancer, Small cell lung cancer, EGFR protein, Novel drugs, Efficacy prediction, Cancer cell proliferation, pIC50, Bioactivity class, Lipinski Descriptors, Molecular Fingerprint, ML algorithms

[1] *Luddy School of Informatics, Computing, and Engineering, Indiana University, Bloomington, IN, USA*

Contents

1. Problem and Data Description

1.1 Problem Statement

Lung cancer remains a leading cause of cancer-related deaths worldwide, with small cell lung cancer (SCLC) being the most prevalent type. Despite advances in the treatment of lung cancer, high mortality rates persist due to the limited efficacy and frequent negative side effects associated with current medications. Therefore, discovering novel therapeutics that are more effective, have fewer adverse effects, and can overcome drug resistance is crucial for improving treatment options for lung cancer patients.

The abnormal activation of the Epidermal Growth Factor Receptor (EGFR) protein is a significant factor in the development and progression of lung cancer. The Epidermal Growth Factor Receptor (EGFR) is a protein responsible for facilitating cell growth and division. In cases of EGFR-positive lung cancer, a mutation or defect in the gene leads to continuous EGFR growth, causing uncontrolled cellular proliferation and ultimately, cancer. While chemotherapy remains one of the most effective solutions to treating cancer, its associated side effects such as fatigue, hair loss, and appetite changes have led to the exploration of alternative therapies. Specifically, various drugs are being tested for their efficacy in inhibiting EGFR protein multiplication. To measure a drug's efficiency in inhibiting EGFR protein growth, we utilized the Inhibitory Concentration (IC 50) value, a quantitative measure of the amount of drug required to inhibit a given biological process by 50 percent.

In this project, we utilized the ChEMBL dataset, a chemical database of bioactive molecules, to obtain the canonical notation of molecular formulas for each drug. We then utilized the PubChem database to convert the canonical notation to molecular fingerprint, a unique binary representation of each molecule. We will implement machine learning models, namely SVM Regressor, KNN Regressor, Random Forest Regressor, XGboost Regressor, and PCA Regressor, to predict

the IC50 value for a given chemical. Our study aimed to compare the performance of these three models in predicting drug efficacy.

Thus, through this project we focus on solving the urgent need to enhance patient outcomes and lessen the burden of this illness serves as the driving force for a medication development endeavor for lung cancer. It provides the opportunity for substantial advancements in our understanding of lung cancer biology, improved treatment choices, higher survival rates, and better patient quality of life.

1.2 Data Description

The ***ChEMBL dataset and Pubchem dataset*** has been utilized to obtain the necessary data for this study. It contains over 2 million curated bioactivity data entries, sourced from more than 76,000 documents and 1.2 million assays. The data covers 13,000 targets, 1,800 cells, and 33,000 indications. The version used for this study was ChEMBL version 26, as of March 25, 2020.

The ChEMBL dataset is a curated chemical database that contains bioactive molecules with drug-like properties. It provides canonical notations of molecular formulas for each drug and integrates chemical, bioactivity, and genetic data to support the development of novel pharmaceuticals.

In addition, the study utilizes the PubChem database, which includes data on small molecules as well as larger molecules such as nucleotides, carbohydrates, lipids, peptides, and chemically modified macromolecules.

2. Data Preprocessing & Exploratory Data Analysis

2.1 Data Preprocessing: Handling Missing Values

Data collection for this study began with a targeted search for 'EGFR' molecules within the ChEMBL database. The focus was on 'Single Protein' target types, and the molecular id corresponding to the human-specific type was extracted. The resulting dataset was then mined for samples with the standard type of 'IC50' to obtain a comprehensive understanding of the bioactivity data.

To ensure the quality of the data, data pre-processing was conducted to remove missing values and duplicates, and compounds were classified and labeled according to their appropriate bioactivity thresholds. Following which, each compound was classified based on the bioactivity class namely ***active, intermediate and inactive***. The resulting data with the bioactivity curation is used for the further processes.

2.2 Exploratory Data Analysis

Following the data preprocessing step in the earlier step, the next step involves conducting 'Exploratory Data Analysis' to gain insights into the data. The dataset consists of the molecular id, its corresponding canonical formula in the form of 'SMILE', and the 'IC 50' value indicating the bioactivity data. ***Lipinski's rule*** of five is applied to the canonical formula to determine if the chemical compound is an active drug in humans. The rule states the following:

a The partition coefficient or 'log p' value should not exceed 5

b There should be no more than 5 hydrogen bond donors

c There should be no more than 10 hydrogen bond acceptors

c The molecular mass should not exceed 500 daltons

To conform to the above-mentioned criteria, the logarithm of the 'IC 50' value is computed to the negative logarithmic scale which is essentially -log10(IC50) and recorded as 'pIC50'. A function is then created to test the molecule with the given Lipinski descriptors using its canonical formula. The Lipinski descriptors in our case are ['MW', 'LogP', 'NumHDonors', 'NumHAcceptors']. Subsequently, exploratory data analysis is carried out on the dataset, utilizing 'pIC50' as the selection criteria and the Lipinksi descriptors. The results indicate that 'pIC50' is the primary measure used to distinguish active, intermediate and inactive compounds.

Fig 1., is a histogram plot showing the distribution of the pIC50 variable in the dataset with 20 bins. From the plot, we can infer the following:

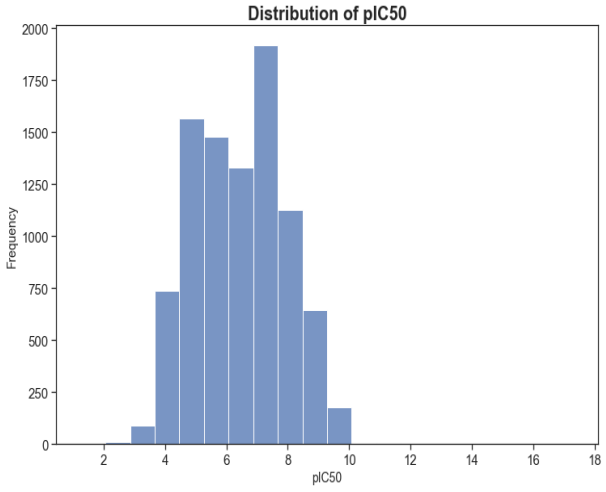

Figure 1. Frequency distribution of pIC50

The distribution of pIC50 appears to be slightly skewed to the right. The majority of the compounds have a pIC50 value between 5 and 8. There are very few compounds with pIC50 values less than 4. The distribution has a long tail to the right, indicating the presence of outliers with higher pIC50 values.

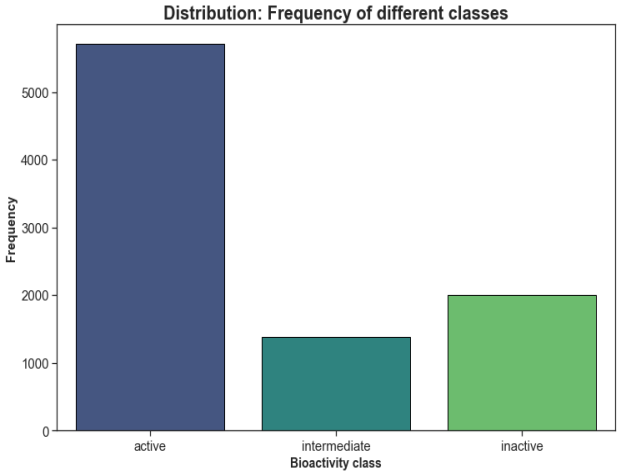

Figure 2. Frequency distribution of Bioactivity classes

From Fig 2., countplot, The most common bioactivity class is ”active,” followed by ”inactive” and ”intermediate.” The distribution of bioactivity classes is not evenly spread and so, we move on to perform EDA on the Lipinksi descriptors to find credible results.

2.2.1 Exploratory Data Analysis via Lipinski Descriptors

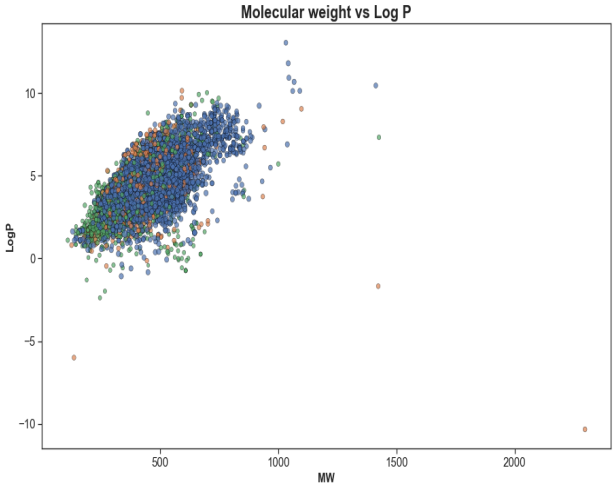

Figure 3. Scatter Plot: Molecular Weight vs Log P

Fig 3., suggests that the majority of compounds in the dataset have a MW below 1000 and a LogP value between -2 and 6. The size of the data points represents the pIC50 value, with larger points indicating compounds with higher potency. The color of the data points represents the bioactivity class, with inactive compounds represented by green, active compounds by blue, and intermediate compounds by orange. With the huge cluster of datapoints there seems to be no clear correlation between MW and LogP with respect to bioactivity class or potency, suggesting that other factors may be more important in determining bioactivity.

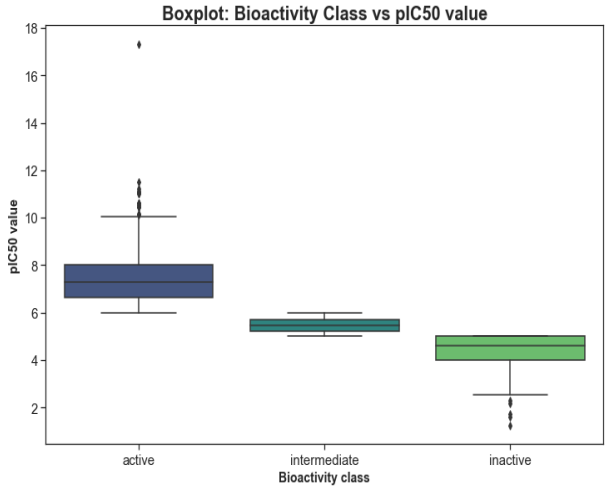

Figure 4. Box Plot: Bioactivity Class vs pIC50

Let us now have a comparitive study between the bioactivity classes and each Lipinski descriptor using EDA.

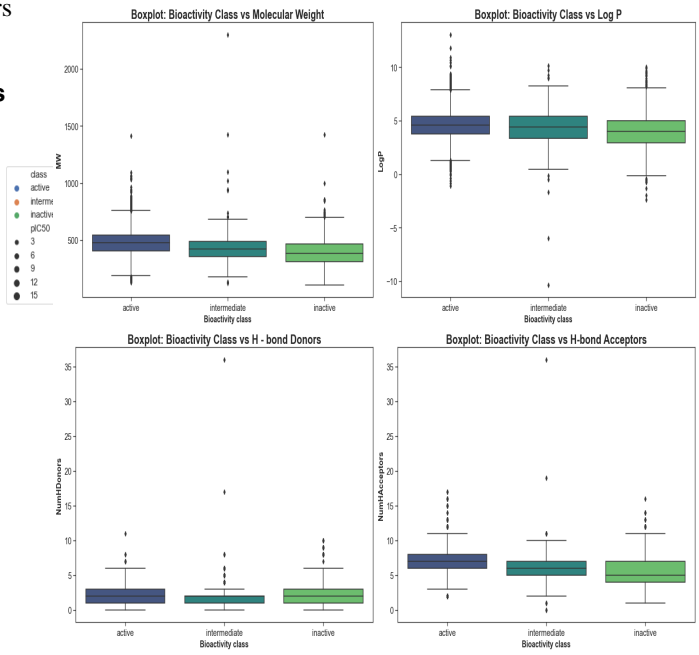

Figure 5. Bioactivity Class with Lipinski Descriptors

From the above subplot in Fig 5., we can analyze the bioactivity class for each lipinski descriptor and see that the median value for each is very variant. This can be slightly tedious and so we plot a pairplot between each and every descriptor along with the pIC50 values to see if any constructive analysis can be made.

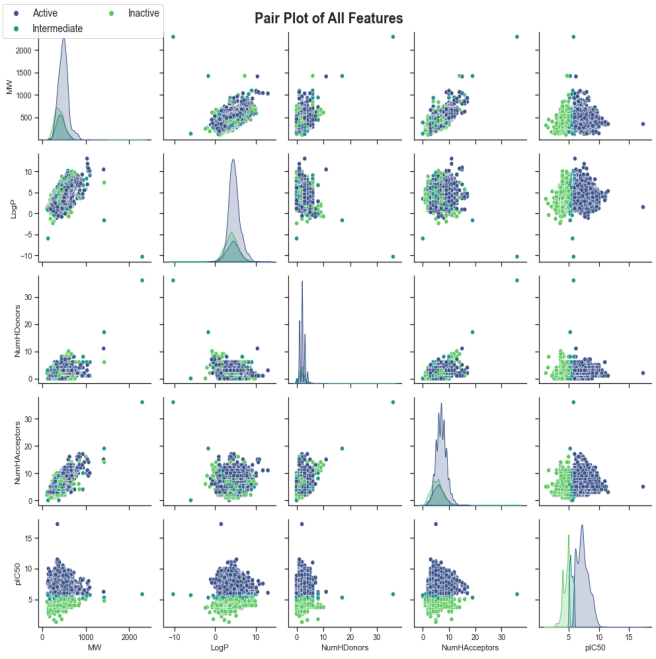

Figure 6. Pair plot of all factors

Fig 9., suggests that the 2 bioactivity classes are spanning similar chemical spaces as evident by the scatter plot of MW vs LogP in Fig 3.

Lastly, we perform Pearsons Correlation between pIC50 and all the Lipinksi descriptor to check the mos highly correlated factors thereby concluding the best features to be selected in the future steps of the project.

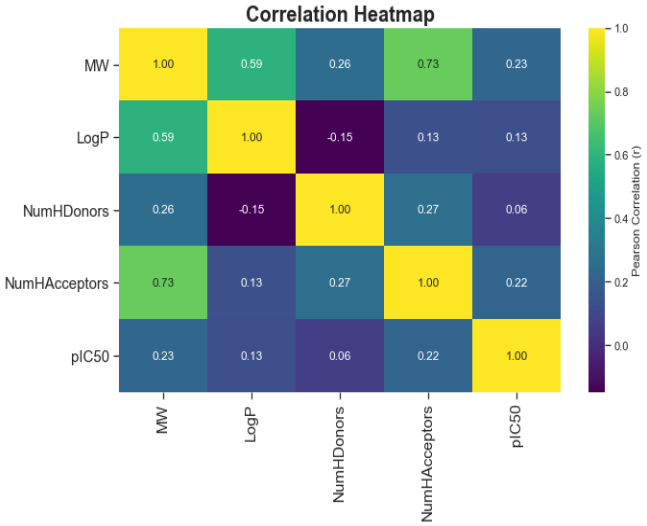

Figure 7. Correlation heatmap of all features

Fig 7., shows that most of the features in the dataset have a low degree of correlation, except for NumHAcceptors and molecular weight, which exhibit a strong positive correlation. Thus, we have decided to keep all the features in the analysis.

3. Algorithm and Methodology

3.1 Computation of Molecular Fingerprint

In the next phase of our project, we aimed to convert the canonical formula of our molecules into their corresponding molecular fingerprints using a software tool called 'PaDEL-Descriptor'. This tool was specifically designed to calculate molecular fingerprints and descriptors. To ensure the accuracy of the fingerprints, the software removed any irrelevant organic compounds and salts from the molecules and standardized the data.

The molecular fingerprints were represented by a series of 0's and 1's in each column, with each column corresponding to a 'PubchemFPx' value. The 'PubchemFP' was used to differentiate between different molecules, and x ranged from 0 to 880. To obtain the necessary files for the software, we downloaded 'padel.zip' and 'padel.sh' from the original 'Chembl' dataset for use. The final dataset for analysis included the **molecular identification, the molecular fingerprint** of 881 columns, and the corresponding **'pIC50'** values.

3.2 Variance Threshold Calculation

In our project, the 'pIC50' value is a continuous variable, which prompted us to use a regressor model for our analysis. To ensure the accuracy of our model, we split the available data into two sets - 80% for training and 20% for testing, thereby setting the threshold value as 0.80.

3.3 Modeling and Algorithm Implementation

Upon Data Preprocessing cleaning and computation of Molecular Fingerprint, we realised that out data is continuous in place of discrete due to which applying classification or clustering models was not a feasible task. Thus, we tested the data using regressor models and found it to be the most optimum method to predict the efficacy of a drug. As a result, we experimented with eight different regressor models, and after careful evaluation, we selected the top five performing models namely SVM Regressor, KNN Regressor, Random Forest Regressor, XGboost Regressor, and PCA Regressor. To compare and validate the accuracy of these models, we employed two widely used model validity techniques - Root Mean Square Error (RMSE) and Mean Absolute Error (MAE).

3.3.1 Support Vector Regressor

Support Vector Regressor (SVR) is a type of regression algorithm that uses support vector machines (SVMs) to predict continuous values. It works by finding the hyperplane that best fits the data while minimizing the margin of error. SVR is particularly useful when dealing with non-linear data as it can effectively handle high-dimensional feature spaces.

Figure 8. Support Vector Regressor

We are using sklearn module to train our SVM Regressor Model and set parameters as **C=1.0, epsilon=0.1, gamma='scale', coef0=0.0, shrinking=True, tol=1e-3, cache-size=200 and max-iter = -1**.

3.3.2 Principal Component Regressor

Principal Component Regressor is a regression method that uses Principal Component Analysis (PCA) to reduce the dimensionality of the dataset while retaining the maximum amount of variation present in the original data. The method creates new independent variables, called Principal Components (PCs), that explain the variation in the data. The PCs are then used as inputs for the regression model to predict the target variable.

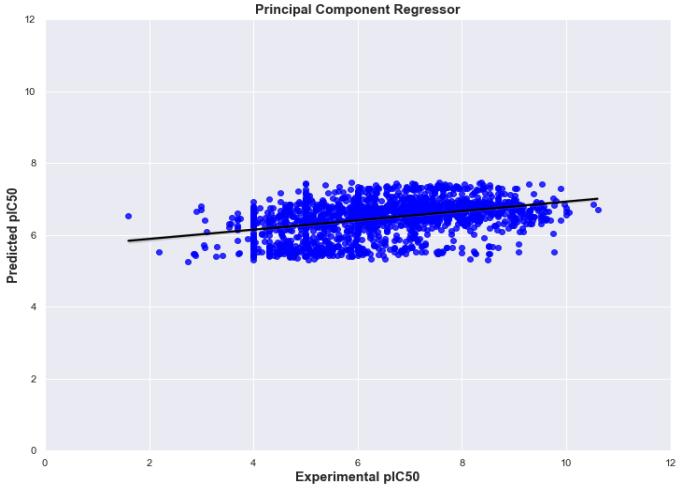

Figure 9. Principal Component Regressor

To train our PCA Regressor Model and Linear Regression Model, we are utilizing the sklearn module and selecting parameters, including **n-components = 10**. Our approach involves fitting PCA on training data, followed by transforming both training and testing data using PCA. We then fit the Linear Regression Model on the transformed data.

3.3.3 Random Forest Regressor

Random Forest Regressor is a type of ensemble machine learning algorithm that is used for regression problems. It combines multiple decision trees to make more accurate predictions. The algorithm randomly selects subsets of the data and features, thereby reducing the risk of overfitting and increasing the model's robustness.

Figure 10. Random Forest Regressor

We are using sklearn module to train our Random Forest Regressor Model. We can select number of parameters for our model. We are using parameters such as **n-estimators = 200** which represents the number of trees, **max-depth = 100** which sets the maximum possible depth of each tree.

3.3.4 K-Nearest Neighbor Regressor

KNN regressor is a non-parametric machine learning algorithm used for regression tasks. It works by finding the K-nearest neighbors to a data point and predicts the output based on the average value of the target variable of these neighbors. The value of K can be tuned based on the data and problem at hand.

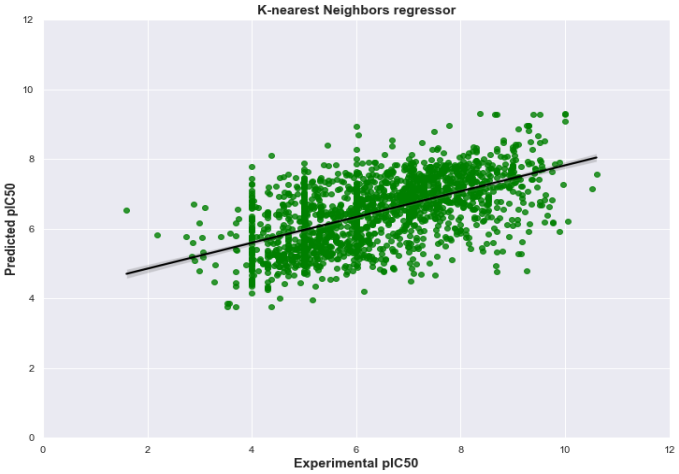

Figure 11. KNN Regressor

We are using sklearn module to train our KNN Regressor Model with **n-neighbors = 10.**

3.3.5 XGBoost Regressor

XGBoost regressor is an advanced implementation of gradient boosting algorithm for regression problems. It is a popular machine learning algorithm that can handle a large amount of data and has high accuracy. XGBoost uses a combination of decision trees and boosting to improve the performance of the model.

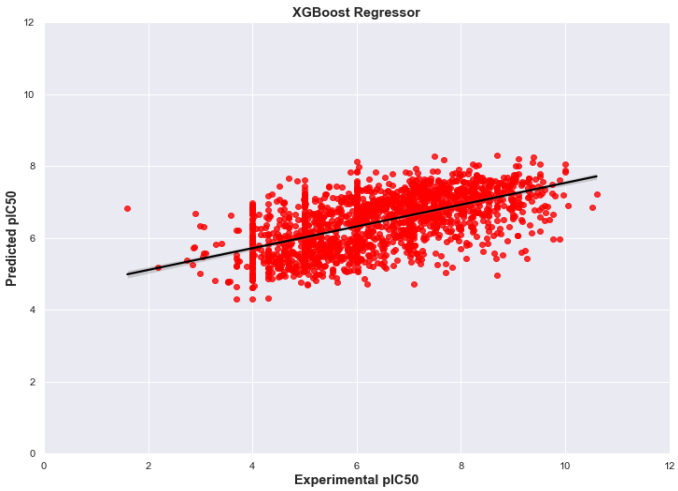

Figure 12. XGboost Regressor

We are using sklearn module to train our XGBoost Regressor Model. We are using **n-estimators = 100, max-depth = 5 and random-state = 123**. We are also using **learning-rate = 0.1**. Learning rate is a weight applied to each regressor at each boosting iteration.

4. Experiments and Results

The figures in Section 3 display the comparison between experimental and predicted 'pIC50' values for each of the top five models. The performance of a regressor line is considered optimal when the slope is 1 and the intercept is 0. Observing the plots, it can be noted that the Random Forest model has the closest regressor line to the origin, indicating the best performance. The SVM model is ranked second in terms of performance, whereas the Principal Component Regressor has the poorest performance among the five models.

Regressor Model Applied	RMSE Value	MAE Value
Support Vector Regressor	1.19	0.9
Principal Component Regressor	1.384	1.133
Random Forest Regressor	1.184	0.877
KNN Regressor	1.225	0.933
XGBoost Regressor	1.21	0.954

Figure 13. Comparison of Results

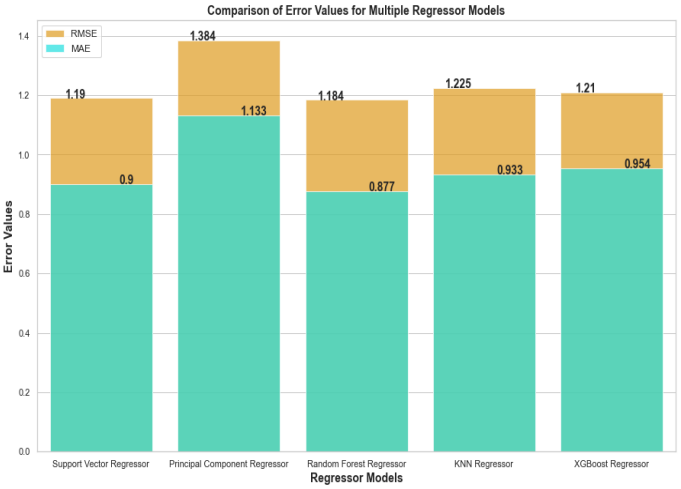

Figure 14. Graphical Comparison of Results

As discussed previously, the performance plot revealed that the Random Forest regressor outperformed other models, achieving the lowest RMSE and MAE values. After comparing the RMSE and MAE scores of each model from Fig 13, it was clear that the Random Forest regressor had the lowest RMSE score, depicted in yellow, and the lowest MAE score, depicted in blue. As a result, we conducted additional experiments by increasing the 'n-estimators' parameter from 200 to 250 and 500, as well as adjusting the 'max-depth' value, to test if the RMSE value could be improved. However, the changes did not result in any significant improvement to the RMSE value. Therefore, we concluded that the Random Forest regressor was the best performing model, followed by the SVM regressor.

5. Deployment and Maintenance

This project consists of static information and lacks user interaction. We make use of Google Cloud to deploy a static website in order to maintain it as a review page for users where they can understand the methods used in predicting the efficacy of a drug in inhibiting the proliferation of cancer cells harboring EGFR protein defects.

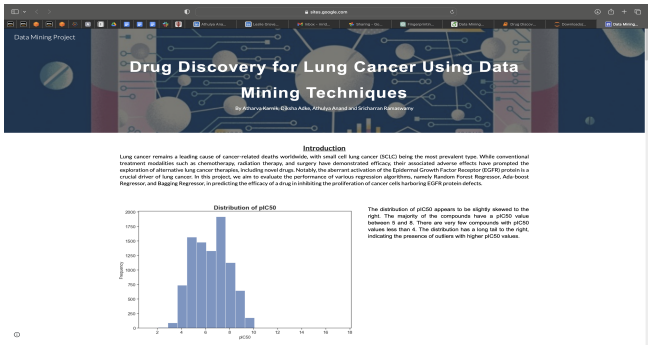

Figure 15. Snippet of the website deployed on Google Cloud

Project Deployment and Maintanence

6. Summary and Conclusions

In conclusion, Random Forest regressor is the best performing model based on comparing **RMSE and MAE values of 1.184 and 0.877**, respectively. These are considered acceptable values as they lie between the range of **0 - 10** as taken for our target variable. SVM regressor is the second-best model with RMSE and MAE values of 1.19 and 0.9 respectively. We can observe that the Regression lines for both models are closer to the origin than other models and that of PCA regressor which has the farthest regression line from the origin with the highest RMSE value. Using the Random Forest Regressor model, pIC50 values were predicted very close to the experimental values.

*Therefore, we can conclude that the **Random Forest Regressor model** is effective for predicting pIC50 values and in predicting the drug efficacy.*

Acknowledgments

We extend our appreciation to the authors of the datasets used in this project for making their data publicly available. Additionally, we also extend our gratitude to Dr. Hasan Kurban, AI's and TA's for their constant support and guidance through the execution of this project.

References

1. https://www.ebi.ac.uk/chembl/
2. https://pubchem.ncbi.nlm.nih.gov/docs/about
3. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3245175/
4. https://en.wikipedia.org/wiki/Small-cell$_{c}arcinomaSmall-cell_{l}ung_{c}ancer$