PREDICTION OF PIC50 VALUES FOR CERVICAL CANCER USING MACHINE LEARNING

ABSTRACT

Cervical cancer, primarily caused by Human Papillomavirus (HPV) infection, is a significant health concern affecting women globally. This study has conducted an extensive analysis of bioactive compounds associated with the target protein Cyclin-Dependent Kinase 1 (CDK1), pivotal in regulating cell division and a potential therapeutic target for cervical cancer treatment. Initial investigation involved identifying IC50 values, indicative of compound concentrations required to inhibit 50% of CDK1 activity. These values were transformed into PIC50 values, a logarithmic scale widely used in pharmacology for simplified interpretation. Leveraging experimental PIC50 values as a benchmark, this study aimed to construct a predictive machine learning model capable of forecasting PIC50 values based on CDK1 bioactivity compounds. Various machine learning algorithms were employed for model training and testing, with iterative refinement to ensure robustness and accuracy. Validation of the model's performance involved comparing predicted PIC50 values against experimental counterparts, visualized through scatter plots to highlight discrepancies. Transitioning the model into a web-based application marks its practical utility, facilitating accessibility for researchers, pharmaceutical companies, and healthcare professionals. Through a user-friendly interface, the application provides predicted PIC50 values based on input data related to CDK1 bioactivity compounds, offering valuable insights for drug discovery and development efforts targeting cervical cancer and other diseases. Features such as data validation, visualization tools, and integration with external resources enhance usability, empowering users with comprehensive analysis capabilities. This research represents a significant contribution to advancing therapeutic strategies for cervical cancer and underscores the potential of machine learning in accelerating drug discovery processes.

1. INTRODUCTION

One of the biggest threats to world health is still cancer. Cervical cancer (CC) is one of the most deadly disease kinds that impact women globally. CC was the fourth most frequent cancer in women in 2020, accounting for 342,000 deaths worldwide and 604,000 new cases. Nonetheless, CC is among the malignancies that respond best to early detection and appropriate management [1]. There are societies existing, where women are health illiterate and are not given enough consideration especially in rural areas. According to the WHO, 303,000 women died from pregnancy-related reasons in 2016, 2.7 million babies died within the first 28 days of their lives, while 2.6 million babies were stillborn. Many of these fatalities might be avoided with proper fetal and neonatal care, yet only 64 percent of women worldwide receive antenatal care four or more times during their pregnancy [2].The fetal phase lasts from eight weeks from fertilization until the moment of childbirth during pregnancy. Periodically checking on the fetus to make sure it is healthy is really important. To protect the health and wellbeing of the unborn child, pregnant mothers must receive frequent antenatal care during each trimester of their pregnancy. Furthermore, these areas can be enhanced by leveraging crucial data in healthcare. A common technique for diagnosing cervical cancer is cytological examination, also known as the Pap smear test. In this procedure, doctors remove cells from a patient's cervix and look at them under a microscope on glass slides to determine whether the patient has cancer. The screening of Pap smears is important in helping women avoid cervical cancer. Given that each slide contains thousands of cells, manual screening is time-consuming and subjective for cytologists. Therefore, for cervical cancer, the automatic computer-aided diagnosis (CAD) systems must be used. Cervical cytology inspection can be performed more quickly and precisely with the assistance of CAD technologies [3]. The primary mechanism through which cervical cancer develops is through the interference of viral oncoproteins E6 and E7 with the tumor suppressor proteins p53 and pRB (retinoblastoma). The severity of the illness is linked to abnormalities in cellular adhesion, cellular control, host cell immunomodulation, and genotoxicity . Chemotherapeutic drugs, which are not specific to their targets, or more invasive and costly surgical and ablative methods can be used to treat infections. Additionally, millions of patients, especially in underdeveloped nations, have limited access to them. As such, the availability of very powerful natural therapies targeted against the virus is one of the primary possibilities for treating problems associated with HPV [4]. Malignant tumors contain cancerous cells that proliferate locally or spread to nearby body areas when they grow out of control. In addition, they may affect the brain, bone, liver, lungs, and other areas before traveling via the lymphatic or circulatory systems to other areas. A wealth of evidence suggests that some viral infections, including hepatitis B and C, Epstein-Barr, HIV, Human Papillomavirus (HPV), Human Herpesvirus 8 (HHV-8), and others, increase the risk of acquiring specific cancers. Particular forms of HPV infection and smoking are two of the biggest risk factors for cervical cancer. Seventy-five percent of all CC cases globally are caused by HPV types sixteen and seventeen,

with the remaining forty percent coming from HPV types 31 and 45. There are between 150 and 200 different types of HPV that have been identified. These are categorized as follows: three are probable high risk (26- 53 and 66), twelve are 2low risk (6-11- 40- 42- 43- 44- 54- 61- 70- 72- 81 and CP6108), and fifteen are high risk (1- 16- 18- 31- 33- 35- 39- 45- 51- 52- 56- 58-59-68- 73 and 82). The majority of cases of cervical cancer can be avoided with effective primary (HPV vaccine) and secondary preventive strategies (screening for and treating precancerous lesions). As long as cervical cancer is identified early and treated appropriately, it is among the most curable types of cancer. Palliative care combined with the right treatment can help control cancers that have been detected in their later stages. Within a generation, cervical cancer as a public health issue can be eradicated with a comprehensive strategy to prevention, screening, and treatment. Effective therapy for cervical cancer, a major worldwide health burden, requires a clearer knowledge of its molecular foundations. One interesting candidate is CDK1, a key regulator of the cell cycle. A complex between CDK1 and cyclin B1 is formed at the G2/M phase transition, which activates cyclin B1's kinase activity. This versatile protein coordinates DNA replication and cell division by phosphorylating substrates [8]. As a crucial factor influencing the course of mitosis, CDK1 has the ability to start mitosis, and its primary function is to regulate G2/M. Numerous lectures have clearly explained the direct impact of CDK1 on cell cycle regulation in addition to its possible mechanisms, which have been thoroughly researched. Cyclin-dependent kinases of protein (CDKs) are significant proteins required for the expression and control of several elements required for the control of a cell's cycle. The management of cell cycle is greatly aided by CDKs in normal cells, and cancer cells are able to evade this control due to abnormal CDK expression. In humans, there are 20 CDKs (1–20) [10] . The frequency and fatalities of cancer remain concerning despite advancements in detection and treatments. For this reason, novel therapeutic and diagnostic approaches must be investigated immediately in order to effectively manage this fatal illness. One feature of cancer that sets it apart is interruption of a cell's cycle. Cyclin-dependent kinases (CDKs), cyclins, and checkpoints all work together to control the cell cycle, that is a highly conserved and tightly regulated cell activity. A tumor grows because a result of uncontrolled proliferation of cells brought on by changes in the regulatory mechanisms that cause the cell cycle control system to fail. Furthermore, since the growth of malignancies depends on the proliferation of cells, targeting cellular protein regulators is a potential treatment approach. The progression of a cell's cycle depends on the activity of cyclin-dependent kinase 1 (CDK1), a member of the kinase family. Inhibiting CDK1 has been shown in numerous trials to be an extremely successful anticancer method for cancer treatment. Strong CDK1 inhibitors and their impact on tumor cell proliferation have been examined in a number of research [3]. Furthermore, CDKs are proteins that affect the course of the cell cycle, making them interesting targets in cancer research. The way that CDKs interact with phosphatases, specific inhibitors, and cyclin-dependent kinases controls how active they are. Different CDK complexes function at different phases. An essential modulator of the initiation and course of a cell's cycle during mitosis is CDK1. Previous research has confirmed CDK1 as a therapeutic candidate by showing that aberrant CDK1 expression or loss of function

is linked to G2 phase arrest and different kinds of tumors [7]. The two most important critical genes of both the normal-CC and non-malignant-CC networks were found to be cyclin-dependent kinase 1 (CDK1) and CDK2. CDK1 dysregulation is clearly seen in cervical cancer, with overexpression seen in tumor tissues and cells. As a result, CDK1 has gained interest as a possible treatment option. With the goal of inducing apoptosis, inhibiting tumor development, and cell cycle arrest, researchers investigate CDK1 inhibitors [8]. CDK1's presence in CC indicates that it is active, and it was examined in silico using molecular docking with the secondary metabolites found in Indian cooking spices. In order to fully understand the genetic networks involved in the development of cervical cancer, CDK1 must be well understood. Novel treatments aimed at CDK1 or associated pathways may help improve the prognosis of cervical cancer in an advanced stage [9]. Cervical cancer research can benefit from the application of Quantitative Structure-Activity Relationship (QSAR) modeling, which uses a compound's chemical structure to predict its biological activity. By correlating chemical features with biological effects, such as the inhibition of mutant CDK1 protein aggregation or the modulation of cellular pathways involved in disease progression, QSAR models can help identify and optimize potent. The phrase "quantitative structure activity relationships" (QSARs) refers to a computerized statistical technique that helps explain the variation in observed structural alterations caused by the interchange. This notion holds that a group of comparable substances' biological activity is the outcome of many physiochemical investigations. These analyses show which physiochemical qualities are beneficial to the activity in question and how to best maximize the latter by choosing substituents that strengthen these properties. The primary goal of Quantitative Structure Activity Relationship (QSAR) and Quantitative Structure Property Relationship (QSPR) research is to establish a mathematical correlation between the activity or property under investigation and one or more descriptive parameters or descriptors associated with the molecule's structure. The goal of machine learning (ML), a branch of artificial intelligence, is to create algorithms that let computers learn from data. ML models are not explicitly designed; rather, they pick up patterns and relationships through examples. There are several types of ML, including supervised learning (using labeled examples), unsupervised learning (identifying patterns in unstructured data), and deep learning (using neural networks). ML applications ranging from image recognition to healthcare. Cervical cancer is a significant global health issue, and ML techniques are increasingly playing a pivotal role in improving its diagnosis, prognosis, and treatment. Here’s how ML is utilized: Early Detection and Screening: Cervical cancer screening is essential for identifying pre-cancerous lesions early. ML algorithms analyze screening test results, such as Pap smears or HPV tests. By processing large datasets, ML models can detect subtle patterns that might be missed by human observers. These models enhance the accuracy of identifying abnormal cells. Prioritizing patients for further evaluation based on ML predictions reduces unnecessary invasive procedures [11]. Survival Prediction: ML models predict survival outcomes for cervical cancer patients. Researchers explore various algorithms, including Random Forest, Logistic Regression, and Support Vector Machines. These models analyze patient data, considering clinical features, treatment history, and genetic markers.

The goal is to estimate survival probabilities. Evaluation metrics like the area under the curve (AUC) assess model performance for overall survival and disease-free survival. Challenges and Considerations: Interpretability: While ML models provide accurate predictions, understanding their decision-making process remains challenging. Researchers actively work on interpretable ML methods. Explainability: Clinicians need insights into why a model makes specific predictions. Efforts focus on making ML models more transparent. Imbalanced Datasets: Ensuring balanced data representation across patient groups is essential to avoid biased predictions. Standardization: Developing standardized ML algorithms for survival prediction requires ongoing research. Image-Based Diagnosis: Deep learning techniques excel in image-based tasks. Automated systems assist doctors and pathologists by providing quicker, error-free decisions based on medical images [12]. The diagnosis and prognosis of cervical cancer are significantly impacted by the classification of cervical cytology images. Machine learning helps individuals' process vast amounts of complex medical data in healthcare and then analyze it for therapeutic insights. Doctors can then use this information to provide medical care. As a result, patient satisfaction can be improved when machine learning (ML) is employed in healthcare. Cervical cancer arises from untreated human papillomavirus (HPV) infection of the cervix. The human papillomavirus (HPV) is the most prevalent infectious agent associated with cervical cancer because it induces neoplastic growth. Neoplastic progression is the term used to describe the aberrant cell multiplication that arises from a malignant phase and the incorrect proliferation of cervical cancer cells [5]. A major contributing factor to CC is HPV infection. HPV16 (q21–q31 of chromosomal no. 13; HPV18 (q24 of chromosome no. 8) DNA integrates into the host cell genome, disrupting the open reading frame and causing overexpression of the E6 and E7 genes. It has been confirmed that E6 and E7 bind to p53 and retinoblastoma (Rb), two cell cycle regulators, to cause cancer. Massive volumes of data are routinely produced by the healthcare sector, and these data can be utilized to forecast future illness based on medical history and treatment history [6]. Moreover, these domains can be improved through the utilization of critical healthcare data. Large volumes of intricate medical data are processed by machine learning, which enables people to examine the data for therapeutic insights. Medical professionals can then use this information to treat patients. Information on bioactivity is crucial for understanding how chemicals interact with biological targets. For the purpose of cervical cancer treatment optimization, drug development, and customized medicine, knowledge of the interactions between certain molecules and cellular constituents is essential. The bioactivity collection is a veritable gold mine of data that helps ML models anticipate outcomes and direct therapeutic choices. Source: A variety of sources, including as high-throughput screenings, literature mining, and experimental assays, are frequently used to curate the bioactivity dataset [11]. Researchers gather information on substances examined against particular molecular targets (such as proteins, enzymes, and receptors) in relation to cervical cancer. Features: Every data point depicts a chemical and the bioactivity that goes along with it. Features consist of: Chemical Structure: Characteristics or identifiers that point to the chemical makeup of the substance. Bioactivity: Data, either quantitative or qualitative, regarding a compound's action (such as

inhibition or binding affinity) against a particular target. Additional Descriptors: Functional groups, physicochemical characteristics, and other pertinent data. Target proteins in cervical cancer studies could be: Oncogenes: Proteins that contribute to the initiation and spread of cancer. Proteins known as tumor suppressors control cell division and avert the development of tumors. Viral proteins: In particular, proteins connected to HPV that are important in cervical cancer. ML systems provide predictions about the impact of novel chemicals on certain targets by learning from bioactivity data. Regression models are used to calculate inhibition constants or binding affinities. Active and inactive substances are identified by classification models. As ML continues to evolve, its impact on cervical cancer care grows. By harnessing critical healthcare data, ML empowers clinicians, transforms diagnostics, and ultimately improves patient well-being. The journey toward better cervical cancer management is paved with data-driven insights and the promise of a healthier future.

2. REVIEW OF LITERATURE

2.1. Among women worldwide, cervical malignant development is the fourth most common cause of illness death. The progression of cervical cancer is associated with infection with the human papillomavirus (HPV). Cervical cancer is now preventable thanks to early screening, which lowers the disease's worldwide impact. Women in underdeveloped nations tend not to participate in enough screening programs due to the high expense of routine examinations, low awareness, and restricted access to healthcare facilities. In this way, a very high level of risk is expected for each individual patient. Numerous risk factors are associated with the production of malignant cervical tissue. In order to assess the risk factors of malignant cervical development, this research suggests a method called CervDetect, which makes use of machine learning algorithms. Pre-processing the statistics with CervDetect entails the usage of Pearson correlation among the center and output variables. CervDetect selects important features by using the random forest (RF) feature selection technique. Finally, CervDetect employs a hybrid strategy to identify cervical cancer by fusing shallow neural networks with radiofrequency technology. The findings demonstrate that CervDetect beats the most recent research in cervical cancer prediction, with an accuracy of 93.6%, a mean squared error (MSE) error at 0.07111, a false-positive percentage (FPR) of 6.4%, and a false-negative rate (FNR) at 100% [13].

2.2. One of the most common and deadly tumors that plague women is cervical cancer. In spite of this, if detected in a precancerous stage, this cancer is entirely curable. The pap smear technique is a common diagnostic tool for the identification of cervical cancer. The hand-operated screening system has a high false-positive rate because of carelessness. Cervical cytology picture classification and division can be done automatically with deep learning-based computer-aided diagnostic techniques, which can increase the efficacy and efficiency of manual screening. An overview of machine learning and deep learning methods for assessing cervical cancer is provided in this survey [14].

2.3. The Pap smear is one of the most widely used methods for early cervical cancer detection. Cervical cancer is the second most common type of cancer among women worldwide. India and other developing nations must overcome obstacles to deal with an increasing number of cases daily. This article applies a variety of offline and online machine learning methods to the detection of cervical cancer using benchmarked data sets. This paper improves the number of features by utilizing additional tree classifiers and tackles the segmentation problem with hybrid techniques. In terms of the percentage of data used for training, accuracy, precision, recall, and F1 scores are rising and can reach 100% for some algorithms [15].

2.4. A prevalent malignant tumor of the female reproductive system, cervical oncogen is one among the world's top causes of death for women. The survival prediction approach works well for the analysis of time-to-event, which is essential for any clinical investigation. The purpose of this project is to methodically look at the application of machine learning to survival analysis in cervical cancer patients [16].

2.5. Among gynecologic cancers, cervical cancer is still one of the most common worldwide. Since cervical cancer is a disease that can be greatly prevented, early screening is the best way to reduce the incidence of cervical cancer worldwide. But in poorer nations, where access to healthcare facilities is limited and procedures are costly, the vulnerable patient populations cannot afford routine examinations. This research presents a novel ensemble approach to forecasting cervical cancer risk. This approach overcomes the drawbacks of earlier cervical cancer research by implementing a voting strategy. To enhance prediction performance, a data rectification mechanism is suggested. An optional gene-assistance module is also incorporated to improve the prediction's resilience. Several measures are taken in order to assess the suggested approach. The findings suggest that the voting technique can be a useful tool for predicting the risk of having cervical cancer. The suggested approach is more workable and scalable than the alternatives [17].

2.6. The primary type of cancer found in women, cervical cancer (CC), can be effectively treated when detected early, yet it remains a significant cause of death, especially in less developed regions. The aim of this research was to create machine learning models using clinical data to accurately spot early-stage cancer. The CC dataset, sourced from the Kaggle repository, included four attribute categories: biopsy, cytology, Hinselmann, and Schiller. These categories were used to segment the dataset into four groups. Three techniques—logarithm, sine function, and Z-score—were applied to modify the features in the datasets. Different supervised machine learning methods were evaluated for classification performance. The Random Tree (RT) algorithm achieved the highest accuracy of 98.33% for biopsy and 98.65% for cytology datasets. For Hinselmann (99.16%) and Schiller (98.58%) datasets, the Random Forest and instance-based K-nearest neighbor algorithms performed the best. Logarithmic feature transformation showed superior results for biopsy datasets, while the sine function was more effective for cytology datasets. Both sine and logarithmic functions performed well for the Hinselmann dataset, whereas Z-score was most effective for the Schiller dataset. Various feature selection techniques were applied to the modified datasets to identify and rank critical risk variables. This study indicates that utilizing clinical data, along with appropriate system design, tuning, machine learning techniques, and classification, can reliably identify CC in its early stages [18].

2.7. It is essential to investigate the genetic variations associated with cervical cancer development post HPV infection, despite the widely acknowledged connection between the virus and cervical malignancies. To identify potential genetic markers for diagnosing or prognosing cervical cancers, our study proposes an integrative machine learning approach. This approach involves three main steps: analyzing gene expression patterns in individual datasets, conducting meta-analysis across multiple datasets, and performing feature selection and machine learning analysis.Through this comprehensive analysis, incorporating seven supervised and one unsupervised methods. By this 21 significant gene expressions are identified. Subsequently, a functional analysis using Gene Set Enrichment Analysis was conducted on this set, revealing enrichment in a nine-gene expression signature. This signature includes upregulated genes like PEG3, SPON1, BTD, and RPLP2, and downregulated genes such as PRDX3, COPB2, LSM3, SLC5A3, and AS1B, suggesting their potential relevance in cervical cancer development [19].

2.8. Cervical cancer is influenced by a myriad of risk factors. The objective of this research was to devise a prognostic model utilizing individual medical histories and preliminary screenings to anticipate patient outcomes concerning cervical cancer. Introducing a decision tree (DT) classification framework, the study aimed to scrutinize the various factors associated with cervical malignancies. Through extensive exploration, including recursive attribute elimination (RAE) and the minimum LASSO, efforts were made to pinpoint the most pivotal characteristics for predicting cervical cancer. Given the dataset's substantial imbalance and missing data, a composite approach incorporating under- and oversampling techniques, denoted as SMOTETomek, was employed. An exhaustive comparative analysis of the proposed model was undertaken, evaluating classifier precision, sensitivity, and specificity concerning feature selection and addressing class imbalance. The DT, integrating chosen attributes from SMOTETomek and RAE, demonstrated notable performance, achieving 100% sensitivity and 98.72% accuracy. This underscores the efficacy of the DT classifier in navigating classification complexities, particularly in scenarios involving reduced attributes and significant class imbalances [20].

2.9. Cervical cancer, a prevalent gynecological malignancy, is predominantly associated with human papillomavirus infection. Various risk factors contribute to its development. Understanding the significance of cervical cancer test variables is crucial for accurate patient classification. This study aims to deepen insights into cervical cancer risk variables by leveraging R's machine learning capabilities. Multiple feature selection strategies are explored to pinpoint essential characteristics for cervical cancer prediction. Through extensive model training iterations employing diverse feature selection techniques, significant attributes are identified,

leading to the development of an ideal feature selection model. Furthermore, the study endeavors to construct several classifier models utilizing C5.0, random forest, rpart, KNN, and SVM algorithms. Each algorithm undergoes thorough training. Notably, the C5.0 and random forest classifiers demonstrate commendable accuracy in identifying women displaying clinical indications of cervical cancer [21].

2.10. This study harnessed advanced machine learning methodologies, recognized as the most efficacious means for addressing the complex task of predicting recurrent cervical cancer. Historically, clinical diagnosis of recurrent cervical cancer relied on clinicians' subjective interpretations of diverse risk factors. Despite extensive clinical research, the identification of pertinent risk factors for recurrence has remained challenging due to their diverse nature. In this investigation, three distinct machine learning techniques—support vector machine, C5.0, and extreme learning machine—were scrutinized to unveil crucial risk indicators for predicting cervical cancer recurrence propensity. Leveraging data sourced from the tumor registry at Chung Shan Medical University, encompassing pathology and medical records, the performance of these techniques was meticulously evaluated. Findings underscore the superiority of the C5.0 model in identifying factors predisposing to recurrence. Our analysis highlights four pivotal variables significantly correlated with heightened recurrence risk: cell type, RT target summary, pathologic stage, and pathologic T. Of particular note are pathologic T and pathologic stage, identified as noteworthy and independent predictors. Future clinical endeavors should consider stratifying patients based on these prognostic markers to assess the efficacy of adjuvant therapy. Furthermore, bolstering post-treatment surveillance may facilitate earlier detection of relapse and more precise assessment of recurrent status, thereby enhancing overall prognosis [22].

2.11. Cervical cancer stands as one of the most prevalent illnesses affecting women worldwide. This condition triggers the abnormal growth of cervix cells, leading to tumor formation. The phase of cervical cancer determines how far the illness has gone from the cervix to other regions of the body. This crucial information is derived from various diagnostic procedures performed subsequent to a physician's physical examination, including colposcopy, biopsy, and imaging studies. Determining the most suitable treatment plan heavily relies on staging. In the field of oncology, accurate diagnosis and optimal treatment selection are paramount. Treatment modalities for cervical cancer are well-established, encompassing surgical interventions, radiation therapy, chemotherapy, or a combination thereof, depending on the cancer's stage. The objective of this study is to identify an efficient algorithm that aids in accurately determining cancer staging through the utilization of data mining technologies. Through the evaluation of accuracy, sensitivity, and specificity of various data mining classification algorithms, their performance was scrutinized, ultimately revealing J48 as the most effective algorithm among the assessed options [23].

2.12. Cervical cancer poses a significant health challenge, underscoring the importance of early detection. Unfortunately, there lacks a universally accepted measurement technique with standardized calibration, often leading to diagnoses reliant on subjective assessments by healthcare providers. To aid in early-stage cervical cancer diagnosis, doctors can benefit from a novel measuring system integrating an optoelectronic sensor with a machine learning algorithm. This article highlights pioneering research in detecting cervical cancer using an optical sensor and forecasting algorithm. Leveraging the refractive index, a fundamental property of all materials, enables insights into tissue condition by determining its value and monitoring changes over time. The analytical software underwent training and validation using datasets derived from optical measurements. Detailed talks address data pretreatment and machine learning outcomes using four separate methods (eXtreme gradient boost, Random Forest, Naïve Bayes, and Convolutional Neural Networks), and an assessment of their effectiveness in classifying tissue as healthy or diseased. Our innovative approach facilitates rapid sample measurement and automatic result classification, holding promise as a valuable tool to aid medical professionals in cervical cancer diagnosis [24].

2.13. In the context of patients undergoing radical hysterectomy post neoadjuvant chemotherapy (NACT), there exists a notable gap in establishing precise predictive models for progression-free survival (PFS). Despite numerous studies identifying factors associated with favorable treatment outcomes in locally advanced cervical cancer, an accurate predictive tool for PFS remains elusive. This study aimed to explore the potential of machine learning (ML) as a predictive tool for PFS following neoadjuvant treatment. A retrospective observational analysis was conducted on patients with locally advanced cervical cancer (FIGO stages IB2, IB3, IIA1, IIA2, IIB, and IIIC1) monitored at a tertiary center between 2010 and 2018. Clinical and demographic data were collected at the 24-month follow-up or at therapy baseline, along with post-surgery histology and magnetic resonance imaging (MRI) exam results. Through meticulous feature selection, a core set of attributes was identified. Subsequently, three distinct machine learning algorithms—Logistic Regression (LR), Random Forest (RFF), and K-nearest neighbors (KNN)—were trained and validated using 10-fold cross-validation to predict 24-month PFS. Our analysis included 92 patients. The presence or absence of fornix infiltration on pre-treatment MRI, along with parametrium invasion and lymph node involvement on post-surgery histology, constituted the attribute core set for training machine learning algorithms. Notably, RFF demonstrated the highest performance, with an accuracy of 82.4%, precision of 83.4%, recall of 96.2%, and an area under the receiver operating characteristic curve (AUROC) of 0.82. Thus, our study successfully developed an accurate machine learning model for predicting 24-month PFS [25].

2.14. The diagnosis of cervical cancer is not standardized. Reliable early detection is provided by combining machine learning with optoelectronic sensors. Preliminary studies indicate that assessment can be achieved with optical sensors and prediction algorithms. Machine learning classifies tissue health with excellent accuracy (>89%). Doctors can diagnose patients more quickly because to this solution's automated classification and quick measuring [26].

3. AIM & OBJECTIVE

AIM OF THE PROJECT

The aim of this project is to use machine learning techniques to forecast PIC50 values based on the bioactivity components of the target protein CDK1, thereby simplifying the QSAR drug design process.

SCOPE OF THE PROJECT

The primary goal of the project comprises,

- Developing a machine learning model to predict the PIC50 values of CDK1 protein
- Comparing different machine learning algorithms to the model to get the best forecasting algorithm for predicting PIC50 values.
- Analyzing the bioactivity molecules of the target protein CDK1 to determine the experimental PIC50 values, this will serve as the data set of inputs for a machine learning model.

4. METHODOLOGY

4.1 Identification of Chemical Compounds for Target protein CDK1 in ChemBL Database

Cervical cancer has been selected as the focal disease for the prognosis model developed in this study. Subsequently, our investigation aims to pinpoint the specific target protein associated with this ailment through an extensive review of relevant research literature. Our findings have led us to identify CDK1 as the target protein of interest. It is imperative to ascertain the chemical compounds correlated with this target protein, as they play a pivotal role in elucidating CDK1's bioactivity. To accomplish this task, leverage is placed on the comprehensive CHEMBL database, meticulously curated to include bioactive compounds possessing drug-like attributes. This repository integrates chemical, bioactivity, and genetic data to facilitate the translation of genomic insights into efficacious pharmaceutical interventions. Employing a systematic approach, a targeted search is conducted within CHEMBL to extract all pertinent chemical targets associated with CDK1. Our search criteria dictate that identified targets must exhibit a minimum bioactivity threshold of 1000, a critical cutoff point necessary for ensuring adequate representation of both active and inactive compounds essential for our machine learning model's predictive capacity. Among the targets identified, we encounter CHEMBL308, meeting our predefined criteria with a bioactivity score of 3894, surpassing the stipulated minimum threshold.

4.2 Data Splitting

An 80:20 split is maintained between the training and testing sets of the dataset. The training set is used to teach the machine learning model how to understand complex relationships and patterns seen in the dataset. In parallel, the testing set is used to assess the model's effectiveness and reveal how well-suited it is to new data. The training set is divided into two subgroups for the purpose of cross-validation. This split makes it easier to examine how well the model performed during training and adjust its parameters. In this iterative process, the validation subset is used to assess performance and make any necessary adjustments, while the training subset is used to train the model. By ensuring that the model is thoroughly evaluated on untested data and extensively trained on a significant chunk of the data, this tactical method efficiently reduces over fitting and ensures optimal performance on new, unknown data outside of the training domain.

4.3 Algorithms

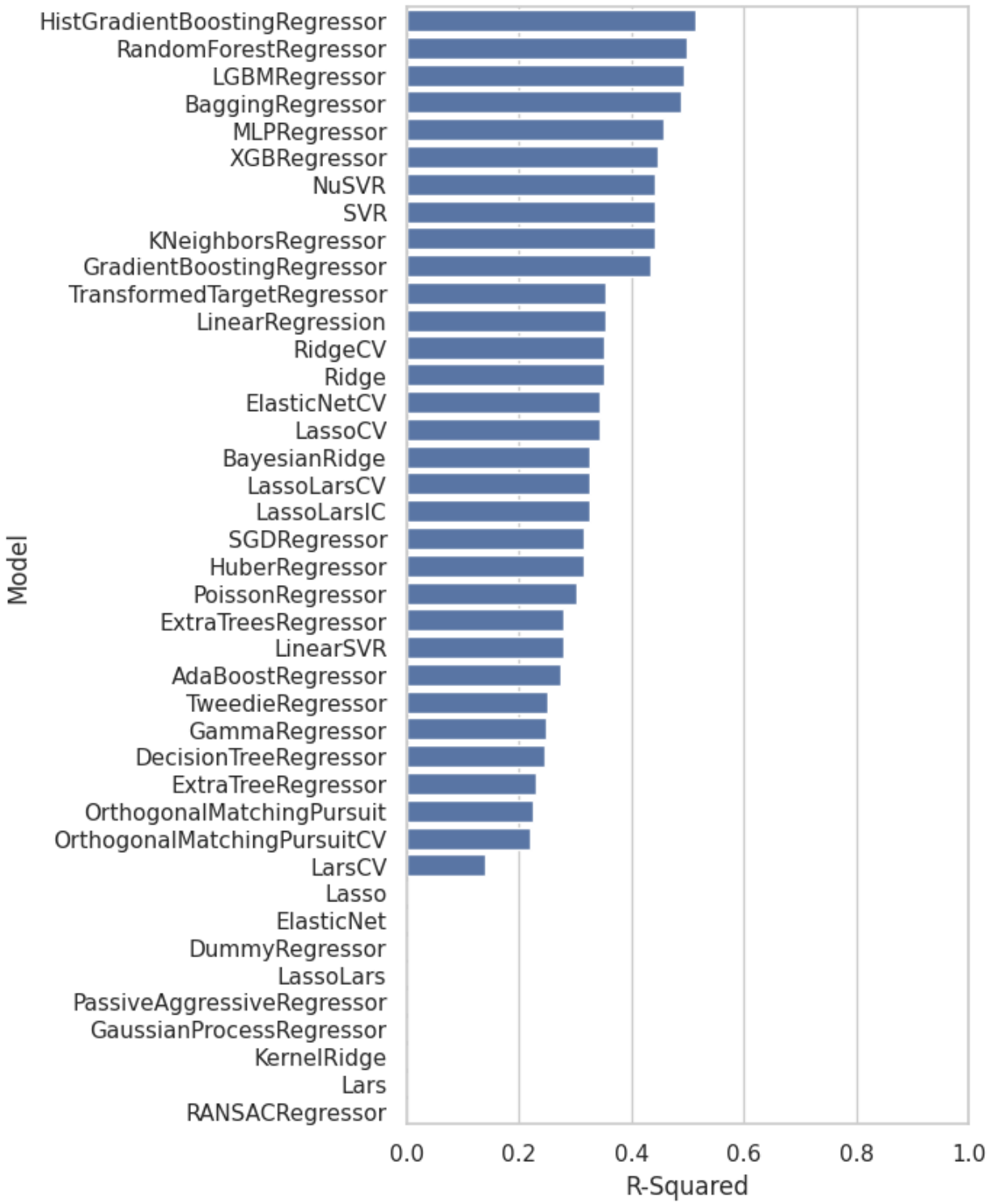
Following the partitioning of the dataset into training and testing subsets, the machine learning prediction model undergoes execution employing approximately 40 distinct algorithms. The principal objective is to assess and contrast the efficacy of diverse algorithms with the intent of identifying the most apt one for the dataset, thereby striving for optimal output.

Data visualization of model performance

```
[ ]  # Bar plot of R-squared values
     import matplotlib.pyplot as plt
     import seaborn as sns

     #train["R-Squared"] = [0 if i < 0 else i for i in train.iloc[:,0] ]

     plt.figure(figsize=(5, 10))
     sns.set_theme(style="whitegrid")
     ax = sns.barplot(y=predictions_train.index, x="R-Squared", data=predictions_train)
     ax.set(xlim=(0, 1))
```



1. HistGradientBoostingRegressor:

The 'HistGradientBoostingRegressor' is a powerful algorithm for regression tasks that use histogram-based gradient boosting to build decision trees. It is particularly well-suited for large

datasets with 10,000 or more samples, outperforming traditional gradient boosting methods. The algorithm constructs histograms of feature values to efficiently find optimal splits during tree growth. It natively handles missing values (NaNs) and assigns samples with missing values to the appropriate child node during training.Various loss functions can be chosen, including ‘squared_error’, ‘absolute_error’, ‘gamma’, ‘poisson’, and ‘quantile’. Hyperparameters include learning_rate’, ‘max_iter’, ‘max_leaf_nodes’, ‘max_depth’, ‘min_samples_leaf ’, and ‘l2_regularization’. The ‘HistGradientBoostingRegressor’ strikes a balance between accuracy and speed, making it useful when training time is a concern. Inspired by LightGBM, it achieves similar performance with fewer trees. It is suitable for regression tasks requiring accurate predictions, large datasets with missing values, and situations where training time is important.

2. RandomForestRegressor

The Random Forest Regression algorithm is a machine learning technique that combines predictions from multiple decision trees to create a more accurate and stable overall prediction. It leverages the collective intelligence of these trees to improve performance. The algorithm works by building multiple decision trees (the "forest") using different subsets of the training data, each trained on a random subset of features and samples (Bootstrap Aggregating or Bagging). The final prediction is an aggregation of the predictions from individual trees. Key features of Random Forest include its robustness to overfitting, ability to handle large datasets with high dimensionality, and its ability to perform regression and classification tasks. It also ensures randomness and diversity by running trees in parallel, ensuring diversity among the trees.The algorithm's performance depends on hyperparameters like the number of trees, maximum depth, and minimum samples per leaf. Experimenting with different settings can help find the best configuration for your specific problem. It can be used for predicting house prices, stock prices, handling noisy data or missing values, and dealing with large datasets efficiently.

3. LGBMRegressor

The LGBMRegressor algorithm, also known as LightGBM, is an open-source machine learning library developed by Microsoft that uses gradient boosting to create a robust predictive model. It is particularly efficient and performs well on large datasets. LightGBM uses a histogram-based approach to construct decision trees, which speeds up the training process. Key parameters include boosting_type, num_leaves, learning_rate, n_estimators, and max_depth. The algorithm can be used for various tasks, such as predicting continuous values, binary or multiclass classification tasks, and lambdarank tasks. It is memory-efficient and scales well, handling large datasets, high-dimensional features, and missing values effectively. The algorithm balances accuracy and speed. Use cases for LightGBM include predicting house prices, stock returns, or

any continuous target variable, handling sparse data or categorical features, and solving regression problems efficiently. To optimize performance, fine-tune hyperparameters like `learning_rate`, `num_leaves`, and `n_estimators`. LightGBM is a fantastic choice for regression problems and can be used in various scenarios, such as predicting house prices, stock returns, handling sparse data or categorical features, and solving regression problems efficiently. It is essential to fine-tune hyperparameters to optimize performance for specific tasks.

4. BaggingRegressor

The 'BaggingRegressor' algorithm is an ensemble meta-estimator used for regression tasks. It combines predictions from multiple base regressors, typically decision trees, to create a more robust and accurate final prediction. The algorithm reduces variance by introducing randomness during model construction. It works by fitting base regressors on random subsets of the original dataset, making individual predictions. The final prediction is an aggregation of these individual predictions. There are three types of bagging: printing, bagging, random subspaces, and random patches. The parameters of the algorithm include the ‘estimators’, 'n_estimators', 'max_samples', 'max_features', and 'bootstrap'. The algorithm can be used for predicting house prices, stock returns, handling noisy data or missing values, and improving model stability and robustness. The 'BaggingRegressor' is a valuable tool in your regression toolbox, as it can handle various tasks such as predicting house prices, stock returns, handling noisy data or missing values, and improving model stability and robustness. Experimenting with hyperparameters can optimize performance for specific problems.

5. MLPRegressor

The Multi-Layer Perceptron Regressor (MLPRegressor) is a neural network-based algorithm used for regression tasks. It is part of the scikit-learn library and consists of an input layer, one or more hidden layers, and an output layer. Each layer contains multiple neurons (nodes), connected by weighted edges. The neural network can be trained and optimized using either the LBFGS optimizer or stochastic gradient descent (SGD), and can control learning rate, regularization, and other hyperparameters. MLPRegressor can handle both regression and classification tasks, can model complex nonlinear relationships, and is effective for large datasets. Hyperparameters include hidden_layer_sizes, activation, solver, L2 regularization strength, learning rate, batch size, and more. Use cases for MLPRegressor include predicting house prices, stock returns, handling noisy data or missing values, and capturing intricate patterns in data. Experimenting with hyperparameters and architecture is crucial for optimal performance. Neural networks, including MLPRegressor, are powerful tools for regression tasks.

6. XGBRegressor

The XGBRegressor, short for Extreme Gradient Boosting Regressor, is a powerful machine learning algorithm that excels in regression tasks. It is an implementation of the gradient boosting algorithm, which combines multiple weak learners to create a strong predictive model. Developed by Tianqi Chen, XGBoost is highly efficient and effective, dominating structured or tabular datasets for both classification and regression tasks. Key features of XGBoost include scalability, model performance, and suitable for tabular data with features like numerical values, categorical variables, and missing values. Hyperparameters of XGBoost include n_estimators, learning_rate, max_depth, and sample. It is suitable for predicting house prices, stock returns, handling noisy data or missing values, and achieving accurate results with structured data. XGBoost is the algorithm of choice for many Kaggle competition winners and consistently delivers top performance across various domains. To optimize your XGBoost model, it is essential to fine-tune hyperparameters and explore feature importance. It is a versatile tool for regression problems, and it is recommended to fine-tune hyperparameters and explore feature importance to optimize your XGBoost model.

7. NuSVR

Nu Support Vector Regression (NuSVR) is a variant of Support Vector Machines (SVMs) designed for regression tasks. It is used for predicting continuous numerical values and shares similarities with SVMs but introduces a parameter called `nu` to control the number of support vectors. The `nu` parameter replaces the epsilon parameter used in epsilon-SVR, controlling both the number of training errors and the fraction of support vectors. The value of `nu` lies in the interval (0, 1], where 0.5 is the default. NuSVR supports various kernel functions, such as linear, polynomial, radial basis function (Gaussian), and sigmoid kernels. Hyperparameters include a penalty parameter for the error term, kernel type, and other parameters like `degree`, `gamma`, and `coef0`. NuSVR performs well on structured data with features like numerical values and categorical variables, is effective for handling noisy data and missing values, and is used when you need accurate regression predictions. It is important to fine-tune hyperparameters and explore the impact of `nu` on your specific problem.

8. SVR

Support Vector Regression (SVR) is a machine learning algorithm used for regression analysis, aiming to find a function that approximates the relationship between input variables and a continuous target variable while minimizing prediction error. Unlike traditional regression, SVR focuses on finding a hyperplane that best fits the data within a specified margin. It constructs a hyperplane in a high-dimensional space, aiming to maximize the margin (distance) between the

hyperplane and the data points. SVR allows some data points to lie within the margin (controlled by the parameter `epsilon`). Kernel functions are used to transform the input data into a higher-dimensional space, with common kernels including linear, polynomial, and the Radical Basis Function (RBF) for complex patterns. Hyperparameters control the trade-off between fitting the training data and allowing errors, while `C` controls the trade-off between fitting the training data and allowing errors. SVR performs well on structured data with features like numerical values and categorical variables, is effective for handling noisy data and missing values, and is used when accurate regression predictions are needed. It is important to fine-tune hyperparameters and explore the impact of the margin (`epsilon`) on specific problems.

9. KNeighborsRegressor

The KNeighborsRegressor algorithm is a supervised machine learning technique used for regression tasks. It is based on the idea that similar data points tend to have similar target values. The algorithm works by identifying the k nearest neighbors of a new data point, based on a distance metric such as Euclidean distance. The predicted target value for the new point is an aggregation of the target values of its k neighbors. Hyperparameters include `n_neighbors`, `weights`, and `algorithm`. Common distance metrics include Euclidean distance, Manhattan distance, and Minkowski distance. KNeighborsRegressor can be used for predicting house prices, stock returns, handling noisy data or missing values, and performing simple and interpretable regression tasks. To use KNeighborsRegressor, choose an appropriate value for `k` and explore different distance metrics based on your specific problem. It is a valuable tool for regression tasks, and it is essential to choose an appropriate value for `k` and explore different distance metrics based on your specific problem.

10. GradientBoostingRegressor

The Gradient Boosting Regressor algorithm is a powerful ensemble method used for regression tasks. It combines multiple weak learners, usually decision trees, to create a strong predictive model. The algorithm works by fitting a regression tree on the negative gradient of the given loss function, optimizing arbitrary differentiable loss functions. Trees are added sequentially, correcting errors made by previous models. Hyperparameters include the number of boosting stages (trees), learning rate, max_depth, and sample fraction. Loss functions can be chosen from'squared_error', 'absolute_error', 'huber', and 'quantile'. The Gradient Boosting Regressor performs well on structured data with features like numerical values and categorical variables, is effective for handling noisy data or missing values, and is used when accurate regression predictions are needed. It is essential to fine-tune hyperparameters and explore feature importance to optimize the Gradient Boosting Regressor model.

11. TransformedTargetRegressor

The `TransformedTargetRegressor` is a meta-estimator in scikit-learn designed for regression tasks. It is useful for applying a non-linear transformation to the target variable (y) in regression problems. This transformation can be given as a transformer (such as the QuantileTransformer) or a function and its inverse (e.g., `np.log` and `np.exp`). During training, the regressor (e.g., LinearRegression) fits on the transformed target (e.g., `func(y)` or `transformer.transform(y)`). During prediction, the inverse function (e.g., `inverse_func`) is applied to the regressor's predictions. Alternatively, the transformer's inverse transform (e.g., `transformer.inverse_transform`) can be used. The hyperparameters of the `TransformedTargetRegressor` include the base regressor (automatically cloned for each fit), the estimator for transforming the target variable, a function to apply to y before fitting (returning a 2-dimensional array), and a function to apply to the regressor's predictions (returning a 2-dimensional array). The `check_inverse` function is used to verify that the transform followed by inverse_transform leads to the original targets. The `TransformedTargetRegressor` is a valuable tool for handling non-linear relationships in regression. It is essential to choose an appropriate transformation and explore different base regressors.

12. LinearRegression

Linear Regression is a fundamental machine learning algorithm used for regression tasks. It aims to model the relationship between one or more independent variables (features) and a continuous dependent variable (target). It works by fitting a linear equation to the data, which has the form `y = mx + b`. The assumptions of Linear Regression include linearity, independence, and homoscedasticity. It minimizes the sum of squared differences between predicted and actual values. Linear Regression has no hyperparameters to tune, making it a straightforward yet powerful tool. It serves as a foundation for more complex algorithms and provides valuable insights into relationships between variables. It can be used for predicting house prices, stock returns, or any continuous target variable. Linear Regression is simple and interpretable, making it a powerful tool for predicting various outcomes. It serves as a foundation for more complex algorithms and provides valuable insights into relationships between variables. It is a straightforward yet powerful tool that serves as a foundation for more complex algorithms.

13. RidgeCV

RidgeCV is a linear regression algorithm that combines ridge regression (L2 regularization) with built-in cross-validation to find a balance between data fit and preventing overfitting. It optimizes the regularization parameter (alpha) using cross-validation and performs efficient

Leave-One-Out Cross-Validation (LOOCV) by default. The `alphas` parameter specifies a range of alpha values to try during cross-validation, with larger alpha values increasing regularization strength and smaller alpha values allowing the model to fit the data more closely. RidgeCV benefits include effectively handling multicollinearity, improving model stability and generalization, and helping prevent overfitting. It can be used in various scenarios, such as predicting house prices, stock returns, or any continuous target variable, and dealing with high-dimensional data or correlated features. It is essential to explore different alpha values and evaluate the model's performance to ensure its effectiveness. RidgeCV is a valuable tool for regression tasks and should be explored to evaluate its performance.

14. Ridge

Ridge Regression, also known as Tikhonov Regularization, is an extension of linear regression that introduces regularization to improve model performance. It addresses the need for regularization in linear regression, which can suffer from overfitting when there are many features or multicollinearity. Ridge regression adds a regularization term to the cost function, which includes an additional term, the sum of squared coefficients multiplied by a hyperparameter `alpha`. The goal is to minimize both the error and the magnitude of coefficients. The hyperparameter `alpha` controls the strength of regularization, with smaller values leading to less regularization and larger values increasing regularization. Ridge Regression has several benefits, including effectively handling multicollinearity, improving model stability and generalization, and preventing overfitting by shrinking coefficients. It can be used in various cases, such as predicting house prices, stock returns, or any continuous target variable, and dealing with high-dimensional data or correlated features. Choosing an appropriate value for `alpha` based on the specific problem is crucial, as it strikes a balance between fitting the data and preventing overfitting.

15. ElasticNetCV

ElasticNetCV is a regression method that combines the strengths of ridge regression (L2 regularization) and lasso regression (L1 regularization). It is designed for regression tasks aiming to predict a continuous target variable, balancing the trade-off between data fit and preventing overfitting. The algorithm optimizes two hyperparameters: ‘alpha’ (the regularization strength) and ‘l1_ratio’ (the mix of L1 and L2 penalties). It performs cross-validation to find the best combination of these hyperparameters. ElasticNetCV combines both L1 (lasso) and L2 (ridge) penalties, with the ‘l1_ratio’ parameter controlling the balance between these penalties. It is useful for predicting house prices, stock returns, handling multicollinearity effectively, and providing a flexible regularization approach. The hyperparameters ‘alpha’ determine the overall

strength of regularization and ‘l1_ratio’ control the mix of L1 and L2 penalties. It is essential to explore different values of ‘alpha’ and ‘l1_ratio’ to find the best combination for your specific problem. ElasticNetCV is a valuable tool for regression tasks and can be found in various sources, including the Contiki Cooja Project, PadaKuu.com, and the Machine Learning Compass.

16. LassoCV

The LassoCV algorithm is a powerful method for regression tasks that combines Lasso regression (L1 regularization) with cross-validation. It automatically selects the optimal regularization parameter (alpha) for Lasso. The algorithm performs cross-validation to find the best alpha value, optimizing the objective function – ‘(1 / (2 * n_samples)) * ||y - Xw||^2_2 + alpha * ||w||_1’. The regularization strength (alpha) controls the trade-off between fitting the data well and preventing overfitting. Larger alpha values increase regularization, shrinking coefficients. Benefits of LassoCV include effective handling of multicollinearity, improving model stability and generalization, and automatically selecting the best alpha using cross-validation. It can be used for predicting house prices, stock returns, or any continuous target variable, especially when dealing with high-dimensional data or correlated features. It is essential to explore different alpha values and evaluate the model's performance. LassoCV is a valuable tool for regression tasks.

17. BayesianRidge

The Bayesian Ridge Regression algorithm is a powerful method for regression tasks that combines Bayesian principles with linear regression. It incorporates probabilistic reasoning into linear regression, modeling the target variable as a probability distribution. The algorithm estimates the posterior distribution of model parameters (coefficients) given the data, assuming a Gaussian prior distribution for the coefficients. It computes the posterior distribution using Bayes' theorem. Bayesian Ridge introduces regularization to prevent overfitting and balances fitting the data well with keeping the model simple. The regularization strength is controlled by hyperparameters, which allow fine-tuning the regularization. The algorithm can be used for predicting house prices, stock returns, or any continuous target variable, providing a probabilistic approach to regression. It is essential to explore different hyperparameter values and evaluate the model's performance. Bayesian Ridge Regression is a valuable tool for regression tasks, and its performance should be evaluated using different hyperparameter values.

18. LassoLarsCV

The LassoLarsCV algorithm is a powerful method for regression tasks that combines the Lasso (L1 regularization) with the Least Angle Regression (LARS) algorithm. It performs cross-validated Lasso regression using the LARS algorithm and automatically selects the optimal regularization parameter (alpha) for Lasso. LARS is an efficient algorithm for fitting linear models with high-dimensional data, and LassoLarsCV optimizes the alpha value using cross-validation to balance data fitting well with preventing overfitting. The regularization strength (alpha) controls the strength of regularization, with larger alpha values increasing regularization and smaller alpha values allowing the model to fit the data more closely. LassoLarsCV benefits from handling multicollinearity effectively, improving model stability and generalization, and automatically selecting the best alpha using cross-validation. It can be used for predicting house prices, stock returns, or any continuous target variable, especially when dealing with high-dimensional data or correlated features.

19. LassoLarsIC

The LassoLarsIC algorithm is a powerful method for regression tasks that combines the Lasso (L1 regularization) with the Least Angle Regression (LARS) algorithm. It uses the Akaike information criterion (AIC) or the Bayes information criterion (BIC) for model selection, automatically selecting the optimal value of the regularization parameter (alpha). This algorithm balances fitting the data well with preventing overfitting and handles multicollinearity effectively. It can be used for predicting house prices, stock returns, or other continuous target variables, especially when dealing with high-dimensional data or correlated features.

20. SGDRegressor

The SGDRegressor algorithm is a linear regression model in scikit-learn that uses stochastic gradient descent to train linear models. It updates the model's parameters incrementally, one data point at a time, making it useful for large datasets. The algorithm fits a linear regression model using stochastic gradient descent, with regularization added to prevent overfitting. It can be used for predicting house prices, stock returns, and handling large datasets efficiently. It's essential to fine-tune hyperparameters and explore different loss functions.

21. HuberRegressor

The HuberRegressor algorithm is a robust regression model that combines the advantages of mean squared error (MSE) and mean absolute error (MAE) loss functions. It is designed to be less sensitive to outliers and balances robustness against outliers with the need to fit the data well. The algorithm optimizes the squared loss for samples with small absolute deviations, and switches to the absolute loss for larger deviations. The parameter 'epsilon' controls the number of samples classified as outliers. It is useful for predicting house prices, stock returns, and dealing with noisy or outlier-containing data.

22. PoissonRegressor

The Poisson Regressor is a Generalized Linear Model (GLM) used for regression analysis, specifically designed for counting data and contingency tables. It assumes that the response variable Y follows a Poisson distribution and uses the 'log' link function to relate expected counts to predictor variables. The model has parameters such as alpha, fit intercept, solver, maximum iterations, tolerance, warm start, verbosity, coefficients, intercept, and number of features. It is particularly useful when dealing with count-based data, such as the number of events, occurrences, or accidents. Poisson regression is commonly used in fields like epidemiology, finance, and social sciences.

23. ExtraTreesRegressor

The ExtraTreesRegressor is an algorithm that builds an ensemble of decision trees (extra-trees) and averages their predictions to improve accuracy and reduce overfitting. It is an extension of the Random Forest algorithm and creates multiple decision trees, introducing randomness during tree construction. The final prediction is an average of predictions from all trees. The ExtraTreesRegressor has parameters such as n_estimators, criteria, max_depth, min_samples_split, min_samples_leaf, max_features, bootstrapping, and oob_score. It is useful for predicting continuous numeric values like house prices and stock prices, and providing feature importance to help identify influential features.

24. LinearSVR

Linear Support Vector Regression (LinearSVR) is an algorithm used for regression tasks, aiming to predict continuous numeric values based on input features. It is an extension of the Support Vector Machine (SVM) for regression and works similarly to SVR with a linear kernel, but

implemented using liblinear. LinearSVR offers more flexibility in choosing penalties and loss functions and scales better for large datasets. Parameters of LinearSVR include epsilon, tolerance, regularization, loss function, intercept, intercept scaling, and dual optimization. It is useful when you need a linear regression model that can handle large datasets efficiently and works well for both dense and sparse input data.

25. AdaBoostRegressor

AdaBoostRegressor is an ensemble learning algorithm used for regression tasks. It is a meta-estimator that combines multiple weak regressors, usually decision trees, to create a strong ensemble model. It iteratively trains weak models, adjusting their weights based on prediction errors, with a focus on difficult cases. The algorithm works by assigning equal weight to each data point, increasing the weight of misclassified instances in each iteration, focusing more on misclassified samples, and resulting in a weighted average of individual regressor predictions. The parameters of AdaBoostRegressor include the number of base estimators (weak regressors), learning rate, loss function, and random_state. It is useful when a robust regression model adapts well to complex relationships in data and performs well even with noisy or sparse datasets.

26. TweedieRegressor

The TweedieRegressor is a Generalized Linear Model (GLM) used for regression tasks. It models the relationship between input features and a continuous target variable based on the Tweedie distribution, which encompasses various distributions depending on the power parameter. The parameters include power, which determines the underlying target distribution, regularization (alpha), link function, solver, max iterations, tolerance, and warm start. The TweedieRegressor is useful for modeling different GLMs based on the specified power parameter. For example, the TweedieRegressor can be used to create a TweedieRegressor, fit it to data, and make predictions.

27. GammaRegressor

The GammaRegressor is a Generalized Linear Model (GLM) used for regression tasks. It models the relationship between input features and a continuous target variable based on the Tweedie distribution, which encompasses various distributions depending on the power parameter. The parameters include power, which determines the underlying target distribution, regularization (alpha), link function, solver, max iterations, tolerance, and warm start. It is useful when you need a robust regression model that adapts well to complex relationships in the data and performs well even with noisy or sparse datasets. For example, the example code demonstrates

creating a GammaRegressor, fitting it to data, and making predictions. The GammaRegressor is useful for modeling different GLMs based on the specified power parameter. For more details, refer to the scikit-learn documentation.

28. Decision Tree Regressor

The Decision Tree Regressor is a powerful algorithm used for regression tasks, a tree-like model where each internal node represents a feature, each branch represents a decision rule, and each leaf node represents the predicted output. The prediction at each leaf node is the average of the target values in that node. The algorithm has parameters such as the criteria, max depth, minimum samples split, minimum samples leaf, and max features. It is useful when you need an interpretable model that captures non-linear relationships and can handle both continuous and categorical features.

29. OrthogonalMatchingPursuit

The Orthogonal Matching Pursuit (OMP) algorithm is an iterative greedy technique used for sparse signal recovery and feature selection. It is an iterative greedy algorithm that seeks to reconstruct signals using a limited set of measurements. It intelligently selects elements from a "dictionary" to match the signal, operating in a stepwise manner. The process continues until a specified sparsity level is reached or the signal is adequately reconstructed. OMP works by sequentially selecting the most correlated element with the current residuals, adding this element to the set of selected features. The residuals are updated, and the process repeats until the desired sparsity or accuracy is achieved. OMP is commonly used in compressed sensing, feature selection, signal processing, and image or audio processing.

30. Orthogonal Matching PursuitCV

The Orthogonal Matching PursuitCV (OMP) algorithm is a powerful technique used for sparse signal recovery and feature selection. It is an iterative greedy algorithm that seeks to reconstruct signals using a limited set of measurements. It intelligently selects elements from a "dictionary" to match the signal, operating in a stepwise manner. The process continues until a specified sparsity level is reached or the signal is adequately reconstructed. OMP works by sequentially selecting the most correlated element with the current residuals, adding this element to the set of selected features. The residuals are updated, and the process repeats until the desired sparsity or accuracy is achieved. OMP is commonly used in compressed sensing, feature selection, signal processing, and image or audio processing.

31. LarsCV

The LarsCV (Least Angle Regression with Cross-Validation) algorithm is a powerful technique used for sparse signal recovery and feature selection. It is an extension of the Least Angle Regression (LARS) algorithm, which combines LARS with cross-validation to find an optimal regularization parameter (alpha) for linear regression. LARS is a forward selection algorithm that adds features to the model stepwise, while LarsCV extends LARS by performing cross-validation to select the best alpha value. It efficiently explores the entire regularization path. LarsCV is useful when identifying relevant features while avoiding overfitting and performing linear regression with automatic feature selection. An example of using LarsCV is creating a LarsCV model, fitting it to data, and making predictions. It is particularly useful when performing linear regression with feature selection. LarsCV is particularly useful when you need an efficient way to perform linear regression with feature selection.

32. Lasso

The Lasso (Least Absolute Shrinkage and Selection Operator) algorithm is a powerful technique used for regression tasks. It enhances regular linear regression by slightly changing its cost function, resulting in less overfit models. It combines feature selection and regularization to improve prediction accuracy and interpretability. Lasso works by adding a penalty term to the traditional linear regression model, encouraging sparse solutions by forcing some coefficients to be exactly zero. It effectively selects relevant features while avoiding overfitting. Parameters of Lasso include the Alpha (α) parameter, the Maximum Iterations parameter, the tolerance parameter, and the Link Function parameter. It is useful when identifying important features for prediction and handling high-dimensional data efficiently.

33. ElasticNet

Elastic Net is a powerful technique used for regression tasks that combines features from two popular regularized linear regression models: Ridge and Lasso. It aims to strike a balance between the L1 (Lasso) and L2 (Ridge) penalties, offering the best of both worlds. Elastic Net adds both L1 and L2 penalties to the linear regression cost function, with a hyperparameter α controlling the trade-off between L1 and L2 penalties. By adjusting α, one can emphasize feature selection (like Lasso) or feature shrinkage (like Ridge). Elastic Net is useful for handling high-dimensional data efficiently, selecting relevant features while avoiding overfitting, and combining Ridge and Lasso benefits.

34. Dummy Regressor

The Dummy Regressor is a regression model that provides predictions based on simple strategies without considering input data. It serves as a baseline for comparing other existing regressors and is useful for establishing a basic reference point when evaluating more complex models. The Dummy Regressor employs several strategies for making predictions, including the mean strategy, which always predicts the mean of the training target values, the median strategy, which predicts the median of the training target values, the quantile strategy, which predicts a specific quantile of the training target values, and the constant strategy, which predicts a custom value provided by the user.

35. LassoLars

LassoLars, also known as Least Angle Regression (LARS) Lasso, is a linear regression technique that combines the benefits of Lasso regularization with the efficiency of the LARS algorithm. It is designed for regression tasks where the goal is to predict a continuous target variable by selecting relevant features and estimating their coefficients. The algorithm works by iteratively adding features to the model while controlling the magnitude of their coefficients. It starts with an empty model and gradually includes features that are most correlated with the target. The regularization term (L1 penalty) encourages some coefficients to become exactly zero, leading to feature selection. LassoLars is useful when identifying important features for prediction, handling high-dimensional data efficiently, and obtaining interpretable models.

36. PassiveAggressiveRegressor

The Passive Aggressive Regressor is an online learning algorithm used for regression tasks. It is part of the family of online learning algorithms, which handle large-scale data or real-time streams. The name "Passive Aggressive" reflects its behavior, which is passive if the prediction is correct and aggressive if the prediction is incorrect. The model adapts to new data points without retraining on the entire dataset. The parameters of the Passive Aggressive Regressor include the regularization parameter C, the maximum number of iterations over the training data, and the stopping criterion for convergence (tol). It is commonly used for tasks like detecting fake news on social media or handling continuous data streams. Examples of using the Passive Aggressive Regressor include creating a model, fitting it to data, and making predictions. It is particularly useful for scenarios where data arrives sequentially and requires adaptive learning.

37. GaussianProcessRegressor

Gaussian Process Regression (GPR) is a non-parametric regression technique used in machine learning and statistics. It is a Bayesian approach that models the relationship between input variables and output as a Gaussian process. GPR captures uncertainty in predictions by considering all possible functions consistent with the data. It treats the target variable as a random function and models the joint distribution of observed data and unobserved function values using a kernel function (also known as covariance function). The kernel defines the similarity between data points and controls the smoothness of the estimated function. Key concepts in GPR include the kernel, hyperparameters, predictive mean and variance, and use cases. It is useful when modeling complex, non-linear relationships, incorporating uncertainty into predictions, and performing regression with limited data.

38. KernelRidge

Kernel Ridge Regression (KRR) is a powerful regression technique that combines ridge regression (linear least squares with l2-norm regularization) with the kernel trick. It learns a linear function in the space induced by the kernel and the data, and for non-linear kernels, it corresponds to a non-linear function in the original space. KRR uses a kernel mapping internally, defining the similarity between data points and controlling the smoothness of the estimated function. It can handle both linear and non-linear relationships. KRR is useful when modeling complex, non-linear relationships, incorporating uncertainty into predictions, and performing regression with medium-sized datasets. A KRR model with an RBF kernel can be created, fitted to data, and made predictions. KRR is particularly useful when a flexible model adapts to the data and captures non-linear patterns.

39. Lars

Least Angle Regression (LARS) is an algorithm used in regression for high-dimensional data, developed by Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. It aims to find relevant features and estimate their coefficients by estimating the coefficients of a linear combination of potential covariates. LARS starts with an empty model and gradually includes features most correlated with the target, increasing the estimated parameters in a direction equiangular to each one's correlations with the residual. LARS is computationally efficient, producing a full piecewise linear solution path, useful for cross-validation or tuning the model. It is stable and intuitively behaves. LARS is particularly effective in contexts where the number of predictors is significantly greater than the number of data points ($p \gg n$). LARS is particularly useful when dealing with high-dimensional data and provides an interpretable solution path.

40. RANSACRegressor

The Random Sample Consensus (RANSAC) Regressor is an iterative algorithm used for robust regression. It is designed to estimate model parameters from a subset of inliers within a larger dataset, effectively handling outliers and noisy data. RANSAC works by randomly selecting a subset of data points (inliers), fitting a model to the inliers, identifying other data points consistent with the model, and repeating the process to find the best model. It is useful for handling data with outliers or noise, estimating model parameters robustly, and removing outliers from a dataset. In Python, the RANSAC Regressor is used to create a RANSAC Regressor, fit it to data, and make predictions. It is particularly useful when dealing with noisy data and outliers.

4.4 Python Packages Implemented

Python

Python is frequently used in the field of machine learning because of its simplicity, accessibility, and rich library support for tasks such as data manipulation and model creation. Python's versatility as a tool for facilitating the building of predictive models is demonstrated by its extensive library collection, which includes NumPy, Pandas, Scikit-learn, TensorFlow, and Keras, which are required for data processing, analysis, and algorithm implementation. Its simple syntax makes it approachable to beginners while still providing the power needed for expert applications. Python also allows object-oriented programming and is cross-platform compatible, rendering it suitable for a variety of machine learning workloads. Machine learning in Python entails teaching computers to learn from data, find patterns, and make predictions without explicit programming. This is accomplished by a variety of machine learning approaches, notably supervised and unsupervised learning, which are supported by Python's computational libraries. Python's contribution to machine learning includes an active community that constantly creates and maintains a myriad of modules and structures, ensuring that Python remains at the forefront of machine learning innovation.

1. chembl_websource_client

A Python interface for querying the ChEMBL database. Enables easy retrieval of chemical data. Supports multiple search types, including compound and target queries. Streamlines the integration of ChEMBL data into bioinformatics processes. Provides a programmatic interface for accessing ChEMBL online services.

2. Pandas

An free analysis of information and manipulation program. Provides data structures for efficiently storing and processing huge datasets. Includes functions for reading, writing, and cleaning data. Supports a variety of file types, including CSV, Excel, SQL, and JSON. Supports complicated data operations such as merging, restructuring, and pivoting.

3. From google.colab import drive

In Google Colab notebooks, a tool is available for mounting Google Drive, providing seamless access to files stored within it. This functionality greatly aids collaborative projects by enabling

easy sharing of datasets and models among team members. Moreover, it offers the advantage of persistent storage across different Colab sessions, ensuring that files remain accessible and intact. This feature streamlines the process of loading data for analysis within Colab, enhancing efficiency and convenience for users.

4. Pubchempy

A Python package created for the PubChem Compound and Substance databases provides a means to programmatically access chemical information. It supports searches based on compound names, formulas, and structures, allowing users to efficiently retrieve relevant data. Furthermore, it enables the retrieval of bioactivity information, offering insights into the properties and functions of different compounds. Additionally, it allows users to download chemical structures in various formats, enhancing its usefulness for a wide range of research and analytical tasks.

5. Miniconda3

An unobtrusive installer has been developed for Conda, a package and environment management system. This installer is characterized by its lightweight nature and rapid installation process, making it particularly suitable for users seeking to install only Conda and its essential dependencies. It simplifies the process of setting up isolated environments tailored to individual projects. Additionally, it ensures cross-platform compatibility across Windows, macOS, and Linux operating systems, catering to a diverse user base with varying system preferences.

6. Rdkit

A cheminformatics toolkit, available as open-source software, offers a comprehensive suite of tools for molecular operations and visualization. It boasts support for an extensive array of chemical file formats and incorporates modules dedicated to molecular descriptors and fingerprints. Furthermore, it seamlessly integrates with machine learning libraries, enabling users to leverage predictive modeling capabilities for various cheminformatics tasks.

7. From rdkit import chem

Utilizing the Chem module from RDKit, this tool provides a range of functionalities for chemical operations. Users can create, edit, and query molecules seamlessly. It supports widely-used chemical notation systems such as SMILES and SMARTS. Additionally, it enables the generation of both 2D and 3D molecular structures. Moreover, it simplifies the process of calculating various molecular properties.

8. From rdkit.chem import descriptors, lipinski

The toolkit incorporates modules aimed at calculating molecular descriptors and Lipinski's Rule of Five, both pivotal in medicinal chemistry and drug discovery endeavors. The descriptors module offers a diverse range of calculations related to molecular properties, while the Lipinski module evaluates drug-likeness based on these properties. This functionality proves invaluable in assessing pharmacokinetic and pharmacodynamic profiles, contributing significantly to the evaluation of potential pharmaceutical compounds.

9. Numpy

An essential component for scientific computations within Python, this package serves as a cornerstone. It facilitates the handling of expansive, multi-dimensional arrays and matrices, along with an extensive repertoire of mathematical functions designed for array manipulation. Its optimization for performance, leveraging C and Fortran code, ensures efficient execution of numerical operations. Crucial for numerical computations across diverse scientific domains, this package underpins critical research and analysis endeavors.

10. From numpy.random import seed

Setting the seed for NumPy's random number generator is a crucial step that guarantees the reproducibility of random operations. This functionality proves invaluable for debugging and testing endeavors, ensuring consistent results across various runs. Particularly essential for initializing stochastic processes within simulations, setting the seed ensure reliability and consistency in the outcomes generated.

11. From numpy.random import random

The creation of random floats within the half-open interval [0.0, 1.0) constitutes a valuable feature with broad applicability in simulations and random sampling tasks. This functionality is an integral component of NumPy's comprehensive random module, which encompasses a diverse array of random number generators. Its versatility allows for the generation of random arrays of any desired shape, catering to a wide range of computational needs. Particularly essential for stochastic processes and Monte Carlo methods, this capability plays a pivotal role in various scientific and mathematical simulations.

12. From scipy.stats import mannwhitneyu

The Mann-Whitney U test is a statistical method utilized to evaluate whether two independent samples originate from identical distributions, presenting a nonparametric alternative to the t-test in scenarios where data fails to meet the t-test's assumptions. Notably, this test is applicable even with small sample sizes and does not necessitate the data to conform to a normal distribution. Its significance lies in its contribution to hypothesis testing within the realm of statistics, providing a robust analytical tool for assessing differences between sample populations.

13. Seaborn

An advanced Python data visualization library, built upon the foundation of Matplotlib, offers a user-friendly interface for crafting visually appealing statistical graphics. It introduces pre-designed themes to effortlessly style Matplotlib graphics, streamlining the creation of intricate visualizations such as heatmaps, violin plots, and pair plots. Seamlessly integrating with Pandas data structures, this library enhances the visualization capabilities of Python, empowering users to convey complex insights with clarity and elegance.

14. Matplotlib.pyplot

This library serves as a powerful tool for generating static, interactive, and animated visualizations within the Python environment. Its pyplot module, resembling the command style functions of MATLAB, facilitates Matplotlib's operation. With its comprehensive features, users gain significant control over the elements present in their figures, enabling precise customization. Matplotlib is a popular choice for producing a diverse range of graphs and charts, thanks to its versatility. Additionally, it boasts support for multiple backends, ensuring compatibility across different platforms and environments.

15. From rdkit.Chem.Fingerprints import FingerprintMols

This tool is employed to create molecular fingerprints, distinctive identifiers crucial for distinguishing molecules. These fingerprints play a vital role in conducting similarity searches and studying structure-activity relationships. RDKit offers a variety of fingerprint algorithms, simplifying the process of comparing chemical structures. Particularly valuable in cheminformatics, these fingerprints are instrumental in tasks such as screening databases and clustering compounds based on their structural similarities.

16. PyBioMed

This Python library serves as a valuable resource for extracting diverse biochemical and pharmacological characteristics. It offers a range of tools tailored for analyzing biological sequences and chemical compounds effectively. Capable of computing molecular descriptors, fingerprints, and ADMET properties, it proves indispensable in computational drug discovery and bioinformatics endeavors. Moreover, its compatibility with other cheminformatics tools such as RDKit enhances its utility and flexibility in various research and analytical tasks.

17. From PyBioMed.PyMolecule.PubChemFingerprints import calcpubchemFingerAll

This function is designed to compute all PubChem fingerprints associated with a specific molecule. These fingerprints play a crucial role in conducting searches for compounds with similar characteristics. By leveraging these fingerprints, researchers can effectively identify promising drug candidates, facilitating the process of virtual screening in drug discovery efforts. This function is an integral component of the PyBioMed library, a comprehensive resource supporting computational biology and chemistry endeavors.

18. Wget

This free tool offers non-interactive downloading capabilities for files from the internet. It accommodates HTTP, HTTPS, and FTP protocols, allowing users to retrieve files efficiently. Additionally, it includes features like resuming aborted downloads and recursively downloading files, enhancing its functionality. Widely employed in scripts and batch files, it caters to various automation needs. Furthermore, it is compatible with Unix, Windows, and Mac OS X systems, ensuring accessibility across different platforms.

19. Padelzip from GitHub

This pertains to the PaDEL software, a tool utilized to compute molecular descriptors and fingerprints. The compressed version of this software is available for download from GitHub repositories. It encompasses an extensive array of descriptor calculation functionalities. Widely employed in cheminformatics applications for tasks like drug design and property prediction, it seamlessly integrates into diverse cheminformatics workflows, contributing to the efficiency and effectiveness of analyses.

20. From sklearn.model_selection import train_test_split

This function divides arrays or matrices into randomized train and test subsets, providing a valuable means to assess the efficacy of machine learning models. It offers flexibility in allocating data to training and testing sets according to specific requirements. By setting a random state, the split becomes reproducible, ensuring consistent results. This functionality is integrated within the scikit-learn library, a popular choice in the realm of machine learning, further enhancing its accessibility and utility in various analytical tasks.

21. From sklearn.ensemble import RandomForestRegressor

This tool encompasses a random forest regressor designed for regression tasks, functioning as a meta estimator. It fits numerous classifying decision trees across different subsets of the dataset, enhancing predictive accuracy while mitigating overfitting issues. Capable of managing a high volume of features and effectively handling non-linear data, it proves invaluable in diverse analytical contexts. This functionality is integrated into the scikit-learn library, renowned for offering straightforward and effective tools for data mining and analysis purposes.

22. From sklearn.feature_selection import VarianceThreshold

This feature selector serves to eliminate low-variance features, offering a means to enhance both model performance and computational efficiency. It proves beneficial as a preliminary step before model training, aiding in the reduction of data dimensionality. Integrated within the scikit-learn library, it seamlessly interfaces with Python's numerical and scientific libraries like NumPy and SciPy, further bolstering its utility in various data analysis tasks.

23. From sklearn.impute import SimpleImputer

This tool offers fundamental techniques for handling missing values within datasets. It provides options such as mean, median, mode, or a specified constant value to substitute the missing data. Crucial for data preparation before analysis, it plays a key role in ensuring dataset cleanliness. By facilitating the maintenance of data integrity in the presence of incomplete datasets, it contributes to more robust analyses. This functionality is integrated into the scikit-learn library, an open-source machine learning toolkit renowned for its versatility and effectiveness.

24. lazy predict

This Python library offers an efficient approach to modeling, characterized by its simplicity and speed. It proves advantageous for generating baseline models with minimal coding effort, facilitating quick comparisons across a diverse array of models to identify the most suitable one. Suitable for tasks encompassing both classification and regression, it supports rapid prototyping and streamlines the process of model selection, empowering users with efficient tools for data analysis and decision-making.

25. From lazy_predict.Supervised import LazyRegressor

The lazy_predict library includes a class designed to automate the fitting of numerous regression models, streamlining the process of model evaluation. This functionality offers a swift overview of the performance of various models with respect to a given dataset, significantly reducing time during the initial testing phase. Its straightforward interface simplifies the task of comparing different models, providing valuable insights into identifying promising candidates that may require further refinement. This tool proves invaluable for efficiently pinpointing models worthy of additional fine-tuning efforts.

26. Import sys

Including the sys module in your Python script grants access to variables managed by the interpreter, enabling interaction with the Python runtime environment. This module empowers users to manipulate various aspects of the runtime environment, including the Python module search path. Its functions are instrumental in system-level programming tasks, making it an indispensable tool for developers working on projects that require direct interaction with the underlying operating system environment

5. RESULT & DISCUSSION

During the forecasting of PIC50 values for the CDK1 protein, the chemical compounds employed for prediction are retrieved and organized into a data frame table. Approximately 1689 compounds meeting the criteria of adhering to the Lipinski rule of five are extracted and listed in table one.

```
[ ] df
```

	molecule_chembl_id	canonical_smiles	standard_value	bioactivity_class
0	CHEMBL95827	COc1ccc2[nH]c3c(c2c1)CC(=O)Nc1ccccc1-3	900.0	active
1	CHEMBL420455	CC(C)(C)OC(=O)C1C(=O)N(C(=O)OC(C)(C)C)c2ccccc2...	150000.0	inactive
2	CHEMBL100312	CC(C)(C)OC(=O)n1c2c(c3cc(Br)ccc31)CC(=O)Nc1ccc...	70000.0	inactive
3	CHEMBL296586	O=C1Cc2c([nH]c3ccc(Br)cc23)-c2ccccc2N1	400.0	active
4	CHEMBL98360	O=C1Cc2c([nH]c3ccc(Br)cc23)-c2cc(Br)ccc2N1	300.0	active
...	...	...	...	...
1684	CHEMBL5177698	CNc1nc(C(=O)/C=C/c2ccc(OC)cc2)c(C)s1	1470.0	intermediate
1685	CHEMBL133342	CC[C@@H](CO)Nc1nc(NCc2ccccc2)c2ncn(C(C)C)c2n1	650.0	active
1686	CHEMBL1230165	O=C(O)c1ccc2c(c1)nc(Nc1cccc(Cl)c1)c1ccncc12	10.0	active
1687	CHEMBL1230165	O=C(O)c1ccc2c(c1)nc(Nc1cccc(Cl)c1)c1ccncc12	56.0	active
1688	CHEMBL1231206	Cc1ccc(-c2ccc3c(ccc4sc5c(c43)NC[C@@H](C)NC5=O)...	354.0	active

1689 rows × 4 columns

Figure 5.1 shows compounds meeting the criteria of adhering to the Lipinski rule of five

Utilizing these bioactive compounds, the PIC50 values will be computed by identifying the IC50 values within the bioactivity compounds dataset. Subsequently, correlations between the PIC50 values and 1421 potential compounds will be established to facilitate the training of the machine learning model. Figure 5.2 shows some of the compounds from the dataset.

	PubchemFP0	PubchemFP1	PubchemFP2	PubchemFP3	PubchemFP4	PubchemFP5	PubchemFP6	PubchemFP7	PubchemFP8	PubchemFP9	...	PubchemFP872	PubchemFP873	PubchemFP874	PubchemFP8
0	1	1	0	0	0	0	0	0	0	1	...	0	0	0	
1	1	1	1	1	0	0	0	0	0	1	...	0	0	0	
2	1	1	1	0	0	0	0	0	0	1	...	0	0	0	
3	1	1	0	0	0	0	0	0	0	1	...	0	0	0	
4	1	1	0	0	0	0	0	0	0	1	...	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1416	1	1	1	1	0	0	0	0	0	1	...	0	0	0	
1417	1	1	1	0	0	0	0	0	0	1	...	0	0	0	
1418	1	1	0	0	0	0	0	0	0	1	...	0	0	0	
1419	1	1	0	0	0	0	0	0	0	1	...	0	0	0	
1420	1	1	1	0	0	0	0	0	0	1	...	0	0	0	

1421 rows × 882 columns

chemFP6	PubchemFP7	PubchemFP8	PubchemFP9	...	PubchemFP872	PubchemFP873	PubchemFP874	PubchemFP875	PubchemFP876	PubchemFP877	PubchemFP878	PubchemFP879	PubchemFP880	pIC50
0	0	0	1	...	0	0	0	0	0	0	0	0	0	6.045757
0	0	0	1	...	0	0	0	0	0	0	0	0	0	3.823909
0	0	0	1	...	0	0	0	0	0	0	0	0	0	4.154902
0	0	0	1	...	0	0	0	0	0	0	0	0	0	6.397940
0	0	0	1	...	0	0	0	0	0	0	0	0	0	6.522879
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
0	0	0	1	...	0	0	0	0	0	0	0	0	0	8.045757
0	0	0	1	...	0	0	0	0	0	0	0	0	0	6.187087
0	0	0	1	...	0	0	0	0	0	0	0	0	0	8.000000
0	0	0	1	...	0	0	0	0	0	0	0	0	0	7.251812
0	0	0	1	...	0	0	0	0	0	0	0	0	0	6.450997

Figure 5.2 shows some of the compounds from the dataset

Following the application of different machine learning algorithms on the PIC50 values dataset, The R-squared values and RMSE can be used to assess the accuracy of machine learning methods used to predict the PIC50 values of the target protein CDK1. By comparing the algorithm values. The R-Squared Values, also known as the coefficient of determination, are a measure that indicates how well our machine learning model matches the given data. When plotting data points on a scatter plot and drawing a line across them, the R-squared value indicates their proximity to the model's projected line.

The R-squared value varies from zero to one. If R-squared is one, the model accurately predicts all data points. Each data point is exactly on the line drawn. If R-squared is zero, the model is no better at predicting data points than just drawing a horizontal line through the average of all data points. If R-squared is between 0 and 1, the model is somewhere in the middle, accounting for a given amount of the data's variability. So, the greater the R-squared number, the better the model's ability to explain data variation.

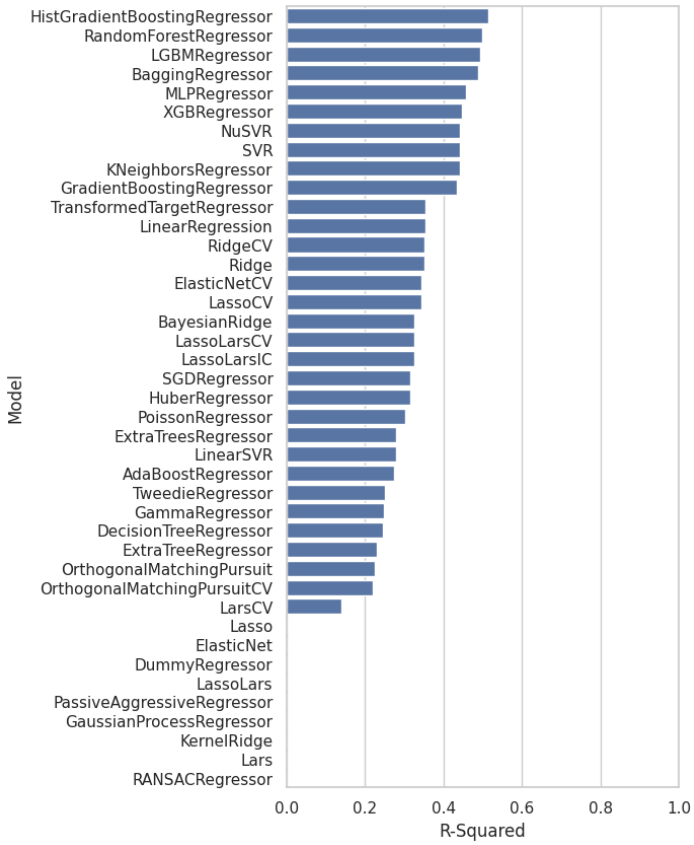


Figure 5.3 shows R-Squared Value based Graph

Secondly Root Mean Squared Error, or RMSE, is an effective method for assessing the effectiveness of a machine learning model. In a nutshell, RMSE evaluates the difference between the actual values in the dataset and the ones predicted by the model. This is performed by computing the mean variance of the expected and observed values. In order to guarantee that positive deviations do not cancel out negative ones, RMSE calculates the square root of the average of these squared variances after squaring each individual difference. When it comes to measuring average variances between predictions and actual data, RMSE offers a single statistic. A ten-unit average deviation from true values is inferred, for example, if the RMSE is 10.

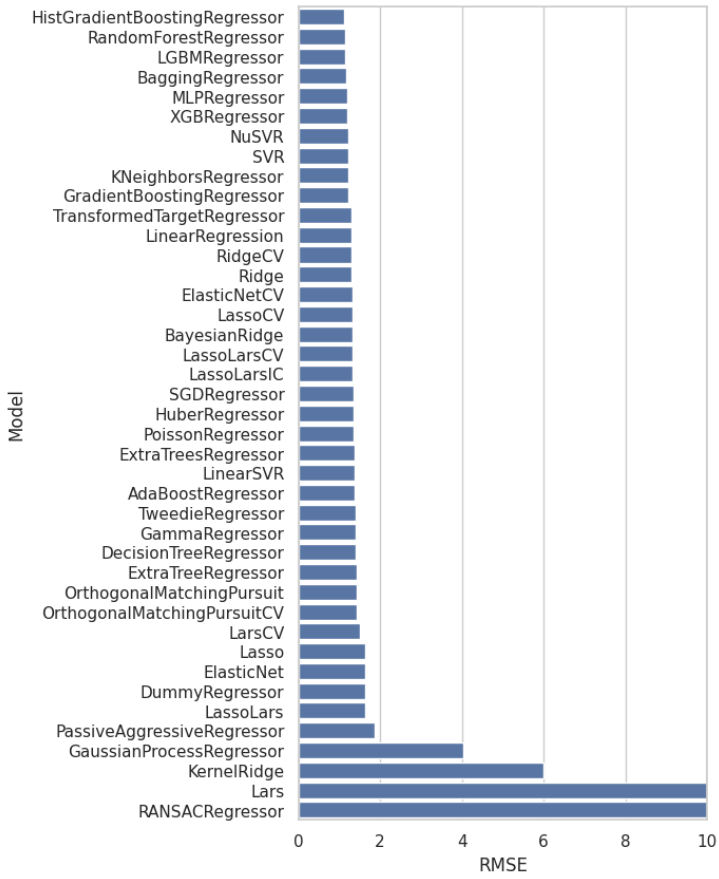


Figure 5.4 shows RMSE values based Graph

The **HistGradientBoostingRegressor** algorithm outperforms the other machine learning algorithms based on comparison of its R-squared and RMSE values with different criteria from the Score Table below. HistGradientBoostingRegressor has an RMSE value of 0.51 and an R-Square value of around 1.13.

Score Table

Model	**Adjusted R-Squared value**	**R-Squared Value**	**RMSE**	**Time Taken**
HistGradientBoostingRegressor	0.07	0.51	1.13	0.88
RandomforestRegressor	0.05	0.50	1.15	1.24
LGBMRegressor	0.04	0.49	1.15	0.20
BaggingRegressor	0.02	0.49	1.16	0.15
MLPRegressor	-0.04	0.46	1.20	3.85
XGBRegressor	-0.05	0.45	1.21	0.24

NuSVR	-0.06	0.44	1.21	0.21
SVR	-0.06	0.44	1.21	0.25
KNeighborsRegressor	-0.06	0.44	1.21	0.11
GradientBoostingRegressor	-0.08	0.43	1.22	0.53
TransformedTargetRegressor	-0.23	0.35	1.31	0.04
LinerRegression	-0.23	0.35	1.31	0.07
RidgeCV	-0.24	0.35	1.31	0.05
Ridge	-0.24	0.35	1.31	0.02
ElasticNetCV	-0.25	0.35	1.31	5.06
LassoCV	-0.25	0.34	1.32	3.69
BayesianRidge	-0.29	0.33	1.33	0.09
LassoLarsCV	-0.29	0.33	1.33	0.26
LassoLarsIC	-0.29	0.33	1.33	0.10
SGDRegressor	-0.30	0.32	1.34	0.04
HuberRegressor	-0.30	0.32	1.34	0.12
PoissonRegressor	-0.33	0.30	1.35	0.43
ExtraTreesRegressor	-0.37	0.28	1.38	1.50
LinerSVR	-0.38	0.28	1.38	0.30
AdaBoostRegressor	-0.38	0.27	1.38	0.18
TweedieRegressor	-0.42	0.25	1.40	0.08
GammRegressor	-0.43	0.25	1.41	0.14
DecisionTreeRegressor	-0.44	0.25	1.41	0.10
ExtraTreeRegressor	-0.47	0.23	1.42	0.09
OrthogonalMatchingPursuit	-0.47	0.23	1.43	0.03
OrthogonalMatchingPursuitCV	-0.48	0.22	1.43	0.05
LarsCV	-0.64	0.14	1.50	0.28
Lasso	-0.91	-0.00	1.62	0.04
ElasticNet	-0.91	-0.00	1.62	0.02
DummyRegressor	-0.91	-0.00	1.62	0.03
LassoLars	-0.91	-0.00	1.62	0.05
PassiveAggressiveRegressor	-1.51	-0.32	1.86	0.04
GaussianProcessRegressor	-10.80	-5.91	4.04	0.41
KernelRidge	-25.11	-12.70	6.01	0.13
Lars	-1121.72	-588.03	39.40	0.11
RANSACRegressor	-579978062925424322674688.00	-304284265408057151848448.00	895576856490.44	1.22

To ensure the accuracy of the generated values, the PIC50 values produced by a machine learning model must be compared to experimental PIC50 values. Creating a scatter plot with both sets of data is one step in the verification process. The shown points form a diagonal line linking the experimental and projected values, indicating that the machine learning model is predicatively successful.

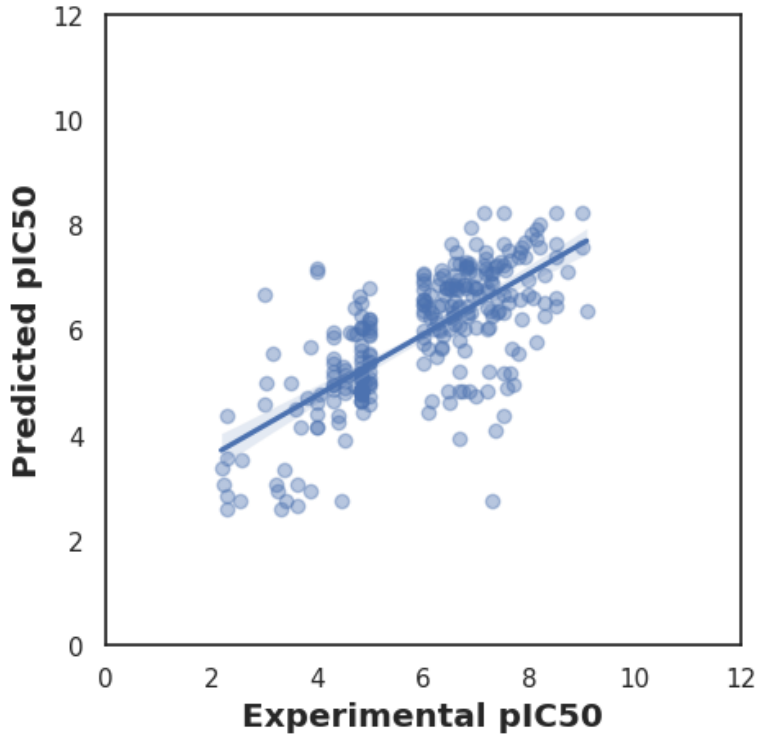


Figure 5.5 shows comparing Experimental and Projected PIC50 Values

6. CONCLUSION

In this study, an in-depth analysis was conducted on the bioactivity compounds associated with the target protein CDK1. Initially, the IC50 values were identified, representing the concentration of a compound required to inhibit 50% of the target protein's activity. Subsequently, these IC50 values were transformed into PIC50 values, a logarithmic scale commonly used in pharmacology to simplify data interpretation, where higher PIC50 values correspond to greater potency. Utilizing these experimental PIC50 values as the benchmark, the endeavor was to construct a predictive machine learning model. The aim of this model is to forecast PIC50 values based on the bioactivity compounds of CDK1. To accomplish this, various machine learning algorithms were employed to train and test the model. The training process involved using a portion of the experimental PIC50 values as input data to teach the model to recognize patterns and relationships within the dataset. The remaining portion of the experimental PIC50 values was then utilized to evaluate the model's predictive performance, serving as the testing dataset. Through this iterative process, our objective is to develop a robust machine learning model capable of accurately predicting PIC50 values for CDK1 based on its bioactivity compounds. This model holds significant potential in aiding drug discovery and development efforts by providing insights into the potency of potential drug candidates targeting CDK1, thereby facilitating the identification of promising therapeutic agents for the treatment of diseases such as cervical cancer. After generating projected PIC50 values with machine learning algorithms, it is critical to compare the accuracy of these predictions to experimental PIC50 values. This comparison is an important step in determining the machine learning model's dependability and efficacy in predicting the potency of drugs that target the CDK1 protein. To depict this contrast, a scatter plot is used. Scatter plots are commonly used in data analysis to show the relationship between two variables. In this case, the experimental PIC50 values are represented on one axis (usually the x-axis), while the anticipated PIC50 values are plotted on the other axis (usually the y-axis). Each data point on the scatter plot represents a chemical, and the coordinates indicate the experimental and projected PIC50 values. A perfect prediction would have all data points sitting on a diagonal line with a slope of one, demonstrating complete agreement between the experimental and anticipated values. Deviations from this ideal line reflect differences between expected and experimental values for certain chemicals. In the future, deploying the model as a web-based application signifies a transition from a research phase to a practical, real-world utility. By making the model accessible through a web interface, it can be utilized by researchers, pharmaceutical companies, or healthcare professionals without requiring advanced technical expertise in machine learning or data analysis. The web-based application would likely offer a user-friendly interface where users can input data related to bioactivity compounds targeting the CDK1 protein, and the model would provide predicted PIC50 values as output. This interface could include features such as data validation to ensure input consistency,

visualizations to aid in result interpretation, and possibly even integration with databases or other external resources for comprehensive analysis.