Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Cancer is still a major problem for world health, hence efforts must be made to enhance the effectiveness of therapy. Chemotherapy is a common traditional cancer treatment procedure that frequently results in variable patient outcomes, stressing the need for more individualised and focused approaches. In this setting, machine learning approaches have shown promise for increasing the study of and therapy for cancer.

Finding lead chemicals that have a favourable effect on particular characteristics associated with cancer is one of the main objectives of cancer research. In this context, high-throughput screening of many compounds has gained popularity. Recent research has shown how well vast panels of cancer cell lines with various genomic origins and growth environments may be screened against chemical libraries. Notably, the National Cancer Institute's "NCI-60" collection, which consists of 59 human cancer cell lines developed for in vitro drug screening, has paved the path for research into the underlying genetic causes of drug sensitivity.

This research has a wide range of practical applications. The prediction models and algorithms created through this study may be used in clinical settings, assisting oncologists in selecting the best course of treatment for specific patients. We can improve treatment outcomes, cut healthcare costs by avoiding ineffective medications, and lessen patient suffering by adapting therapies to the unique genetic and chemical profiles of patients. In the end, our project hopes to have a real influence on cancer research and treatment, helping to advance the ongoing effort to better understand and treat this complicated disease

1.2 Problem Statement

Cancer treatment is a difficult and complex procedure that frequently necessitates a ”trial-and-error” strategy to determine the best medications for specific individuals.The conventional approaches of choosing cancer therapies rely on broad principles and empirical data, which may not sufficiently take into account the distinctive traits and genomic variations of each patient's cancer cells.To solve this issue :

- We aim to develop a machine learning-based prediction model that can accurately forecast the sensitivity of cancer cells to specific drugs.
- We aim to apply machine learning algorithms which will analyze complex relationships between genomic and chemical factors and the corresponding drug responses. Through this comprehensive analysis, we aim identify significant features and patterns that can serve as predictive indicators of drug sensitivity in cancer cells.
- By integrating genomic and chemical data, our model will enable oncologists to make more informed decisions regarding treatment selection, thereby increasing the likelihood of positive outcomes while minimizing side effects.

1.3 Organization

The rest of the report is organized as follows. We review the existing literature in Chapter 2 and find the problems with the existing techniques. In chapter 3, we go through the objectives.In chapter 4, we take a look at brief introduction of Machine Learning. In chapter 5, we discuss the implementation details of the project, explaining all its features, and specified the code snippets wherever required. In chapter 6, we have discussed various results obtained. Finally in chapter 7, we have briefly summarized the project.

Chapter 2

Review of Existing work

2.1 Literature Survey

In this chapter, we survey the recent literature to gain some insights into the prevailing research which has been done in the field. Based on each paper, certain observations and inferences were made and thus our groundwork for the project was started.

2.1.1 Based on multiple molecular data, a computational technique for cancer cell line drug sensitivity prediction

This study uses computational methods, which may be divided into classification and regression approaches, to forecast drug response. Regression approaches try to forecast the value of a certain criterion that assesses a cell line's reaction to a medication, whereas classification methods aim to predict the sensitivity of drug-cell line pairings. To handle prediction challenges, a variety of computer approaches have been developed in the literature.For instance, a heterogeneous network-based strategy was presented by Zhang et al. that takes into account drug-target relationships, drug sensitivity of cell lines, drug similarities, cell line similarities, and Protein-Protein Interaction (PPI) networks. The approach they use, called HNMPRD, makes use of an information flow-based algorithm to forecast new sensitive cell line-drug combos. To improve forecast accuracy, this strategy integrates several data sources.

The RefDNN computational model, developed by Choi et al., combines a deep neural network with ElasticNet regressors. The association between genetic and chemical characteristics and

the pharmacological response of cell lines is captured by this model, which makes use of a large number of ElasticNet regressors. RefDNN seeks to increase the precision of drug response prediction by utilising the strength of deep learning and regularisation approaches. These methodology and approaches illustrate the wide range of computational techniques used to predict drug response that are covered in the literature review. Researchers have created prediction models with the capacity to recognise sensitive cell line-drug pairings by combining a variety of data sources, such as network information, chemical substructures of pharmaceuticals, and transcriptome characteristics of cell lines.

In their research, the authors suggested a way for determining a cell line's sensitivity to several drugs. They used a group of reference medications as a standard for categorising pharmaceuticals. They calculated the possibility that a cell line would show sensitivity to a given drug by comparing that drug's resemblance to the reference set. This strategy makes it possible to reposition anti-cancer medications using the created model.The DSPLMF approach is a significant addition to the field of cell line-drug categorization. This method forecasts the sensitivity using logistic matrix factorization. Three different types of cell line similarity and drug similarity based on chemical structure are both included in the model's regularisation terms. The DSPLMS also takes into account the similarities between cell lines by taking into account cell line response ratings to various medications.

Wang et al.investigated the association between drug and cell line similarity in predicting drug response values in their research. They proposed SRMF, a regularization-based matrix factorization technique that takes into account the chemical similarity of medications and the gene expression similarity of cell lines. Their research centred on the utilisation of repurposed drugs in lung cancer cell lines. Similar to this, CaDRReS, a recommender system that depends on cell line similarities, was created by Suphavilai et al. Based on anticipated medication responses, they showed that CaDRReS can extract insightful information regarding pharmacological processes.

Wei et al. unveiled CDCN, a simple network-based strategy that makes use of the connection between cell lines and medications to forecast treatment response. Despite being straightforward, CDCN had excellent outcomes. While employing manifold learning, Ahmadi Moughari et al. presented ADRML, a framework for predicting the effectiveness of anti-cancer medications. ADRML uses numerous learning approaches while taking into account various cell line and drug similarities to map drug response values into a low-dimensional latent space. The ability of ADRML to correctly forecast drug pathway activities and related effects was demonstrated by the authors.

In addition to utilising previously established similarities, this research provides two brand-new and thorough similarities for anti-cancer medications. For precision medicine models and related fields, the inclusion of these novel drug similarities, which include many sources of information, can be beneficial. The evaluation of the effect of each similarity type on the effectiveness of the suggested model is an intriguing part of this study. Additionally, the proposed solution uses a novel method that combines standardisation and normalisation to improve the attributes of the similarity matrices.

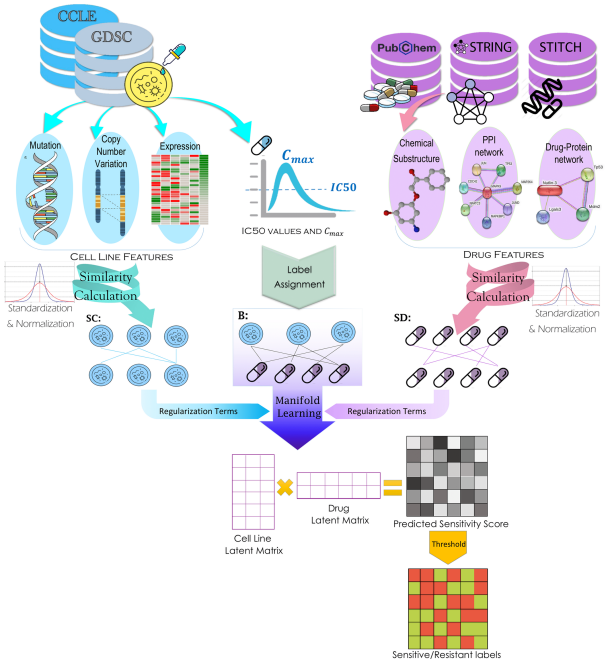

Figure 2.1: Working of method

The method set forward in this study provides flexibility in handling missing data in a variety of ways, either with or without the use of an imputation mechanism. The developed code enables users to use the recommended method in a variety of configurations, such as those with no similarity, only one sort of similarity, or both similarities between cell lines and drugs. The code automatically adapts by using the appropriate loss function and optimisation strategy based on the similarity information the user selects.This study also examined the effect of drug and cell line similarity on the performance of the classification model, revealing insight on the significance of drug and cell line similarity. The study carefully analysed the prediction performance in a variety of conditions, including those in which no similarity was taken into account, in which only cell line similarity was used, in which only drug similarity was taken into account, and in which both cell line and drug similarities were taken into account.

The authors of this work suggested a novel strategy to improve the properties of similarity matrices and deal with negative similarity values. To convert the similarity data into more useful

matrices with desired features, they advised combining standardisation and normalisation procedures. This combination was particularly significant because the convergence of the model is strongly influenced by the algebraic and spectral features of matrices utilised in manifold learning. The authors provided a novel and useful solution in this context by introducing the novel idea of combining symmetric Laplacian normalisation with standardisation.

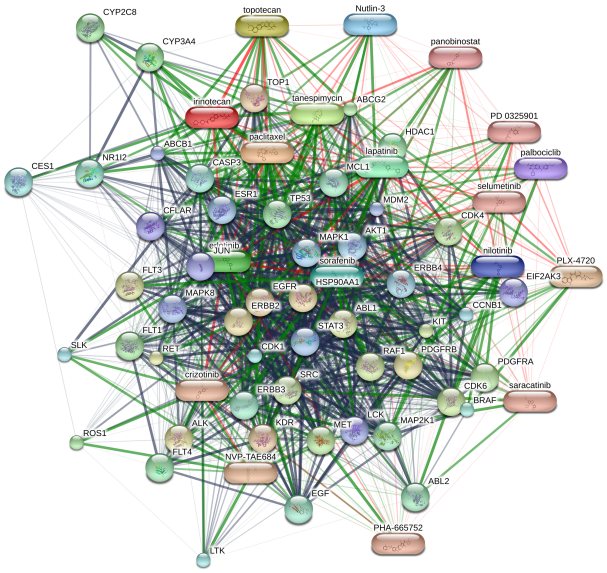

Figure 2.2: Network for drugs in dataset

Additionally, the authors calculated two novel pharmacological similarities using the STiTCH network's Jaccard index and the target Protein-Protein Interaction (PPI) network that was taken from the STRING database with the highest degree of similarity. The updated manuscript addresses additional concerns brought up during the review process and includes a full description of these distinct types of pharmacological similarities. It is important to note that the suggested method for figuring out these pharmacological similarities combines data from a variety of sources, producing thorough and extremely useful comparisons for pharmaceutical molecules. There is little information available about the features and qualities of the pharmaceuticals included in the GDSC dataset because a sizable portion of them do not have FDA approval.

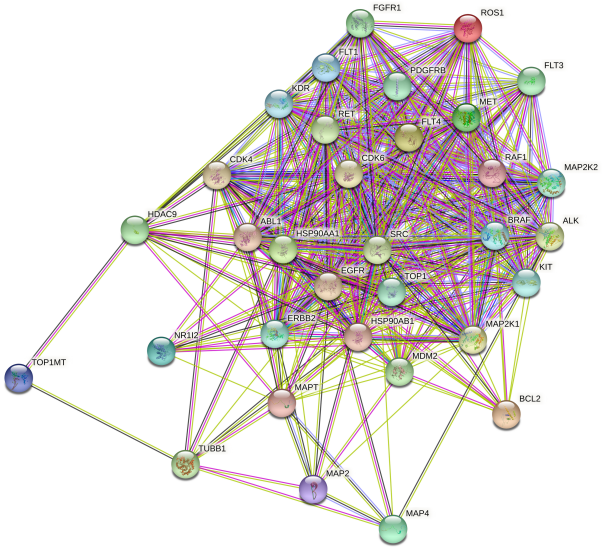

Figure 2.3: String network for drugs

	SC	SD	AUC	AUPR	Accuarcy	F1-score	Precision	Recall
No sim	-	-	0.8542	0.8846	0.7812	0.8291	0.7706	0.8974
Single similarity	SC_E	-	0.9147	0.9372	0.8359	0.8653	0.841	0.891
	SC_M	-	0.9148	0.9373	0.8356	0.8652	0.8401	0.892
	SC_V	-	0.9071	0.9353	0.8288	0.8636	0.8346	0.8948
	-	SD_S	0.9071	0.9353	0.8288	0.8636	0.8346	0.8951
	-	SD_N	0.9072	0.9354	0.83	0.8637	0.8392	0.8899
	-	SD_P	0.9071	0.9353	0.8292	0.8638	0.8357	0.8943
Double similarity	SC_E	SD_S	**0.9158**	0.9373	**0.8388**	0.8714	**0.8426**	0.9026
	SC_E	SD_P	0.9157	0.9373	0.8354	0.8714	0.8423	0.903
	SC_E	SD_N	0.9157	**0.9398**	**0.8388**	**0.8715**	0.8422	**0.9031**
	SC_M	SD_S	0.9148	0.9373	0.8357	0.8652	0.8402	0.8918
	SC_M	SD_P	0.9147	0.9373	0.8354	0.8652	0.8392	0.8929
	SC_M	SD_N	0.9147	0.9373	0.8350	0.8652	0.8388	0.8949
	SC_V	SD_S	0.9157	0.9397	**0.8388**	0.8714	**0.8426**	0.9026
	SC_V	SD_N	0.9157	**0.9398**	0.8387	0.8714	0.8424	0.9028
	SC_V	SD_P	0.9147	0.9372	0.8355	0.8653	0.8390	0.8934

Figure 2.4: Performance of CDSML on GDSC dataset using different scenarios

2.1.2 Predicting medication response using cancer cell lines

7 July 2016 – According to a study that was published in the journal Cell, patient-derived cancer cell lines display genetic changes that are comparable to those seen in real patient tumours. This insightful information makes it possible to forecast how well brand-new medications will work against tumours. The study, carried out jointly by the European Bioinformatics Institute (EMBL-EBI), the Wellcome Trust Sanger Institute, and the Netherlands Cancer Institute, establishes a direct link between particular mutations found in cancer patient samples and their responsiveness to therapies. These results have the potential to dramatically advance personalised cancer treatments by giving doctors useful information to forecast the best possible course of action and locate clinical trials that are appropriate for certain patients.

Researchers effectively combined patient genetic data, lab cancer cell lines, and data on treatment sensitivity in a ground-breaking and extensive study. To find genetic changes linked to cancer, the study examined more than 11,000 patient samples and 29 different tumour types. A huge collection of 1,000 cancer cell lines was used to map the discovered genetic anomalies. The researchers performed extensive testing on the cell lines employing 265 different cancer medicines to assess the effects of these modifications on drug sensitivity. They were able to identify the precise alterations that affected the cell lines' response to various treatments thanks to their thorough research.

The research team came up with two important conclusions. First, they discovered that laboratory cancer cell lines exhibit the same molecular aberrations as patient cancer samples do. This suggests that the cell lines can act as useful models for identifying the most efficient medications for particular individuals. Second, the scientists showed that molecular aberrations, whether singular or combined, found in thousands of patient cancer samples, had a significant impact on how cancer cells react to particular medications. These results underline how crucial it is to comprehend the molecular features of tumours in order to develop efficient therapeutic strategies.

Prior research has concentrated on identifying the molecular aberrations connected to cancer cell biology by sequencing the DNA of patient tumours. Additionally, studies have demonstrated that sizable collections of cancer cell lines created in the lab can be used to assess how different treatments respond to them. This study stands out as the first to systematically integrate these two datasets, though. To acquire a thorough understanding of the connection between molecular aberrations and drug susceptibility in cancer, the researchers combined the genetic data from patient tumours with the drug sensitivity profiles of cancer cell lines.

Figure 2.5: Combining molecular information from patient samples cancer cell lines. [Graphic concept by Fran Irio, Spcer Lips' illustration]

2.1.3 GDSC: a resource for the identification of therapeutic biomarkers in cancer cells

A key component of the research, according to this paper, was the integration of significant genetic and drug sensitivity information into the GDSC database. Two complimentary analytical methods are used to investigate genetic markers linked to medication response. A multivariate analysis of variance (MANOVA) is used to investigate the relationship between various genomic alterations frequently observed in cancer, such as point mutations, gene amplifications, gene deletions, gene rearrangements, and microsatellite instability, and drug sensitivity measurements (such as IC50 values and dose-response curve slopes). By providing both the size of the effects and the statistical significance of each drug-gene interaction, the MANOVA analysis enables the discovery of particular genetic traits that are strongly associated with drug sensitivity.

The researchers used elastic net regression, a penalised linear modelling technique, to find a wide range of interrelated genetic characteristics that affect therapy response. The tissue type, the genomic data used in the prior MANOVA study, and genome-wide transcriptional profiles

were all included in this analysis. The researchers identified which of these characteristics are related to drug responsiveness, as assessed by IC50 values throughout the cell line panel, by using the elastic net approach. A list of mutations, transcripts, and tissue types was created for each drug, and an effect size was given to each of these characteristics.

The database makes use of interactive graphics to make it easier to understand the data. Users can conduct database searches based on ”Compounds” or ”Cancer Genes” on the webpage’s ”Browse our data” section. When you perform a search for ”Compounds,” a list of drug names is given together with their synonyms, prospective therapeutic targets, the number of analysed cell lines (sample size), and the date of the most recent data update. Chemical structures are accessible thanks to the database’s connection to the PUBCHEM database. The individual drug page, which offers details on drug sensitivity and genetic association, can be reached by clicking on the name of a particular medication. The same thing happens when you search for ”Cancer Genes”; you get a list of cancer genes arranged by HUGO nomenclature. Direct links to the gene’s COSMIC page and the UniProt databases, which provide further protein information, are provided on this page. Users can reach the particular gene page, which offers details on drug sensitivity and genetic association, by clicking on a specific gene name.

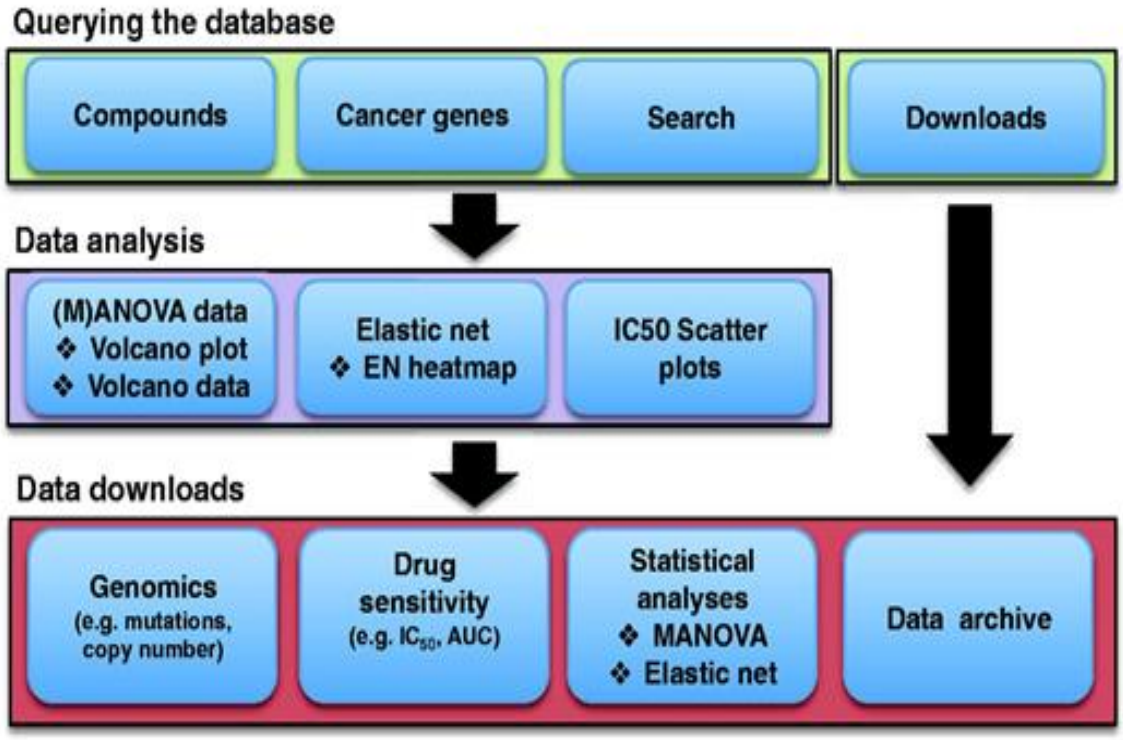

Figure 2.6: An illustration of the database’s layout and contents. Data can be retrieved by screening compound or cancer gene of interest

We introduce the GDSC database as a useful tool for locating therapeutic biomarkers in cancer cells. The database's enormous collection of publicly accessible data on cell line anticancer drug sensitivity is one of its standout qualities. This database can be used by researchers as a reliable source of information for researching and examining the connection between drug sensitivity and cancer cell lines.

The goal of the GDSC database is to uncover possible therapeutic biomarkers for additional preclinical testing by providing a singular integration of comprehensive genetic datasets and drug sensitivity data. Researchers can study the vast drug sensitivity and genetic databases using the database's freely available data, which is presented in user-friendly graphical representations. With the scope and complexity of these datasets, it is anticipated that the GDSC database would experience tremendous growth in the years to come.

The GDSC database's main goal is to facilitate the preclinical identification of therapeutic biomarkers, which will ultimately speed up the development of new cancer treatments. New cancer treatments are now being developed, which is a difficult, expensive, and time-consuming procedure. Researchers want to hasten the discovery and creation of novel cancer medicines by utilising the tools and information offered by the GDSC database.

2.2 Problems with existing Techniques

2.2.1 Limited throughput

Traditional methods frequently had low throughput, which meant they could not process many samples at once. This made it time-consuming and impractical to examine several cancer cells and determine which medications they were most sensitive to.

2.2.2 Low predictive accuracy

Many earlier methods were based on overly simple theories or presumptions that did not adequately account for the complexity of cancer biology and medication response. Due to the anticipated drug sensitivity not matching up well with the actual trial findings, there was a poor level of predictive accuracy.

2.2.3 Lack of molecular specificity

The molecular heterogeneity of cancer cells was frequently missed by earlier methods. The precise genetic or chemical changes in cancer cells that may affect treatment response were not taken into account. As a result, based on molecular features, these approaches were unable to reliably predict sensitivity or resistance to certain medications.

2.2.4 Inability to capture dynamic changes

The dynamic changes that take place in cancer cells over time were not taken into consideration by certain older approaches. Cancer cells have the potential to adapt and develop medication resistance, making earlier forecasts wrong. Traditional methods frequently failed to take into account the dynamic nature of medication response.

2.2.5 Insufficient integration of multi-omic data

Many older techniques did not effectively integrate diverse types of data, such as genomics, transcriptomics, proteomics, and epigenomics. The integration of multi-omic data can provide a more comprehensive understanding of drug sensitivity. However, older techniques often lacked the ability to leverage these different data sources effectively.

2.2.6 Lack of mechanistic insights

The mechanistic understanding of the underlying biological processes behind medication sensitivity or resistance was frequently hampered by older methodologies. It's essential to comprehend molecular mechanisms in order to create tailored medicines and enhance healing.

2.2.7 Limited clinical translation

Many outdated methods were largely employed for research and had little therapeutic application. They were ineffective in bridging the gap between test results and clinical judgement, which reduced their influence on patient treatment.

2.2.8 High cost and resource requirements

Some conventional methods included labour- and money-intensive experimental procedures that called for specialised tools and qualified personnel. This limited their usability in research and clinical settings and their scalability and accessibility.

2.2.9 Non-standardized experimental conditions

Older methods frequently used non-standard experimental settings, such as different cell culture processes, doses of drugs, and lengths of therapy. It was difficult to compare and duplicate the results across multiple investigations due to the variability created by these inconsistencies.

2.2.10 Limited data integration

Older methods frequently concentrated on a particular type of data or a small number of variables, like gene expression levels or assays based on cell lines. This narrow focus made it difficult to fully comprehend the intricate interplay between the many different elements affecting medication response.

Chapter 3

Objectives

3.1 Objectives

The objevtive of this research project is to create a prediction model for cancer cell sensitivity estimation based on machine learning. To improve the precision of medication response predictions, the model will take advantage of the genomic and chemical characteristics of cancer cells. The following are the study's particular goals:

3.1.1 Data Collection and Preprocessing

• Assemble a complete collection including information on the chemical composition, genetic profiles, and therapeutic responsiveness of many cancer cell lines.

• To ensure that the dataset is suitable for machine learning analysis, do data pretreatment activities such as data cleansing, normalisation, and feature engineering.

3.1.2 Feature Selection

• Investigate multiple feature selection methods to find the most insightful chemical and genetic characteristics related to drug sensitivity.

• If required, apply dimensionality reduction methods, such as principal component analysis (PCA) , to reduce number of feature of the dataset while retaining the essential information.

3.1.3 Model Development and Evaluation

• To predict the responsiveness of cancer cells to various medicines, use machine learning models using decision trees, random forests, support vector machines (SVM), or deep learning architectures.

• Use appropriate model assessment metrics to evaluate the performance and robustness of the constructed models, such as R-Squared Value, Mean Square Error, Mean Absolute Error, and Root Mean Square Error.

• Compare the models' performance against baseline procedures or other published predictive models after validating them using the proper cross-validation techniques.

3.1.4 Interpretation and Insights

• To learn more about the genetic and chemical components affecting cancer cell drug sensitivity, analyse the trained models.

• Determine significant biomarkers, genetic mutations, or chemical characteristics that are closely related to drug response.

• To help domain experts understand and comprehend the prediction models, provide interpretable findings and visualisations.

3.1.5 Deployment and Application

• Use the created prediction model to create an intuitive web page that enables scientists and doctors to forecast the treatment sensitivity of cancer cells based on genomic and chemical information.

• Apply the model to real-world datasets to show how it might be used to inform decisions about a person's particular course of treatment for cancer.

Chapter 4

Machine Learning: A brief Introduction

Machine learning is a branch of artificial intelligence (AI) that focuses on creating statistical models and algorithms that let computers learn and anticipate the future or make decisions without being explicitly programmed. It is focused with the creation and improvement of algorithms that enable computers to see patterns and connections in data and then utilise that understanding to provide precise predictions or take appropriate action.

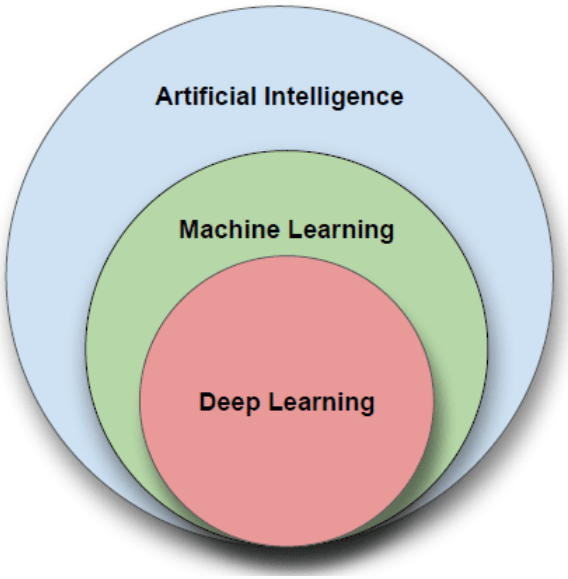

Figure 4.1: Introduction to Machine Learning

Machine learning is fundamentally the study and development of algorithms that can learn from data and make predictions or judgements based on that data. These algorithms are made to automatically decipher and analyse intricate links and patterns in the data, allowing the computer to extrapolate from and make predictions on previously unexplored data.Machine learning al-

gorithms come in a variety of forms, such as supervised learning, unsupervised learning, and reinforcement learning.

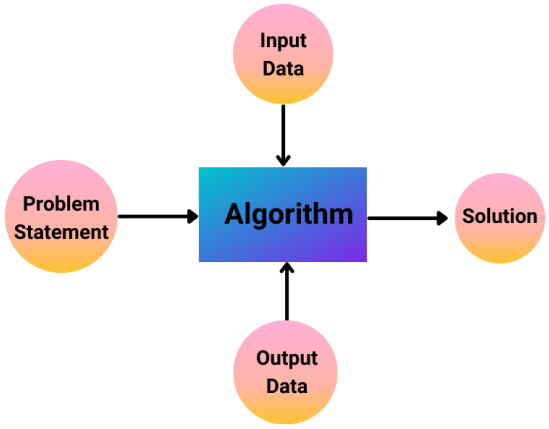

Figure 4.2: Basic Structure

• In supervised learning, each data point is connected to a predetermined output or label, and the system is trained on this labelled data. The computer gains knowledge from this labelled data and can then predict outcomes from fresh, untainted data.

• The algorithm discovers patterns and structures in the data without explicit labelling in unsupervised learning, which works with unlabeled data.

• Training an agent to interact with the environment and discover via mistake how to maximise a reward signal is known as reinforcement learning.

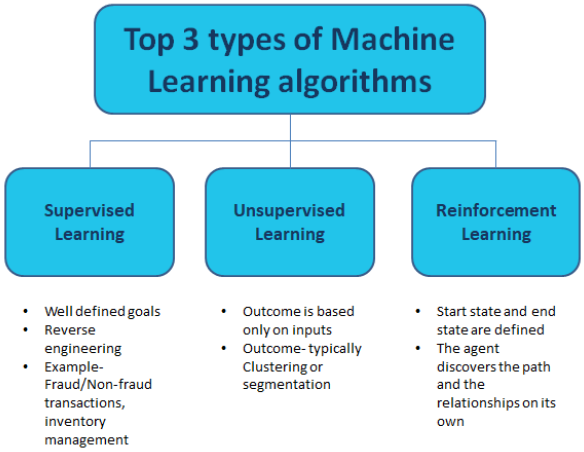

Figure 4.3: Types of ML Algorithms

Three key subfields of machine learning—supervised, unsupervised, and reinforcement learning—each have their own benefits and uses. Which method is deemed ”good” depends on the nature of the issue and the information at hand.

When there is access to labelled training data, supervised learning works well. To identify patterns and generate predictions using this method, examples with labels must be provided. Prediction and classification issues, such spam detection, picture recognition, and sentiment analysis, are frequently addressed by it. Algorithms for supervised learning can generalise patterns and make precise assumptions about new data by using the labelled data.

Unsupervised learning, in contrast, works well in situations when there is only access to unlabeled data. Unsupervised learning looks for hidden patterns, structures, or relationships in the data rather than depending on predefined labels. For applications like clustering, anomaly detection, and dimensionality reduction, this method is frequently utilised. Without the use of explicit labels, unsupervised learning algorithms can reveal important insights and give a deeper knowledge of the underlying data.

A different paradigm that works well for problems involving environment-based decision-making is reinforcement learning. This method involves an agent interacting with the environment and getting feedback in the form of rewards or punishments. Through trial and error, the agent develops the ability to maximise a cumulative reward over time. Games, robots, and autonomous systems have all seen success using reinforcement learning, where the agent discovers the best course of action by observing and utilising its surroundings.

Which learning strategy is ”good” relies on a number of variables, including the availability of labelled or unlabeled data, the nature of the issue, and the task’s precise criteria. To get the best outcomes, it’s critical to thoroughly consider and choose the right learning strategy that fits the features of the situation.

Chapter 5

Implementation

In this chapter, we take a look at the detailed description of the methodologies, techniques, and tools used to develop and implement our machine learning model.

5.1 Dataset Collection

The dataset is taken from the Genomics of Drug Sensitivity in Cancer project.The Genomics of Drug Sensitivity in Cancer dataset is referred to as the GDSC dataset. For a variety of cancer cell lines, it is an extensive collection of genetic and treatment response data. The Genomics of therapeutic Sensitivity in Cancer project, which strives to comprehend the biological factors that influence therapeutic responsiveness in various cancer types, created the dataset.The genetic profiles of cancer cell lines are detailed in the GDSC dataset. It also contains information on how sensitive certain cell lines are to a wide range of anticancer medications.

Old dataset is based on release v1.0 from June 2012.New dataset was Release 8.4 in (July 2022).130612 more IC50s have been added to the GDSC databases. The new GDSC1 dataset contains 36 never-before-seen substances. The GDSC2 dataset now includes 99 chemicals and 160 more cell lines, with 80% more IC50s than version 1.0.

Table 5.1: Dataset Comparison

	GDSC1	GDSC 2
Age	from 2010 to 2015	NEW
Size of Cell Lines	697	969
Size of Compounds	111	297
IC50S	77367	243466

5.2 Data Preprocessing

In order to predict cancer cell sensitivity to medications based on genetic and chemical features, data preparation is a crucial stage in any machine learning effort. In order for machine learning models to successfully learn patterns and produce reliable predictions, it is essential that the quality and reliability of the data be improved.First of all import all the important libraries required.

```
from sklearn.impute import SimpleImputer
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
import seaborn as sns
from sklearn import datasets, metrics
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
```

The dataset primarly contains 10 features.

```
df =
   pd.read_csv('C:/Users/Hp/Desktop/final_year_project/project/drug
   data_csv.csv',
   usecols=['Cell_Line_Name','Drug_Name','Cosmic_ID','Drug_ID',
'TCGA_Classification','PSA','Tissue',
'Tissue_Sub-type','IS_Mutated','IC50'])
```

Since the Cell Line Name and Drug Name is uniquely linked to Cosmic ID and Drug ID respectively, we can remove those columns and reduce dimensions of our dataset.

```
columns_to_remove = ['Cell_Line_Name', 'Drug_Name']
df= df.drop(columns_to_remove, axis=1)
```

Using the simple imputer, we impute the missing values.

```
imputer=SimpleImputer(missing_values=np.nan,strategy='mean')
```

Now, after the general preprocessing of our data, we can now apply different machine learning algorithms and go through the accuracies of each and find the best model which can be used for preprocessing.

5.3 Algorithms

5.3.1 Random Forest

Random Forest is an ensemble learning algorithm used in machine learning for regression tasks. It is an extension of the decision tree algorithm and combines the predictions of multiple individual decision trees to make more accurate and robust predictions.In a Random Forest, a method known as "bootstrap aggregating" or "bagging" is used to assemble a group of decision trees. A randomly chosen subset of the original dataset is used to train each decision tree in the forest, with replacement (meaning some samples may be repeated). The sampling's randomness contributes to the introduction of variation among the trees.

Before applying random forest algorithm, some changes have to be made on the earlier preprocessed dataset.We need to convert all the categorical features consisting of different labels into numerical values.For that purpose, we make use of **One-Hot Encoding with multiple Categorical Values**.In this type of encoding, we only take those labels of categorical features which are repeated many number of times and apply One-Hot Encoding on it.Code for the same is:

```
pd.get_dummies(df,drop_first=True).shape
df.TCGA_Classification.value_counts().sort_values(ascending=False).head(10)
top_6=[x for x in df.TCGA_Classification.value_counts()
.sort_values(ascending=False).head(6).index]
for label in top_6:
   df[label]=np.where(df['TCGA_Classification']==label,1,0)
df[['TCGA_Classification']+top_6].head(10)
def one_hot_top_x(dff,variable,top_x_labels):
   for label in top_x_labels:
      dff[variable+'_'+label]=np.where(df[variable]==label, 1, 0)
one_hot_top_x(df,'TCGA_Classification',top_6)
```

After performing one hot encoding on every categorical features, remove those features, and move the target variable to last column.

```
columns_to_remove = ['TCGA_Classification','Tissue',
   'Tissue_Sub-type']
df = column_to_move = 'IC50'
df = df[[col for col in df.columns if col != column_to_move] +
   [column_to_move]]df.drop(columns_to_remove, axis=1)]
```

Apply Random Forest Algorithm:

```
X = df.drop('IC50')
y = df['IC50']
rf= RandomForestRegressor(n_estimators=100, random_state=42)
```

This model gave the value of R-Squared, Mean Square Error and Mean Absolute Error as **0.72 , 2.07, 1.07** respectively.

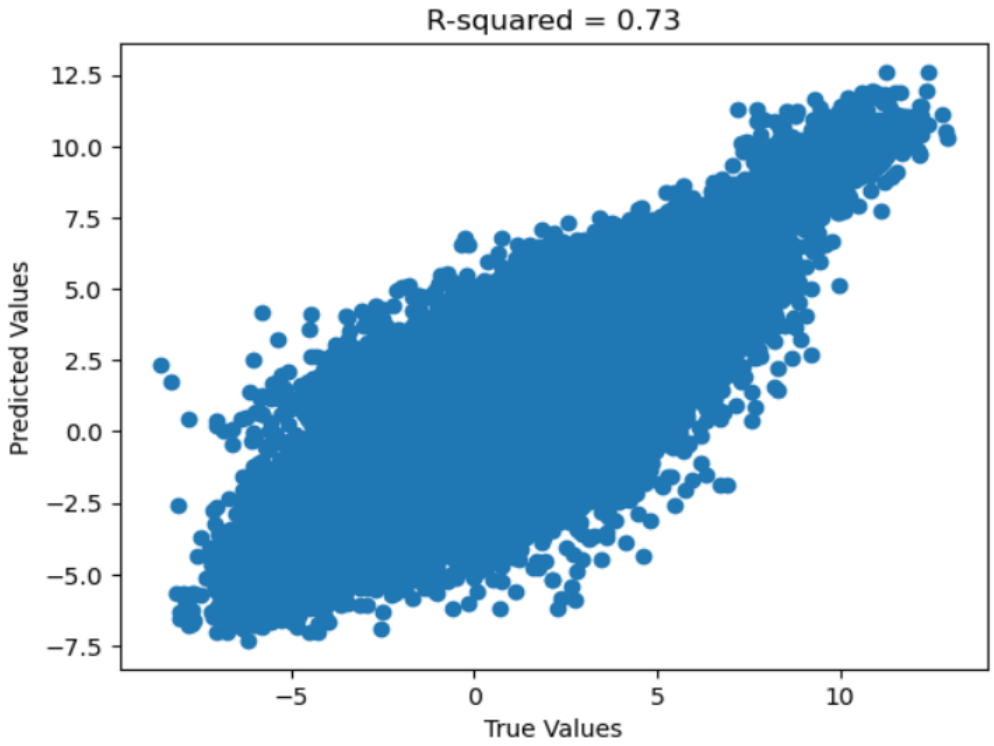

Figure 5.1: R-Squared Graph for Random Forest Algorithm

5.3.2 Support Vector Machine Algorithm

Frequently employed for classification and regression applications, Support Vector Machine (SVM) is a well-known supervised machine learning technique. It is especially useful for binary classification issues, but it can also be expanded to handle multi-class classification.

The fundamental goal of SVM is to identify the best hyperplane for dividing data points into distinct groups. This hyperplane seeks to maximise the margin in the case of binary classification, which is the separation between the hyperplane and the nearest data points for each class. The support vectors, or data points nearest to the hyperplane, are very important in constructing the decision boundary. Code for SVM is as follows:

```
X = df.drop('IC50')
y = df['IC50']
svm= SVR(kernel='rbf')
```

After providing the processed data to this model, it was unable to fit into the dataset. Possible reasons could be the increase in the number of columns from 10 to 55 after performing one-hot encoding on the categorical features which significantly impacted the performance of support vector machine (SVM) algorithm.In our case, the substantial increase in the number of columns could be causing the SVM algorithm to take too 6much time or struggle to fit the data. The high dimensionality of the data can lead to longer training times and potentially inefficient memory usage.

5.3.3 XGBoost (eXtreme Gradient Boosting)

is a strong and well-liked gradient boosting method that is employed for supervised machine learning tasks, particularly in the areas of predictive modelling and data analysis. It is renowned for its efficiency, effectiveness, and capacity for handling huge datasets.In a sequential fashion, XGBoost creates an ensemble of weak prediction models, often decision trees. These weak models are iteratively trained by the method to fix the errors made by the prior models.It updates the model parameters while minimising a defined loss function via gradient descent optimisation.

In this case, we will be using the dataset without one hot encoding.We will convert the categorical features into numerical one using Label Encoding Scheme.Code for the same is :

```
import time
from catboost import CatBoostRegressor, Pool
from sklearn import preprocessing
lbl=preprocessing.LabelEncoder()
X['TCGA_Classification']=lbl.fit_transform(X['TCGA_Classification'].astype(str))
X['Tissue']=lbl.fit_transform(X['Tissue'].astype(str))
X['Tissue_Sub-type']=lbl.fit_transform(X['Tissue_Sub-type'].astype(str))
start=time.time()
xgbr=xgb.XGBRegressor(n_estimators=1000,learning_rate=0.1)
```

This model gave the value of R-Squared, Mean Square Error and Mean Absolute Error as **0.91 , 1.3 , 0.84** respectively.

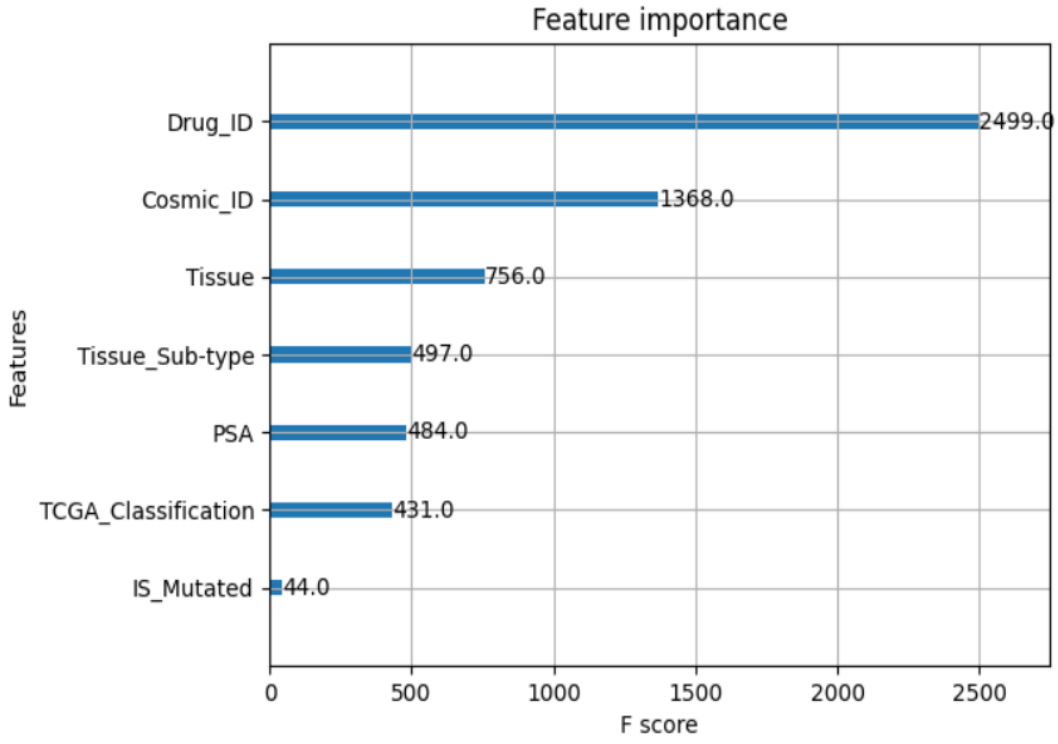

Figure 5.2: Importance that different features hold

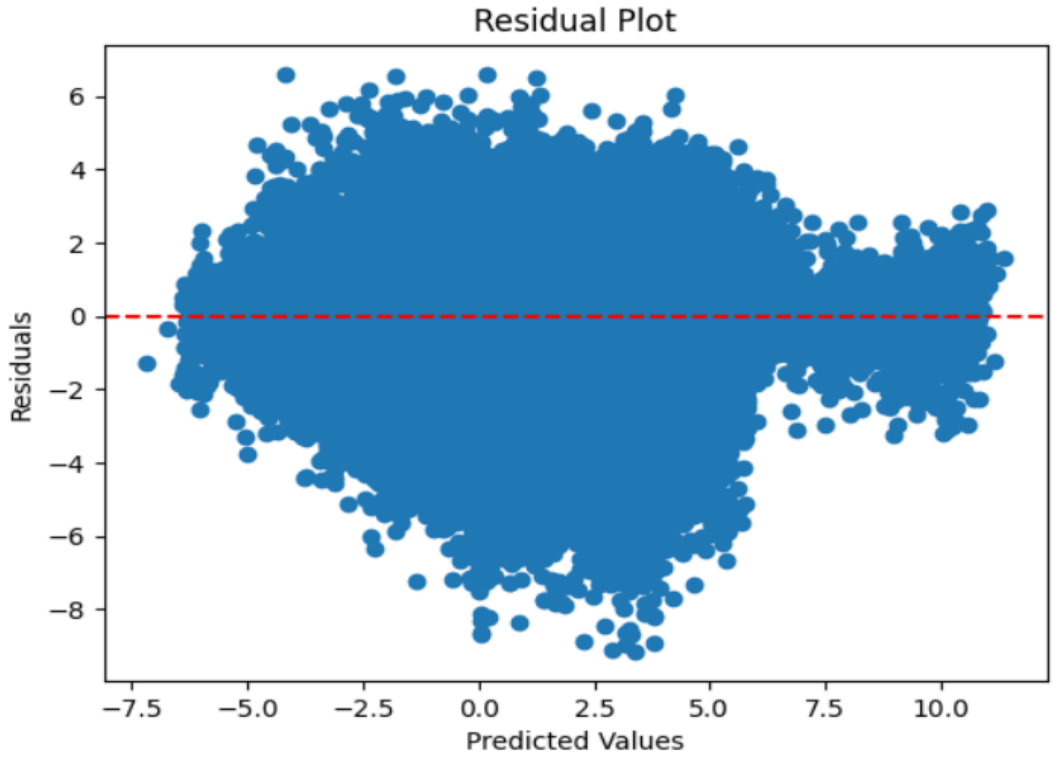

Figure 5.3: Residuals vs Predicted Values Plot

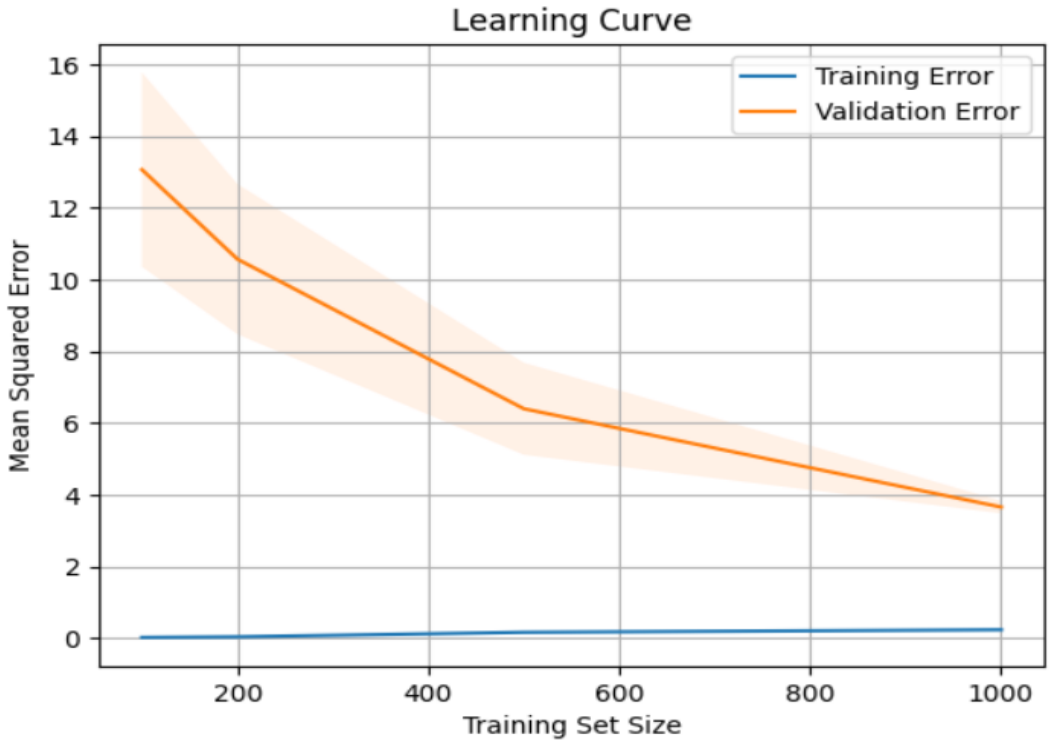

Figure 5.4: Learning curve graph for Xgboost

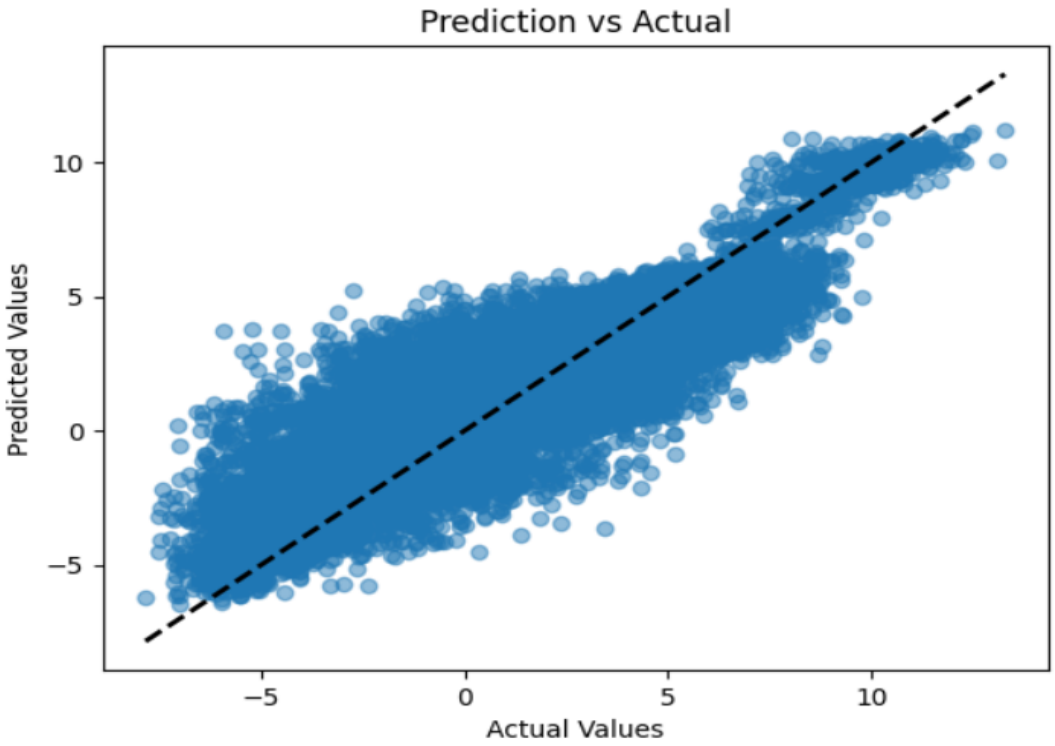

Figure 5.5: Predicted vs Actual value graph for Xgboost

5.3.4 CatBoost Algorithm

A gradient boosting method called CatBoost is created specifically to handle category information in machine learning problems. Its name, "Categorical Boosting," reflects the emphasis on handling categorical data well. The advantage of CatBoost is that categorical variables can be handled automatically without the requirement for manual encoding or preprocessing.

The CatBoost algorithm incorporates several techniques to handle categorical features efficiently. It uses an innovative algorithm called Ordered Boosting, which builds a series of decision trees by splitting the categorical variables in a way that preserves the natural ordering of the categories. This technique helps capture the valuable information present in categorical features.

In this model, we will not do any type of encoding on our dataset as it will be done internally by CatBoost.We just have to provide the categorical features of the dataset in the pool train and pool test.CatBoost efficiently handles categorical features without the need for explicit preprocessing processes like manual encoding by integrating target encoding, ordered boosting, and one-hot encoding.Below is the code for same :

```
import catboost
pool_train=Pool(X_train,y_train,cat_features=['TCGA_Classification','Tissue',
'Tissue_Sub-type'])
pool_test=Pool(X_test,cat_features=['TCGA_Classification',
'Tissue','Tissue_Sub-type'])
cbr=CatBoostRegressor(iterations=100)
cbr.fit(pool_train)
```

This model gave the value of R-Squared, Mean Square Error and Mean Absolute Error as **0.86 , 1.8 , 1.04** respectively.

5.3.5 LightGBM Algorithm

In comparison to existing gradient boosting techniques, LightGBM (Light Gradient Boosting Machine) attempts to give a faster training speed and higher efficiency. It was created by Microsoft and is frequently used in both commercial settings and machine learning contests.In this case, we will not perform any type pof encoding on our dataset , we will directly provide the data along with categorical features to this model.Code:

```
import lightgbm
obj_feat=list(X.loc[:,X.dtypes=='object'].columns.values)
for feature in obj_feat:
   X[feature]=pd.Series(X[feature],dtype="category")
lgbmr=lightgbm.LGBMRegressor()
```

This model gave the value of R-Squared, Mean Square Error and Mean Absolute Error as **0.86 , 2.00 , 1.08** respectively.

5.4 Deploy ML Model On Webpage

Deploying our machine learning model onto a webpage is very important part for usage and application of models. Flask allows us to do that. Flask is a microweb framework that integrates web application with any python program.After implementing the model, we create a pickle file at the end.Code for the same is :

```
pickle.dump(xgbr,open('cancer.pkl','wb'))
model=pickle.load(open('cancer.pkl','rb'))
```

Furthermore we create a flask server by providing it the instance of our model i.e. the pickle file.Code:

```
app = Flask(__name__, static_url_path='/static')
app.template_folder = 'templates'
model=pickle.load(open('cancer.pkl','rb'))

@app.route('/')
def index():
      return render_template('index.html')

@app.route('/predict', methods=['POST', 'GET'])
def predict():
   if request.method == 'POST':
      features = []
      for value in request.form.values():
```

```
            if value != 'submit': # Skip the 'submit' value
                features.append(int(value))
        final_features = [np.array(features)]
        final_features = pd.DataFrame(final_features,
            columns=['Cosmic_ID', 'Drug_ID', 'TCGA_Classification',
            'PSA', 'Tissue', 'Tissue_Sub-type', 'IS_Mutated'])
        prediction = model.predict(final_features)
        output = prediction[0]
    return render_template('index.html', prediction_text='Cancer cell
        IC50 value is {}'.format(output))


if __name__=='__main__':
        app.run()
```

As soon as we provide the input on our webpage, it calls the predict function of the respect model and provide the output, which indeed is displayed on the webpage with the help of flask server.

Code for predicting values on new data:

```
input_data = [[909976,1510,31,10,1,3,0]]
predictions = xgbr.predict(input_data)
```

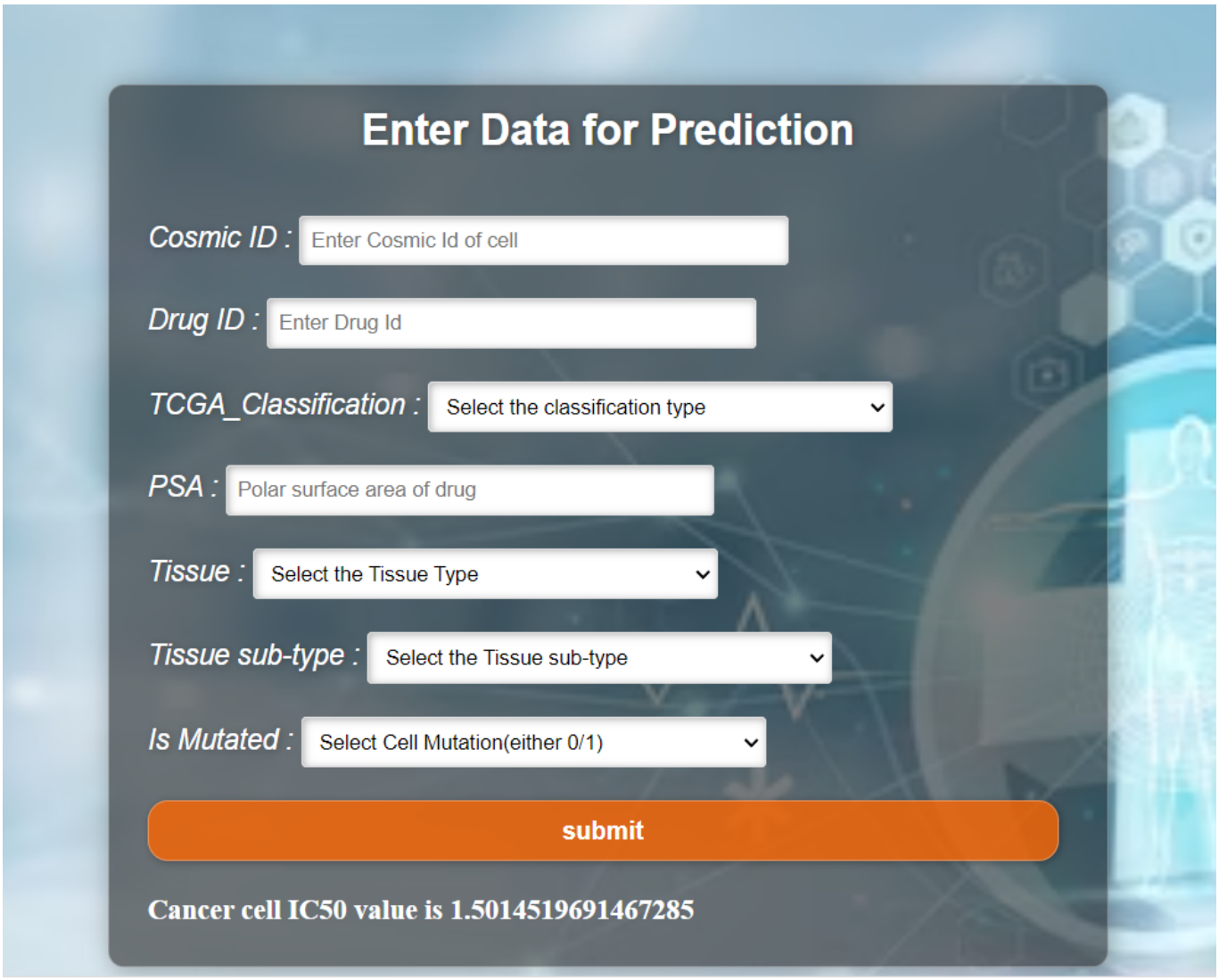

Figure 5.6: Demo

Chapter 6

Results

6.1 Evaluation Metrics

We use three popular metrics to evaluate the performance of all models.

1: R-Squared Value

The coefficient of determination, referred to as the R-squared (R2) value, is a statistical indicator that shows how much of the variance in the dependent variable can be accounted for by the independent variable in a regression model. It gauges how effectively the regression model accounts for the data that have been observed.The range of the R-squared value is 0 to 1.A value of 0 means that no variation in the dependent variable can be explained by the independent variable(s), whereas a value of 1 means that all variation in the dependent variable can be perfectly explained by the independent variable(s).

2: Mean Square Error

It is a frequently employed measurement of evaluation in regression models. In a regression issue, it calculates the average squared difference between the predicted values and the actual values.The average of the squared discrepancies between the projected values and the actual values for each data point in the dataset is used to calculate MSE mathematically. The following is the MSE formula:

The Mean Squared Error (MSE) is defined as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

where:

n is the number of samples

y is the actual value

ŷ is the predicted value

3: Mean Absolute Error

It is a statistic used to assess a regression model's effectiveness. Between actual and anticipated values, it calculates the average absolute difference between them.The formula for MAE is:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

MAE represents the average magnitude of errors, regardless of their direction. It provides a straightforward measure of the model's accuracy, where lower values indicate better performance.

6.2 Competing Models and Comparison between them

- **Random Forest**
- **Support Vector Machine**
- **XGBoost**
- **LightGBM**
- **CatBoost**

Table 6.1: Model Performance Comparison

Model	R-Squared Value	MSE	MAE
Random Forest	0.72	2.07	1.07
Support Vector Machine	-	-	-
XGBoost	0.91	1.3	0.84
CatBoost	0.86	1.8	1.04
LightGBM	0.86	2.0	1.08

In order to choose the optimal model based on these evaluation measures, we should take into account the particular project objectives. A higher R-squared value would suggest a better model if our main objectives are to maximise goodness-of-fit and your capacity to explain variance. On the other hand, lower values of MSE or MAE would indicate better model performance if our goal is to reduce prediction mistakes.

Comparing different model on basis of evaluation metrices :

1: R-Squared Value

In case of XGBoost, the value is highest which is 0.91, which indicates that all variation in the dependent variable can be perfectly explained by the independent variable(s).

2: MSE

XGBoost gave the lowest value which is 0.91, that signifies better model performance, as there is smaller errors between predictions and actual values.

3: MAE

In this case also, XGBoost was ahead of all other models.Lower MAE indicates better model performance and represents smaller errors.

After evaluating our models on all metrices, we came to the result that XGBoost performed best among different algorithms and previously used ones.

Chapter 7

Conclusion

In conclusion, the goal of this study was to create a machine learning model that would predict the sensitivity of cancer cells to various drugs based on their chemical and genomic composition. Data preparation, feature selection, model training, and evaluation were some of the phases that were included in the process.Information on Cosmic ID, Drug ID, TCGA Classification, Tissue, Tissue Sub-type, and IS Mutated was included in the dataset used for this experiment.

We used a variety of algorithms after doing data preprocessing, such as addressing missing values and categorising variables, and the XGBoostRegressor technique proved to be the most effective. Using the test dataset, we assessed the model's performance after training it on the training dataset.The resulting model performed remarkably well in terms of predicting IC50 values. The degree of accuracy attained shows how well the machine learning strategy works to predict the drug sensitivity to cancer cell lines using the given features.It is significant to remember that the dataset utilised and the particular population under study should both be taken into account when interpreting the model's performance. To evaluate the generalizability of the model, additional validation and testing on various datasets are required.

By identifying patient populations who are most likely to respond to a particular medicine, our model plays a critical role in optimising clinical trials. This can aid in choosing trial participants who will be a good fit, boost trial effectiveness, and raise the likelihood of a trial's success.Our model can be used to track the way cancer cells react to the chosen medications after the start of treatment. The model can offer insights into the efficacy of treatment and, if necessary, advise adjustments in therapy by routinely analysing genomic and chemical data from patient samples.

Bibliography

[1] Michael P Menden, Francesco Iorio, Mathew Garnett, Ultan McDermott, Cyril H Benes, Pedro J Ballester, and Julio Saez-Rodriguez. Machine learning prediction of cancer cell sensitivity to drugs based on genomic and chemical properties. *PLoS one*, 8(4):e61318, 2013.

[2] Delora Baptista, Pedro G Ferreira, and Miguel Rocha. Deep learning for drug response prediction in cancer. *Briefings in bioinformatics*, 22(1):360–379, 2021.

[3] George Adam, Ladislav Rampášek, Zhaleh Safikhani, Petr Smirnov, Benjamin Haibe-Kains, and Anna Goldenberg. Machine learning approaches to drug response prediction: challenges and recent progress. *NPJ precision oncology*, 4(1):19, 2020.

[4] Fei Zhang, Minghui Wang, Jianing Xi, Jianghong Yang, and Ao Li. A novel heterogeneous network-based method for drug response prediction in cancer cell lines. *Scientific reports*, 8(1):1–9, 2018.

[5] Zuoli Dong, Naiqian Zhang, Chun Li, Haiyun Wang, Yun Fang, Jun Wang, and Xiaoqi Zheng. Anticancer drug sensitivity prediction in cell lines from baseline gene expression through recursive feature selection. *BMC cancer*, 15(1):1–12, 2015.

[6] James C Costello, Laura M Heiser, Elisabeth Georgii, Mehmet Gönen, Michael P Menden, Nicholas J Wang, Mukesh Bansal, Muhammad Ammad-Ud-Din, Petteri Hintsanen, Suleiman A Khan, et al. A community effort to assess and improve drug sensitivity prediction algorithms. *Nature biotechnology*, 32(12):1202–1212, 2014.

[7] Jessica Vamathevan, Dominic Clark, Paul Czodrowski, Ian Dunham, Edgardo Ferran, George Lee, Bin Li, Anant Madabhushi, Parantu Shah, Michaela Spitzer, et al. Applications of machine learning in drug discovery and development. *Nature reviews Drug discovery*, 18(6):463–477, 2019.