FACULITY OF COMPUTERS
AND INFORMATICS

MENOFIA UNIVERSITY

Bioinformatics Department

Mechanisms of Action (MOA) Prediction Web Application

Mechanism of Action

Team Members :

- **Abdelmoneim Mohamed Rehab**
- **Ahmed Hosni Zaki Shaban**
- **Mahmoud Abd El-Hamid Abu Al-Atta**
- **Mahmoud Abu Khalil Al-Sayyed Gad**
- **Menna Allah Ahmed Taha El-Abd**
- **Merna Raafat Salem Mohamed Salem**
- **Moaz Alaa Abdel Fattah Saleh**

Under Supervision of

Dr. Hayam Mousa

2023/2024

Acknowledgement

We would like to express our deepest gratitude to all those who have supported and guided us throughout the duration of this project. This work would not have been possible without their invaluable assistance and encouragement.

First and foremost, we are deeply thankful to our academic advisor, Dr. Hayam, for their continuous guidance, insightful feedback, and unwavering support. Their expertise and patience have been instrumental in helping us navigate the complexities of this project.

We also extend our heartfelt thanks to our professors and lecturers at Menofia University, whose teachings and advice have significantly enriched our academic journey. Their dedication to education and research has inspired us to strive for excellence.

Special appreciation goes to the members of the Bioinformatics/Computer and information for providing the resources and facilities necessary for the successful completion of this project. Their support has been crucial in enabling us to achieve our research goals.

We are grateful to the Kaggle community and the organizers of the "Mechanisms of Action (MoA) Prediction" competition for providing the dataset and fostering an environment of collaborative

learning. The insights and resources available on the platform have been invaluable.

We would also like to acknowledge the assistance and encouragement from our peers and friends. Their constructive discussions, moral support, and camaraderie have made this journey more manageable and enjoyable.

Finally, we are profoundly grateful to our families for their unwavering support, understanding, and encouragement throughout our academic endeavors. Their belief in us has been a source of strength and motivation.

Abstract

This project focuses on developing a web application designed to predict the mechanism of action (MoA) of various compounds using data from the Kaggle competition "Mechanisms of Action (MoA) Prediction." The competition provided a comprehensive dataset that includes features representing gene expression and cell viability data for numerous compounds. By leveraging this dataset, our goal was to build and deploy machine learning models capable of accurately classifying the MoA of these compounds.

The web application serves as an interactive platform where users can input new data and receive real-time predictions about the MoA of the compounds. Our approach involved extensive data preprocessing, exploratory data analysis (EDA), and the application of advanced machine learning techniques. Among these, a neural network model was developed and fine-tuned, achieving a log loss of 0.017, which indicates high prediction accuracy and reliability.

The web application features an intuitive interface that allows users to input new data and obtain MoA predictions. Additionally, it offers educational resources on biological and pharmacological concepts to assist beginners in understanding the subject matter. This includes the classification of MoAs based on their therapeutic effects, which aids in comprehending the broader context of the predictions.

To illustrate the practical application of this technology, the project includes a case study on Type II Diabetes. This case study demonstrates how personalized medicine is determined by showcasing the relevant metabolic pathways involved in the disease. Furthermore, the application provides in-depth materials for users interested in exploring the topic more thoroughly.

Throughout this documentation, we detail the entire project lifecycle, beginning with background research on MoA and machine learning methodologies. We then delve into the literature review, highlighting previous work and existing models related to MoA prediction. The proposed solution section outlines the steps taken to preprocess the data, select and train models, and develop the web application. We also include a comprehensive analysis of the data, presenting various visualizations and insights gained from the EDA phase.

The testing phase is described in detail, showcasing the performance metrics used to evaluate our models and the results obtained. Finally, the documentation concludes with a discussion of the project outcomes, limitations encountered, and suggestions for future work to further improve the application and expand its capabilities.

This project not only demonstrates the application of machine learning in pharmacology but also provides a user-friendly tool that

can aid researchers in understanding the potential effects of new compounds, ultimately contributing to the advancement of drug discovery and development.

Table of contents

chapter 1: Introduction

chapter 1: introduction

1.1 Problem Statement

The prediction of the mechanism of action (MoA) of various compounds is a critical challenge in the field of pharmaceuticals and medical research. Understanding the MoA of a compound is essential for drug discovery and development, as it informs researchers about how a drug exerts its effects at the molecular level. Accurate prediction of MoA can significantly accelerate the drug development process, reduce costs, and improve the success rate of new drugs.

The Kaggle competition "Mechanisms of Action (MoA) Prediction" provides a dataset that includes features representing gene expression and cell viability data for numerous compounds. The challenge is to develop machine learning models that can classify the MoA based on these features. This project aims to address this challenge by developing a web application that leverages these machine learning models to predict the MoA of new compounds, providing a valuable tool for scientists and pharmaceutical companies in their drug discovery efforts.

1.2 Objectives

The main objectives of this project are:

- **Develop a Predictive Model:** Create a robust machine learning model that can accurately predict the mechanism of action of various compounds using the provided dataset.
- **Build a Web Application:** Develop a user-friendly web application that allows users to input new compound data and receive predictions on their MoA.
- **Facilitate Drug Discovery:** Provide a tool that helps scientists and pharmaceutical companies in understanding the potential effects of new compounds, thereby aiding in the drug discovery and development process.

1.3 Scope

The scope of this project encompasses several key areas:

- **Mechanism of Action (MoA):** Focus on predicting the MoA of compounds, which is vital for understanding drug interactions and effects at a molecular level.
- **Machine Learning:** Utilize machine learning techniques to build predictive models. This includes data preprocessing, feature selection, and model training.
- **Deep Learning:** Explore advanced deep learning models, particularly neural networks, to improve prediction accuracy.
- **Web Application Development:** Develop a web application using modern web frameworks to provide an interactive platform for users to input data and receive predictions.

- **Comprehensive Evaluation:** Evaluate the performance of the predictive models using appropriate metrics such as log loss.
- **Continuous Improvement:** Implement a framework for continuous model improvement based on new data and user feedback.

1.4 Methodology

The methodology of this project involves several key steps, each critical to the successful development of the predictive model and web application:

1. **Data Collection:**

- The primary data source is the dataset provided by the Kaggle competition "Mechanisms of Action (MoA) Prediction." This dataset includes features related to gene expression and cell viability.

2. **Data Preprocessing:**

- **Cleaning:** Handle missing values and outliers to ensure data quality.
- **Encoding:** Convert categorical data into numerical format using techniques such as one-hot encoding and label-encoding.

3. **Feature Selection and Extraction:**

- **Selection:** Identify the most relevant features that contribute to the prediction of the MoA.
- **Extraction:** Create new features that may enhance model performance.

4. **Model Training:**

- Develop various machine learning models, including neural networks, and train them on the preprocessed data.
- Use techniques like cross-validation to tune hyperparameters and prevent overfitting.

5. **Model Evaluation:**

- Evaluate the performance of the models using metrics such as log loss, accuracy, precision, recall, and F1 score.
- Select the best-performing model based on these evaluations.

6. **Model Deployment:**

- Integrate the trained model into a web application.
- Ensure the application is user-friendly and allows users to input new data for prediction.

7. **Continuous Improvement:**

- Monitor the performance of the model in real-world usage.
- Collect user feedback and new data to retrain and improve the model over time.

By following this methodology, we aim to develop a reliable and effective tool for predicting the mechanism of action of compounds, thereby contributing to the field of drug discovery and development.

1.4 Project Management

1.4.1 Team Members

- Abdelmoneim Mohamed Rehab
- Ahmed Hosni Zaki Shaban
- Mahmoud Abdel Hamid Abu Al-Atta
- Mahmoud Abu Khalil Al-Sayyed Gad
- Menna Allah Ahmed Taha El-Abd
- Merna Raafat Salem Mohamed Salem
- Moaz Alaa Abdel Fattah Saleh

1.4.2 Task Description

The project is divided into six main parts:

1. Data Collection and Understanding
2. Data Preprocessing
3. Model Development
4. Web Application Development
5. Testing and Evaluation
6. Deployment and Continuous Improvement

1.4.3 Task Definition

The team tasks can be summarized as follows:

- Literature Review: Mahmoud , Abdelmoneim
- Data Analysis: Hosny , Menna , Merna

- Implementation of Selected Machine / Deep Learning Model: Mahmoud, Abdelmoneim
- Testing And Evaluation: Moaz , Khalil , Abdelmoneim
- Web Application Development: Moaz , Khalil
- Documentation And Presentation: All Students

chapter 2:

Bioinformatics & drug discovery

Chapter 2: Bioinformatics & drug discovery

Introduction

In the modern era of drug discovery, the integration of bioinformatics has revolutionized the way researchers identify and develop new therapeutic agents. Bioinformatics, the interdisciplinary field that combines biology, computer science, and information technology, plays a crucial role in understanding complex biological data. This chapter delves into the synergy between bioinformatics and drug discovery, highlighting the mechanisms of action (MoA) of drugs and their significance in developing effective treatments.

The primary objective of this chapter is to explore how bioinformatics tools and techniques are applied to predict the MoA of various compounds, as exemplified by the Kaggle competition "Mechanisms of Action (MoA) Prediction." By leveraging gene expression and cell viability data, researchers can build predictive models that classify the MoA of compounds, ultimately aiding in the identification of potential drug candidates. This approach not only accelerates the drug discovery process but also enhances the precision of targeted therapies.

2.1 Bioinformatics

2.1.1 What is Bioinformatics ?

Bioinformatics is an interdisciplinary field that combines biology, computer science, and information technology to analyze and interpret complex biological data. It involves the development and application of computational algorithms and software tools to understand biological processes and relationships at a molecular level. In the context of drug discovery and development, bioinformatics plays a crucial role in analyzing vast amounts of genetic, genomic, and biochemical data to identify potential drug targets, predict the effects of new compounds, and understand their mechanisms of action (MOA).

2.1.2 Importance of Bioinformatics in Modern Drug Discovery

Bioinformatics has become an indispensable tool in modern drug discovery, offering numerous advantages that enhance the efficiency and effectiveness of the drug development process. Here's an overview of the key reasons why bioinformatics is so important in this field:

1.Understanding Mechanisms of Action (MoA)

Bioinformatics is crucial for elucidating the mechanisms of action of drugs, which involves understanding how a drug exerts its effects at the

molecular and cellular levels. This knowledge is essential for optimizing drug efficacy and safety.

- **Transcriptomics and Proteomics:** Analyzing changes in gene and protein expression in response to drug treatment.
- **Network Analysis:** Mapping the interactions between different biomolecules to understand the broader impact of the drug.

2. Facilitating Personalized Medicine

Bioinformatics enables the development of personalized medicine approaches by integrating patient-specific data (such as genomic and proteomic profiles) to tailor treatments. This ensures that patients receive the most effective therapies with the least side effects.

- **Biomarker Discovery:** Identifying biomarkers that can predict response to treatment.
- **Patient Stratification:** Classifying patients into subgroups based on their molecular profiles for targeted therapies.

3.Accelerating Target Identification and Validation

Bioinformatics tools help identify potential drug targets by analyzing genetic and molecular data. This involves pinpointing genes, proteins, or pathways that play a critical role in disease processes.

- **Genomic Analysis:** Identifying mutations or gene expressions associated with diseases.

- **Pathway Analysis:** Understanding the biological pathways involved in diseases and how they can be modulated by drugs.

2.1.3 Definition of Bioinformatics in the Context of the Project

In the context of this project, bioinformatics is utilized to predict the (MOA) of pharmaceutical compounds. By leveraging gene expression and cell viability data, bioinformatics tools and machine learning techniques can decipher how different compounds affect biological systems. This predictive capability is essential for drug discovery and development, as it helps researchers understand drug interactions, optimize therapeutic strategies, and advance personalized medicine. This project demonstrates the application of bioinformatics in creating a predictive model and a web application that facilitates MOA prediction, thereby enhancing our ability to develop effective and targeted treatments.

2.2 Drug discovery & MoA

2.2.1. Definition of Mechanism of Action (MoA)

The Mechanism of Action (MoA) of a drug refers to the specific biochemical interaction through which a drug substance produces its pharmacological effect. It involves understanding how a drug affects a biological target, such as a protein or receptor, at the molecular level. The MoA provides detailed information on the drug's target, the

biological pathway it influences, and the subsequent physiological effects.

2.2.2. How do we determine the MoAs of a new drug?

One approach is to treat a sample of human cells with the drug and then analyze the cellular responses with algorithms that search for similarity to known patterns in large genomic databases, such as libraries of gene expression or cell viability patterns of drugs with known MoAs.

Components of MoA:

- **Drug Target:** The biological molecule (e.g., protein, enzyme, receptor) that the drug interacts with.
- **Biochemical Pathways:** The series of chemical reactions and interactions that are influenced by the drug.
- **Physiological Effect:** The overall effect of the drug on the organism, including therapeutic and adverse effects.

Examples of MoAs

- **Enzyme Inhibitors:** Drugs that inhibit the activity of specific enzymes, such as kinase inhibitors used in cancer therapy.
- **Receptor Antagonists:** Drugs that block receptors and prevent biological signals from being transmitted, such as beta-blockers used in cardiovascular diseases.

- **Ion Channel Modulators:** Substances that alter the function of ion channels, used in the treatment of neurological disorders.
- **Signaling Pathway Inhibitors:** Drugs that interfere with specific cellular signaling pathways, such as those involved in inflammatory responses.
- **Gene Expression Modifiers:** Compounds that affect the expression levels of specific genes, used in various genetic disorders.

2.2.3. MoA classification based on their therapeutic effect

Regarding our data which have about 206 MoAs so we decided to classify them to make it more understandable and to get more information about possible drug names, diseases, and treatment so in the future we'll be able to include sequences for the model to train on it, And we classified them into ten categories based on their therapeutic effect :

1. **Anti-inflammatory**: Rheumatoid arthritis, inflammatory bowel diseases (Crohn's disease, ulcerative colitis), psoriasis, asthma.
   Examples: Ibuprofen, aspirin, prednisone, dexamethasone.
   MoAs(acetylcholine_receptor_antagonist, adenosine_receptor_agonist,adenosine_receptor_antagonist, adrenergic_receptor_antagonist, glucocorticoid_receptor_agonist ,histamine_receptor_antagonist,leukotriene_inhibitor, nitric_oxide_production_inhibitor,prostaglandin_inhibitor,

tnf_inhibitor)

2. **Antibiotic/Antiviral/Antifungal**:Bacterial infections (pneumonia, urinary tract infections), viral infections (influenza, HIV), fungal infections (candidiasis, aspergillosis).

   Examples: Penicillin, amoxicillin, azithromycin, oseltamivir, fluconazole.

   MoAs(bacterial_30s_ribosomal_subunit_inhibitor, bacterial_50s_ribosomal_subunit_inhibitor,bacterial_antifolate, bacterial_cell_wall_synthesis_inhibitor, bacterial_dna_gyrase_inhibitor,bacterial_dna_inhibitor, bacterial_membrane_integrity_inhibitor , hiv_inhibitor)

3. **Antineoplastic/Anticancer**: Various cancers (breast cancer, lung cancer, leukemia, etc.).

   Examples: Paclitaxel, doxorubicin, imatinib, tamoxifen.

   MoAs(alk_inhibitor , Aurora_kinase_inhibitor , Bcr-abl_inhibitor , Egfr_inhibitor , Flt3_inhibitor , Hcv_inhibitor , Mtor_inhibitor , Proteasome_inhibitor , Raf_inhibitor , Ras_gtpase_inhibitor , Topoisomerase_inhibitor , tyrosine_kinase_inhibitor)

4. **Neurological Agents**: Alzheimer's disease, Parkinson's disease, epilepsy, depression.

   Examples: Donepezil, levodopa, gabapentin, sertraline.

   MoAs(acetylcholine_receptor_agonist, acetylcholinesterase_inhibitor,cannabinoid_receptor_agonist, dopamine_receptor_agonist,dopamine_receptor_antagonist,

gamma_secretase_inhibitor,gaba_receptor_agonist, gaba_receptor_antagonist,histamine_receptor_agonist, serotonin_receptor_agonist , serotonin_receptor_antagonist , serotonin_reuptake_inhibitor)

5. **Cardiovascular Agents**: Hypertension, coronary artery disease, heart failure.

   Examples: Amlodipine, metoprolol, lisinopril, warfarin.

   MoAs(adenosine_receptor_agonist , adrenergic_receptor_agonist , angiotensin_receptor_antagonist , calcium_channel_blocker,cholesterol_inhibitor, mineralocorticoid_receptor_antagonist , nitric_oxide_donor)

6. **Endocrine Agents**: Diabetes mellitus, thyroid disorders, polycystic ovary syndrome.

   Examples: Metformin, levothyroxine, insulin, tamoxifen.

   MoAs(11-beta-hsd1_inhibitor,akt_inhibitor, aldehyde_dehydrogenase_inhibitor,aromatase_inhibitor, cortisol_synthesis_inhibitor,estrogen_receptor_agonist, estrogen_receptor_antagonist,faah_inhibitor, farnesyltransferase_inhibitor,glucocorticoid_receptor_agonist, hmgcr_inhibitor,insulin_secretagogue,insulin_sensitizer, progesterone_receptor_agonistt , vitamin_d_receptor_agonist)

7. **Immunomodulators**: Rheumatoid arthritis, multiple sclerosis, inflammatory bowel diseases.

   Examples: Methotrexate, infliximab, adalimumab, rituximab.

MoAs(akt_inhibitor,bcl_inhibitor,btk_inhibitor, calcineurin_inhibitor,histone_lysine_demethylase_inhibitor, histone_lysine_methyltransferase_inhibitor,hsp_inhibitor,igf-1_inhibitor , ikk_inhibitor , mtor_inhibitor)

8. **Analgesics/Anesthetics**: Pain management for various conditions (post-surgery, chronic pain, etc.).
   Examples: Lidocaine, morphine, tramadol, acetaminophen.
   MoAs(analgesic,anesthetic_-_local, Serotonin_receptor_agonist)
9. **Gastrointestinal Agents**: Gastroesophageal reflux disease (GERD), peptic ulcers, constipation.
   Examples: Omeprazole, ranitidine, loperamide, ondansetron.
   MoAs(acetylcholine_receptor_agonist, adrenergic_receptor_agonist,calcium_channel_blocker, chloride_channel_blocker,h2_receptor_antagonist, histamine_receptor_antagonist , leukotriene_receptor_antagonist ,muscarinic_receptor_antagonist,opioid_receptor_agonist, Opioid_receptor_antagonist)
10. **Respiratory Agents**: Asthma, chronic obstructive pulmonary disease (COPD), cystic fibrosis.
    Examples: Albuterol, fluticasone, montelukast, theophylline.
    MoAs(adenosine_receptor_agonist , adrenergic_receptor_agonist ,antihistamine,leukotriene_inhibitor, leukotriene_receptor_antagonist,serotonin_receptor_agonist ,

Serotonin_receptor_antagonist)

2.2.4 Diseases

From these MoAs and the classification we got, we managed to search and find the diseases that underlie each category to illustrate and dig deeper into the project idea not only from the ML side but also from Biological and pharmacology side and to provide those information for the students who gonna use our project for the purpose of learning We provided examples for more than one disease , possible drugs or treatments, but below I'll mention just mention some

1. **Anti-inflammatory diseases**

Most dangerous : Inflammatory bowel disease (IBD): While rheumatoid arthritis and asthma can significantly impact quality of life and lead to complications, severe forms of IBD, particularly Crohn's disease, can lead to life-threatening complications such as bowel perforation, severe bleeding, and increased risk of colorectal cancer.

Most widespread : Osteoarthritis: This degenerative joint disease is one of the most common forms of arthritis, affecting millions of people worldwide, especially older adults. It primarily affects weight-bearing joints such as the knees, hips, and spine.

examples where finding effective treatments or cures presents challenges: Systemic lupus erythematosus (SLE): Lupus is a complex autoimmune disease that can affect multiple organs and systems in the

body, leading to a wide range of symptoms such as joint pain, skin rashes, fatigue, and organ damage. Finding effective treatments for lupus is challenging due to its heterogeneity, unpredictable course, and poorly understood underlying mechanisms. While various medications, including corticosteroids, immunosuppressants, and biologic agents, can help manage symptoms and reduce disease activity, there is no cure for lupus, and treatment often involves a trial-and-error approach to find the most suitable therapy for each individual.

2. Antibiotic/Antiviral/Antifungal Diseases

Most dangerous : Sepsis: While not a specific disease itself, sepsis is a life-threatening condition that can occur as a complication of various infections, including bacterial infections treated with antibiotics. Sepsis occurs when the body's response to infection triggers widespread inflammation, leading to organ dysfunction and failure, and it has a high mortality rate if not promptly treated.

Most widespread : Urinary tract infections (UTIs): UTIs are among the most common bacterial infections worldwide, affecting millions of people each year. They can occur in individuals of all ages and genders and are typically caused by bacteria such as Escherichia coli.

examples where finding effective treatments or cures presents challenges: Multidrug-resistant tuberculosis (MDR-TB): Tuberculosis (TB) is caused by the bacterium Mycobacterium tuberculosis and remains a significant global health threat. MDR-TB occurs when the TB bacteria develop resistance to two or more first-line antibiotics, making

treatment significantly more challenging and expensive. The treatment of MDR-TB requires prolonged courses of second-line antibiotics that are often less effective, more toxic, and more costly than standard TB drugs. Additionally, treatment adherence is critical to prevent further drug resistance, but it can be challenging due to the long duration of therapy and potential side effects.

3. Antineoplastic/Anticancer

Most dangerous : Pancreatic cancer: Pancreatic cancer has one of the lowest survival rates among cancers, often due to late diagnosis and aggressive tumor behavior. It tends to spread rapidly and is frequently diagnosed at an advanced stage, making it difficult to treat effectively.

Most widespread : Lung cancer: Lung cancer is one of the most prevalent cancers globally and a leading cause of cancer-related deaths. It is closely associated with tobacco smoking but can also occur in non-smokers due to other factors such as exposure to secondhand smoke, air pollution.

examples where finding effective treatments or cures presents challenges: Glioblastoma multiforme (GBM): GBM is an aggressive form of brain cancer with a poor prognosis and limited treatment options. Despite extensive research efforts, including surgery, radiation therapy, and chemotherapy with drugs like temozolomide, the median survival for patients with GBM remains relatively short. The highly invasive nature of GBM tumors, along with their ability to evade

treatment and recur, presents significant challenges in developing effective therapies. Additionally, the blood-brain barrier limits the delivery of drugs to the brain, further complicating treatment strategies for GBM.

4. Neurological Agents

Most dangerous : Glioblastoma multiforme (GBM): This aggressive form of brain cancer is often considered one of the most deadly cancers. GBM tumors are highly infiltrative, making complete surgical removal challenging, and they tend to recur even after aggressive treatment with surgery, radiation therapy, and chemotherapy. The prognosis for GBM is generally poor, with a median survival of around 12-18 months after diagnosis.

Most widespread : Epilepsy: Epilepsy is a neurological disorder characterized by recurrent seizures and affects people of all ages worldwide. It is estimated that around 50 million people globally have epilepsy, making it one of the most common neurological conditions.

examples where finding effective treatments or cures presents challenges: Amyotrophic lateral sclerosis (ALS): ALS, also known as Lou Gehrig's disease, is a progressive neurodegenerative disorder that affects nerve cells in the brain and spinal cord, leading to muscle weakness, paralysis, and ultimately respiratory failure. Currently, there is no cure for ALS, and available treatments such as riluzole and edaravone provide only modest benefits in slowing disease progression. The

complex underlying mechanisms of ALS, coupled with its heterogeneity and rapid progression, pose significant challenges in developing effective therapies. Additionally, the lack of reliable biomarkers for ALS diagnosis and monitoring further complicates clinical trials and drug development efforts.

2.2.2 Mechanisms of Action (MoAs) in the Context of the LISH-MOA Competition

The information about drugs that do not bind to specific receptors and instead produce their therapeutic effects through chemical or physical interactions may not be directly related to the LISH-MOA competition, as the competition focuses on predicting the mechanisms of action (MoA) of drugs based on their interactions with specific biological targets.

They use machine learning techniques to predict the MoA of drugs based on these features. The focus is on understanding how drugs interact with biological targets, signaling pathways, and cellular processes to produce their effects.

2.2.2.1 Example from the Data:

Let's say there is a drug in the dataset that is labeled as an "adenosine receptor agonist." This indicates that the drug binds to adenosine receptors in the body, activating them and triggering downstream

cellular responses. The MoA label provides specific information about how the drug interacts with biological targets, allowing researchers to understand its mechanism of action.

2.2.2.2 Getting the suitable drugs

When you have a mechanism of action (MoA) associated with multiple drugs, determining the most suitable drug for a particular purpose involves considering various factors. Below are the key steps in this process, using predictive models developed in the LISH-MOA competition:

1. **Analyze Gene Expression and Cell Viability Profiles:**
    - Use predictive models to evaluate the gene expression and cell viability profiles of various drugs.
    - Identify potential candidates that target specific biological pathways, such as the PI3K/AKT/mTOR pathway in cancer treatment.
2. **Evaluate Pharmacokinetics and Safety Profiles:**
    - Assess the pharmacokinetics of the identified drugs to determine how they are absorbed, distributed, metabolized, and excreted in the body.
    - Review the safety profiles of the drugs to ensure they do not produce significant adverse effects.
3. **Investigate Preclinical Efficacy:**
    - Conduct preclinical studies to evaluate the antitumor activity of

the drugs in relevant cancer models.

- For example, test the drugs in breast cancer models to observe their effectiveness in reducing tumor growth.

4. **Prioritize Drug Candidates:**
   - Based on the combined data from predictive models, pharmacokinetics, safety profiles, and preclinical efficacy, prioritize the most promising drug candidates.
   - In this example, Drug A and Drug B are identified as potential therapies for breast cancer patients with dysregulated PI3K/AKT/mTOR signaling.
5. **Plan for Clinical Evaluation:**
   - Prepare for further development and clinical trials to evaluate the efficacy and safety of the prioritized drugs in human patients.
   - Design clinical studies to test the drugs in specific patient populations, ensuring that they target the intended biological pathways effectively.
6. **Consider Additional Factors:**
   - Efficacy: Ensure the drug effectively targets the disease or biological pathway.
   - Safety: Confirm that the drug has a favorable safety profile with minimal adverse effects.
   - Pharmacokinetics: Verify that the drug has suitable pharmacokinetic properties for the intended use.

- Clinical Relevance: Assess the drug's relevance and potential benefits in clinical settings, considering patient demographics and disease characteristics.

2.2.2.3 Deciding the MoA

Determining the best mechanism of action (MoA) for a given drug involves leveraging the available data and employing appropriate evaluation metrics.

1- Data Exploration and Analysis:

- Begin by thoroughly exploring the competition dataset, which likely includes features such as cell viability assays, gene expression profiles, and drug treatment conditions.

Analyze the distribution of MoA labels in the training dataset to understand the prevalence of different mechanisms and potential class imbalances.

2- Feature Engineering:

Extract meaningful features from the raw data, including engineered features derived from cell viability assays and gene expression data.

Utilize techniques such as dimensionality reduction (e.g., PCA) and feature selection to identify the most informative features for predicting MoA.

3- Model Selection and Training:Experiment with various machine

learning algorithms such as random forests, gradient boosting machines, or neural networks.

- Train multiple models using different combinations of features, hyperparameters, and training strategies.
- Evaluate model performance using appropriate evaluation metrics such as accuracy, precision, recall, F1-score, or area under the ROC curve (AUC-ROC).

4- Cross-Validation and Validation Strategy:

- Perform cross-validation to assess the generalization performance of the models and ensure robustness to variations in the dataset.
- Consider utilizing techniques such as stratified sampling to preserve class distributions during cross-validation.

5- Ensemble Techniques:

- Explore ensemble techniques such as stacking, blending, or bagging to combine predictions from multiple base models and improve overall performance.
- Ensemble methods can help mitigate overfitting and capture complementary patterns in the data.

6- Hyperparameter Tuning:

- Conduct hyperparameter tuning using techniques such as

grid search or random search to optimize model performance.

- Tune parameters such as learning rates, regularization strengths, tree depths, or neural network architectures to improve model generalization.

7- Validation on Test Set:

- Assess the performance of the trained models on a held-out test set provided by the competition organizers.
- Submit predictions generated by the best-performing model(s) to the competition platform for evaluation against the ground truth MoA labels.

8- Analysis of Results:

- Analyze the performance of different models and identify factors contributing to their success or failure.
- Examine misclassifications and explore potential reasons for model errors to gain insights into the challenges of prediction

9- Post-Competition Analysis:

- Continue refining and optimizing models even after the competition ends based on post-competition analysis and feedback.

- Collaborate with other participants and researchers to share insights, methodologies, and best practices for predicting MoA.

2.2.3 Purpose of the project

- Advance the field of drug discovery and pharmacology. Predicting the mechanism of action (MoA) of drugs is crucial in understanding how they interact with biological systems, which is fundamental for drug development, personalized medicine, and understanding disease mechanisms.
- Participants in this competition are likely tasked with developing machine learning models or data analysis techniques to accurately predict the MoA of drugs based on data from various biological assays. By accurately predicting the MoA, researchers can potentially identify new therapeutic targets, optimize drug combinations, and improve the efficiency of drug development processes.

Cell viability and gene expression data serve as crucial features for predicting the mechanism of action (MoA) of drugs.

Cell Viability Data: Cell viability assays measure the ability of cells to survive, proliferate, or undergo apoptosis (cell death) in response to various stimuli, including drug treatments. By assessing how different drugs affect cell viability under different conditions, researchers can gain insights into

the potential MoA of those drugs. Changes in cell viability may indicate specific cellular responses, such as cytotoxicity or growth inhibition, which can help classify drugs into different categories based on their effects on cells.

Gene Expression Data: Gene expression refers to the process by which information from a gene is used to synthesize functional gene products, such as proteins. Gene expression profiling involves measuring the levels of gene transcripts (mRNA) or proteins in cells or tissues under different conditions, including drug treatments. Changes in gene expression patterns can provide valuable information about the biological pathways and mechanisms affected by drug treatments. By analyzing gene expression data, researchers can identify key genes or pathways associated with specific MoAs, which can then be used as features for predictive modeling.

In summary, cell viability and gene expression data are used in the LISH-MOA competition to capture the biological responses of cells to drug treatments. These data types provide valuable information for building predictive models that can accurately classify drugs based on their MoA, ultimately contributing to the development of more effective and targeted therapies.

In chapter 5 you'll find a detailed case study which illustrate how the MoAs work on different metabolic pathway and we've choose type || diabetes for this case study for many reasons that'll be highlighted in the chapter

chapter 3: Background

chapter 3: Background

In this chapter, we will provide a comprehensive overview of the key concepts and foundational knowledge necessary for understanding and executing this project. We will cover the following areas:

- **Mechanism of Action (MoA):** We will delve into what MoA is, its significance in drug development, and provide details about the dataset used in the Kaggle competition "Mechanisms of Action (MoA) Prediction."
- **Data Science and Machine Learning:** We will introduce the essential concepts of data science and machine learning relevant to this project, including the techniques and methodologies employed.

3.1 Mechanism of Action

3.1.1 Introduction About MOA

What is the Mechanism of Action (MoA) of a drug? And why is it important?

In the past, scientists derived drugs from natural products or were inspired by traditional remedies.

Very common drugs, such as paracetamol, known in the US as acetaminophen, were put into clinical use decades before the biological mechanisms driving their pharmacological activities were understood.

Today, with the advent of more powerful technologies, drug discovery has changed from the serendipitous approaches of the past to a more targeted model based on an understanding of the underlying biological mechanism of a disease. In this new framework, scientists seek to identify a protein target associated with a disease and develop a molecule that can modulate that protein target.

As a shorthand to describe the biological activity of a given molecule, scientists assign a label referred to as mechanism-of-action or MoA for short.

How do we determine the MoAs of a new drug?

One approach is to treat a sample of human cells with the drug and then analyze the cellular

responses with algorithms that search for similarity to known patterns in large genomic databases,

such as libraries of gene expression or cell viability patterns of drugs with known MoAs.

3.2 Competition

what this competition do?

In this competition, you will have access to a unique dataset that combines gene expression and cell viability data.The data is based on a new technology that measures simultaneously (within the same samples) human cells' responses to drugs in a pool of 100 different cell types (thus solving the problem of identifying ex-ante, which cell types are better suited for a given drug).In addition, you will have access to MoA annotations for more than 5,000 drugs in this dataset.

As is customary, the dataset has been split into testing and training subsets. Hence, your task is to use the training dataset to develop an algorithm that automatically labels each case in the test set as one or more MoA classes. Note that since drugs can have multiple MoA annotations, the task is formally multi-label classification problem

How to evaluate the accuracy of a solution?

Based on the MoA annotations, the accuracy of solutions will be evaluated on the average value of the logarithmic loss function applied to each drug-MoA annotation pair.If successful, you'll help to develop an algorithm to predict a compound's MoA given its cellular signature, thus helping scientists advance the drug discovery process.

Evaluation

For every sig_id you will be predicting the probability that the sample had a positive response for each <MoA> target. For N sig_id rows and M <MoA> targets, you will be making N×M predictions. Submissions are scored by the log loss:

$$\text{score} = -\frac{1}{M}\sum_{m=1}^{M}\frac{1}{N}\sum_{i=1}^{N}\left[y_{i,m}\log(\hat{y}_{i,m}) + (1-y_{i,m})\log(1-\hat{y}_{i,m})\right]$$

where:

- N is the number of `sig_id` observations in the test data ($i = 1, \ldots, N$)
- M is the number of scored MoA targets ($m = 1, \ldots, M$)
- $\hat{y}_{i,m}$ is the predicted probability of a positive `MoA` response for a `sig_id`
- $y_{i,m}$ is the ground truth, 1 for a positive response, 0 otherwise
- $log()$ is the natural (base e) logarithm

Note: the actual submitted predicted probabilities are replaced with $max(min(p, 1-10^{-15}), 10^{-15})$. A smaller log loss is better.

Data in the Competition

The Kaggle competition "Mechanisms of Action (MoA) Prediction" provides a rich dataset that is essential for building predictive models. The dataset includes:

- **Features:** The dataset includes gene expression data and cell viability data. Specifically, it contains:
    - **Gene Expression Features:** These are denoted by g- followed by a number (e.g., g-0, g-1, etc.), and they represent the expression levels of various genes.
    - **Cell Viability Features:** These are denoted by c- followed by a number (e.g., c-0, c-1, etc.), and they represent the viability of cells under different experimental conditions.
- **Training Data (train_features.csv):** Contains the feature data for training the models. This file includes columns for each gene expression and cell viability feature, as well as an identifier for each sample.
- **Target Data (train_targets_scored.csv):** Contains the MoA labels for the compounds in the training set. This file includes binary indicators for each mechanism of action, where 1 indicates the presence of a specific MoA and 0 indicates its absence.

- **Test Data (test_features.csv):** Contains the feature data for which predictions need to be made. The structure is similar to train_features.csv but without the target labels.
- **Additional Metadata:**
    - **train_targets_nonscored.csv:** Contains additional targets that are not scored in the competition but can provide extra context for understanding the data.
    - **sample_submission.csv:** A sample submission file showing the format required for submitting predictions to the competition.

The competition dataset facilitates the development of machine learning models by providing a structured format of input features and corresponding MoA labels. This structured approach enables the training of robust predictive models.

3.2. Data Science and Machine Learning

Definition of Data Science

Data science is an interdisciplinary field that uses scientific methods, processes, algorithms, and systems to extract knowledge and insights from structured and unstructured data. In the context of this project, data science involves leveraging various techniques from statistics, machine learning, and data analysis to understand and predict the mechanism of action (MoA) of different compounds. By analyzing gene expression and cell viability

data, data science helps in identifying patterns and relationships that can predict how a compound will interact with its molecular targets.

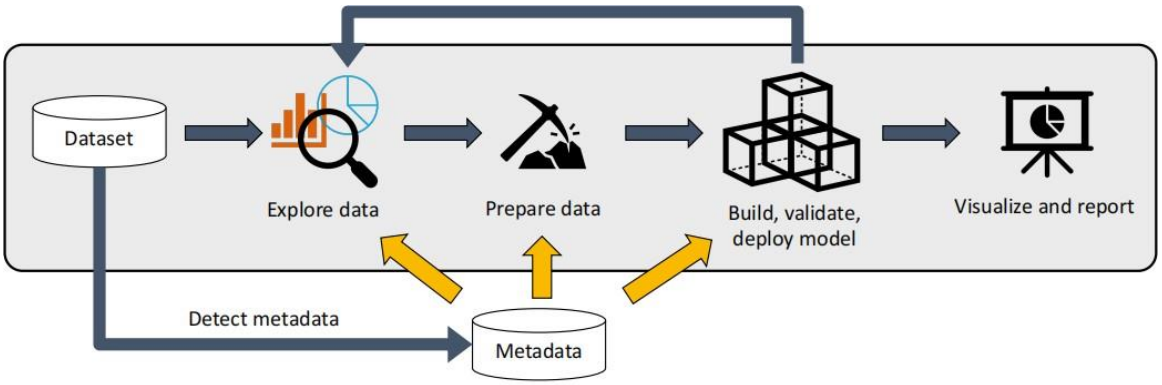


Figure 1.1: A typical workflow for data science projects.

Metadata are useful information throughout a data science lifecycle. Figure 1.1

demonstrates a typical workflow for data science projects. Given a dataset collected

from for example a data lake, a data scientist first explores it with data exploration

tools, such as Power BI and Tableau for tabular data, to gain an understanding of or

spot errors in the dataset. With the knowledge of data characteristics and quality, the

data scientist further prepares the dataset, aligning it to the models they want to build

for particular applications. After that, they build and validate models based on the pre

pared data. With new data problems discovered during this phase, the data scientist may

return to data exploration to gain more knowledge about the data, or data preparation

to further polish the data, before serving the prepared data to the models again. All

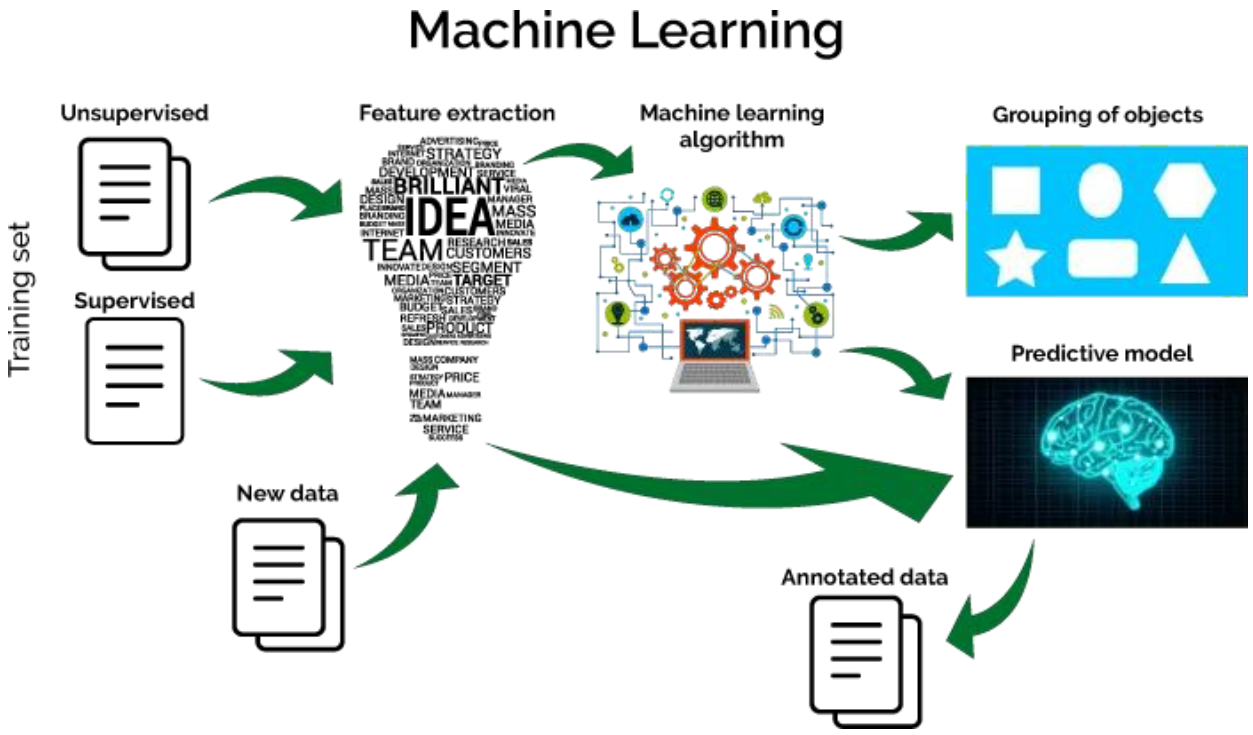
above operations require comprehending the data, which can be facilitated with various metadata.

Machine Learning

Machine learning (ML) is defined as a discipline of artificial intelligence (AI) that provides machines the ability to automatically learn from data and past experiences to identify patterns and make predictions with minimal human intervention. This article explains the fundamentals of machine learning, its types, and the top five applications. It also shares the top 10 machine learning trends.

How does machine learning work?

Machine learning algorithms are molded on a training dataset to create a model. As new input data is introduced to the trained ML algorithm, it uses the developed model to make a prediction.



Machine Learning Workflow

The machine learning workflow is a structured process that guides the development of predictive models. Here is an overview of the workflow as it relates to this project:

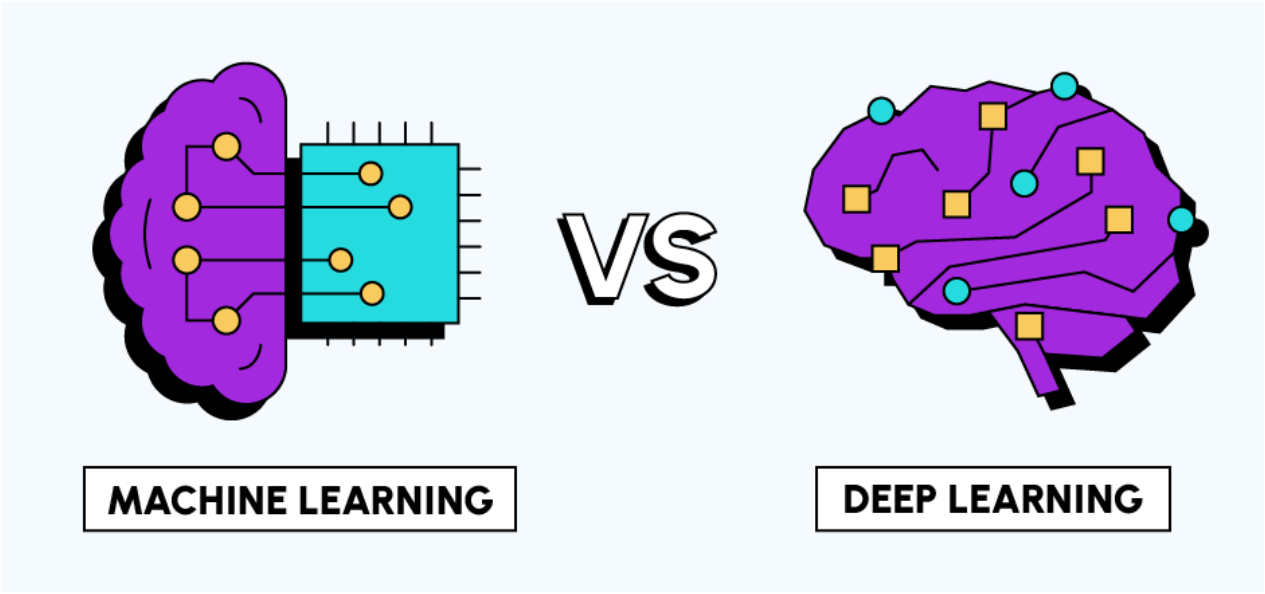
1. **Data Collection:**
    - Gather data from the Kaggle competition dataset, which includes gene expression and cell viability data.

2. **Data Preprocessing:**
    - **Data Cleaning:** Remove or correct any errors or inconsistencies in the data.
    - **Encoding Data:** Convert categorical data into numerical format using techniques such as one-hot encoding.
    - **Feature Scaling:** Standardize or normalize features to ensure they contribute equally to the model.
3. **Exploratory Data Analysis (EDA):**
    - Visualize and summarize the main characteristics of the dataset.
    - Identify patterns, correlations, and potential outliers.
4. **Feature Engineering:**
    - Create new features based on domain knowledge to improve model performance.
    - Select the most relevant features for the model.
5. **Model Training:**
    - Split the data into training and validation sets.
    - Train multiple machine learning models, including neural networks, to learn from the training data.
6. **Model Evaluation:**
    - Evaluate model performance using metrics such as log loss, accuracy, precision, recall, and F1 score.
    - Perform cross-validation to ensure the model generalizes well to unseen data.

7. **Model Optimization:**
    - Tune hyperparameters to optimize model performance.
    - Use techniques such as grid search or random search for hyperparameter tuning.
8. **Model Deployment:**
    - Deploy the trained model in a web application.
    - Ensure the model can make predictions on new data inputs provided by users.
9. **Continuous Improvement:**
    - Monitor model performance over time.
    - Update the model with new data and re-train as necessary to maintain accuracy.

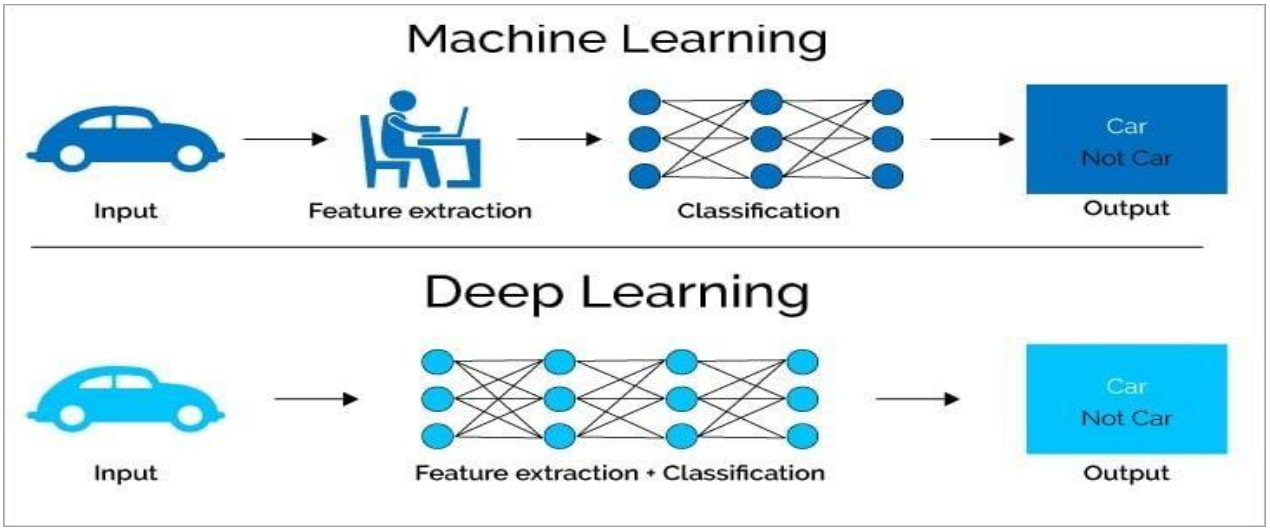
Deep Learning

Deep learning is a subset of machine learning, which in turn is a subset of artificial intelligence (AI). It involves the use of neural networks with many layers (hence "deep") to model complex patterns in data. These layers are composed of nodes, or artificial neurons, that mimic the way human brains process information. Deep learning algorithms automatically learn to recognize intricate structures and features in data through a process called training, which typically involves vast amounts of data and substantial computational power.



First and foremost, while traditional Machine Learning algorithms have a rather simple structure, such as linear regression or a decision tree, Deep Learning is based on an artificial neural network. This multi-layered ANN is, like a human brain, complex and intertwined.

Secondly, Deep Learning algorithms require much less human intervention. Remember the Tesla example? If the STOP sign image recognition was a more traditional machine learning algorithm, a software engineer would manually choose features and a classifier to sort images, check whether the output is as required, and adjust the algorithm if this is not the case. As a deep learning algorithm, however, the features are extracted automatically, and the algorithm learns from its own errors (see image below).



Thirdly, Deep Learning requires much more data than a traditional Machine Learning algorithm to function properly. Machine Learning works with a thousand data points, deep learning oftentimes only with millions. Due to the complex multi-layer structure, a deep learning system needs a large dataset to eliminate fluctuations and make high-quality interpretations.

What is Multi-Label Classification?

To understand multi-label classification, firstly we will understand what is meant by multi-label, and find the difference between multi-label and binary-label.

Multi-label vs. single-label is the matter of how many classes an object or example can belong to. In neural networks, when single-label is required, we use a single softmax layer as the last layer, learning a single probability

distribution that ranges over all classes. In the case where multi-label classification is needed, we use multiple sigmoids on the last layer and thus learn a separate distribution for each class.

In certain problems, each input can have multiple, or even none, of the designated output classes. In these cases, we go for the multi-label classification problem approach.

For example, If we are building a model which predicts all the clothing articles a person is wearing, we can use a multi-label classification model since there can be more than one possible option at once.

Chapter 3:

Literature Review

Chapter 3: Literature Review

A literature review is an essential part of any research project as it provides a comprehensive overview of existing knowledge and research on a particular topic. It helps researchers understand the current state of the field, identify gaps in existing research, and build upon the work of others. By reviewing the literature, researchers can ensure that their work is grounded in the existing body of knowledge and can contribute meaningfully to the field. In addition, a literature review helps in refining the research questions, guiding the methodology, and contextualizing the findings.

In the context of this project, the literature review aims to:

- **Understand Existing Methods:** Review current methods and approaches used for predicting the mechanism of action (MoA) of compounds, with a focus on machine learning techniques.
- **Identify Best Practices:** Identify best practices in data preprocessing, feature selection, feature extraction, and model development to ensure robust and accurate predictions.
- **Explore Web Technologies:** Explore web technologies and frameworks that can be used to develop a user-friendly web application for MoA prediction.

Data Preprocessing

Data preprocessing is a crucial step in the machine learning workflow. It involves preparing and transforming raw data into a format that can be effectively used by machine learning models. Proper preprocessing ensures that the data is clean, consistent, and suitable for analysis, thereby improving the performance and accuracy of the models.

In this project, data preprocessing focuses on encoding categorical data. Encoding is the process of converting categorical data into numerical format so that machine learning algorithms can process it. There are various types of encoding techniques, including one-hot encoding and label encoding.

Types of Encoding

1. **One-Hot Encoding:** This technique converts categorical variables into a series of binary columns, each representing a category. While it is useful for categorical variables with no ordinal relationship, it can lead to a large number of columns if the categorical variable has many levels.
2. **Label Encoding:** This technique assigns a unique integer to each category in a categorical variable. It is simple and effective for ordinal data where the order of the categories matters. However, it can introduce ordinal relationships where none exist, which may not be suitable for all types of categorical data.

In this project, we used label encoding to handle categorical data. The columns we handled using label encoding are cp_dose, cp_type, and cp_time.

Columns and Encoding Example

1. **cp_dose:** This column indicates the dose of the compound. It has two categories: 'D1' and 'D2'.
2. **cp_type:** This column indicates the type of the compound. It has two categories: 'trt_cp' (treatment compound) and 'ctl_vehicle' (control compound).
3. **cp_time:** This column indicates the duration of the treatment. It has three categories: 24, 48, and 72 hours.

Here is an example of how these columns are encoded using label encoding:

Sample ID	cp_dose	cp_type	cp_time
1	D1	trt_cp	24
2	D2	ctl_vehicle	48
3	D1	trt_cp	72

After label encoding, the table would look like this:

Sample ID	cp_dose	cp_type	cp_time

1	0	1	0
2	1	0	1
3	0	1	2

In this example:

- cp_dose: 'D1' is encoded as 0 and 'D2' as 1.
- cp_type: 'trt_cp' is encoded as 1 and 'ctl_vehicle' as 0.
- cp_time: 24 hours is encoded as 0, 48 hours as 1, and 72 hours as 2.

By converting categorical data into numerical format using label encoding, we ensure that the machine learning algorithms can process the data effectively, leading to better model performance and more accurate predictions.

Feature Selection and Feature Extraction

Feature selection and feature extraction are crucial steps in the machine learning process that aim to improve model performance by focusing on the most relevant data. Both techniques help reduce the dimensionality of the dataset, leading to more efficient and effective models.

Feature Selection

Feature selection involves selecting a subset of the most relevant features (variables, predictors) from the original dataset. The goal is to improve the

performance of the model by reducing overfitting, improving accuracy, and decreasing computational cost.

Types of Feature Selection:

1. **Filter Methods:** Select features based on statistical measures. Examples include variance thresholding, correlation coefficients, and chi-square tests.
2. **Wrapper Methods:** Use a predictive model to evaluate the combination of features and select the best performing subset. Examples include recursive feature elimination (RFE) and forward selection.
3. **Embedded Methods:** Perform feature selection during the model training process. Examples include Lasso (L1 regularization) and decision trees.

In this project, we used the **Variance Threshold** method for feature selection. The Variance Threshold removes features that have low variance, assuming that features with higher variance are more likely to be informative.

- **Gene Features:** Features representing genetic data.
- **Cell Features:** Features representing cell data.

We applied the Variance Threshold on both gene features and cell features. Only features with a variance greater than 1 were retained for model training. Below are the results of the variance threshold application.

Features with Variance Greater Than 1:

Index(['g-0', 'g-2', 'g-4', 'g-5', 'g-7', 'g-8', 'g-9', 'g-10', 'g-11', 'g-12',

... 'g-761', 'g-762', 'g-763', 'g-764', 'g-765', 'g-766', 'g-767', 'g-769',

'g-770', 'g-771'], dtype='object', length=585)

Index(['c-0', 'c-1', 'c-2', 'c-3', 'c-4', 'c-5', 'c-6', 'c-7', 'c-8', 'c-9',

'c-10', 'c-11', 'c-12', 'c-13', 'c-14', 'c-15', 'c-16', 'c-17', 'c-18',

'c-19', 'c-20', 'c-21', 'c-22', 'c-23', 'c-24', 'c-25', 'c-26', 'c-27',

'c-28', 'c-29', 'c-30', 'c-31', 'c-32', 'c-33', 'c-34', 'c-35', 'c-36',

'c-38', 'c-39', 'c-40', 'c-41', 'c-42', 'c-43', 'c-44', 'c-45', 'c-46',

'c-47', 'c-48', 'c-49', 'c-50', 'c-51', 'c-52', 'c-53', 'c-54', 'c-55',

'c-56', 'c-57', 'c-59', 'c-60', 'c-61', 'c-62', 'c-63', 'c-64', 'c-65',

'c-66', 'c-67', 'c-68', 'c-69', 'c-70', 'c-71', 'c-72', 'c-73', 'c-75',

'c-76', 'c-77', 'c-78', 'c-79', 'c-80', 'c-81', 'c-82', 'c-83', 'c-84',

'c-85', 'c-86', 'c-87', 'c-88', 'c-89', 'c-90', 'c-91', 'c-92', 'c-93',

'c-94', 'c-95', 'c-96', 'c-97', 'c-98'],

dtype='object')

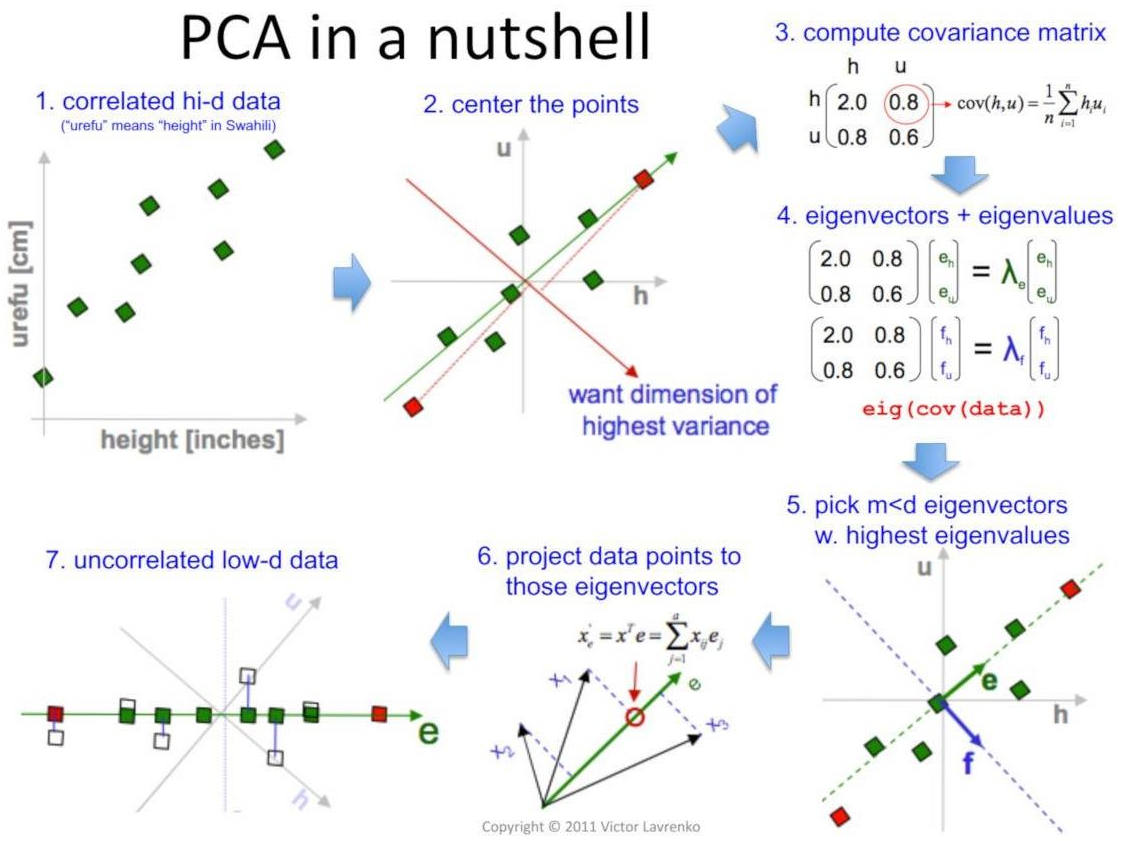
Feature Extraction

Feature extraction transforms the original dataset into a new set of features by reducing the data's dimensionality while retaining important information. This process can uncover new, informative features that enhance model performance.

Types of Feature Extraction:

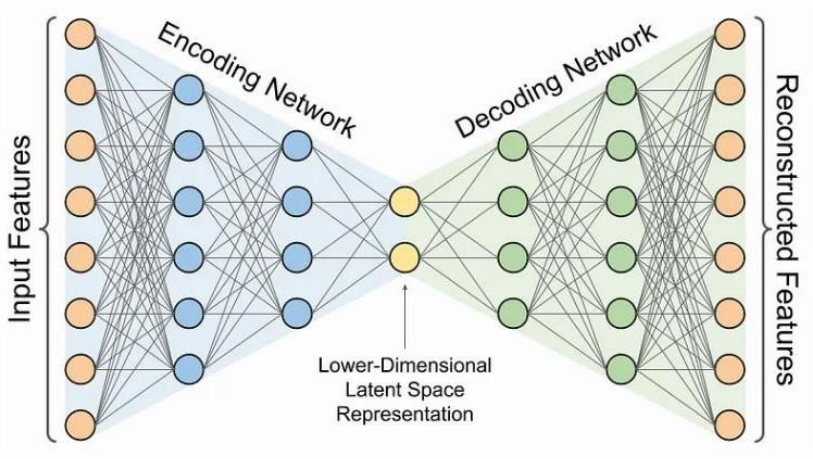
1. **Principal Component Analysis (PCA):** A statistical method that transforms the data into a set of orthogonal (uncorrelated) components, ordered by the amount of variance they explain in the data.
2. **Autoencoders:** A type of artificial neural network used to learn efficient codings of input data. The network is trained to compress the input data into a lower-dimensional representation and then reconstruct it.

PCA



PCA essentially learns a linear transformation that projects the data into another space, where vectors of projections are defined by variance of the data. By restricting the dimensionality to a certain number of components that account for most of the variance of the data set, we can achieve dimensionality reduction.

Autoencoder



Network Architecture for an Ordinary Autoencoder

Autoencoders are neural networks that can be used to reduce the data into a low dimensional latent space by stacking multiple non-linear transformations(layers). They have a encoder-decoder architecture. The encoder maps the input to latent space and decoder reconstructs the input. They are trained using back propagation for accurate reconstruction of the input. In the latent space has lower dimensions than the input, autoencoders can be used for dimensionality reduction. By intuition, these low dimensional latent variables should encode most important features of the input since they are capable of reconstructing it.

In this project, we used both PCA and Autoencoders for feature extraction:

- **PCA:** We applied PCA to reduce the dimensionality of the gene and cell features, transforming them into a smaller set of principal components that retain most of the variance.

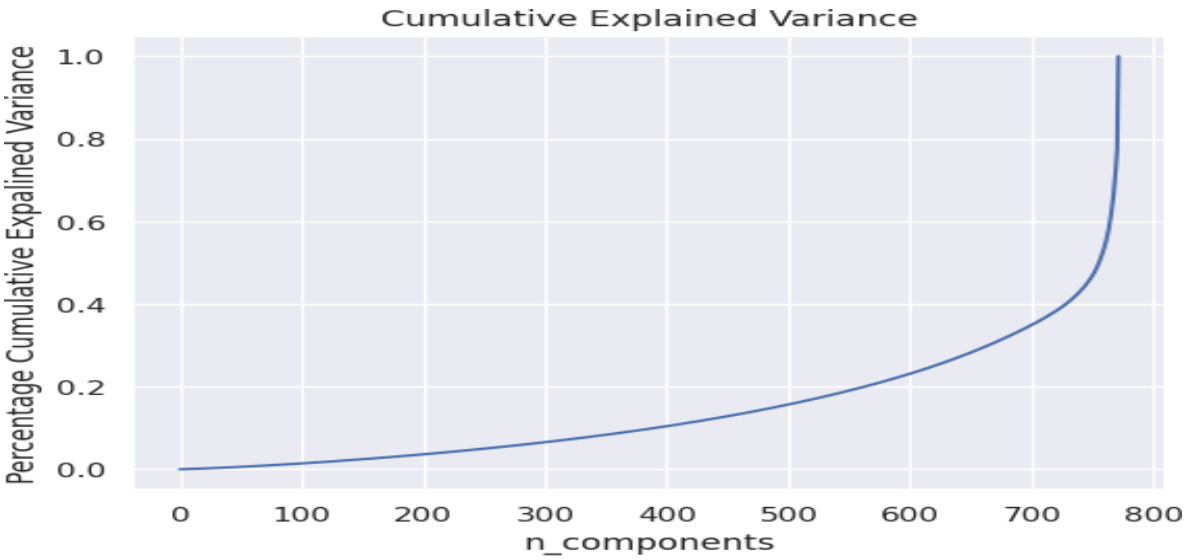
- **Autoencoders:** We trained autoencoders to compress and then reconstruct the data, learning a compact representation of the original features.

After comparing the performance of PCA and Autoencoders, we found that **Autoencoders** provided better feature extraction for our dataset. The autoencoder-based features led to improved model performance compared to the PCA-based features.

Example Results:

The figures below illustrate the variance retained by PCA components and the feature representation learned by autoencoders.

- **PCA Components:**

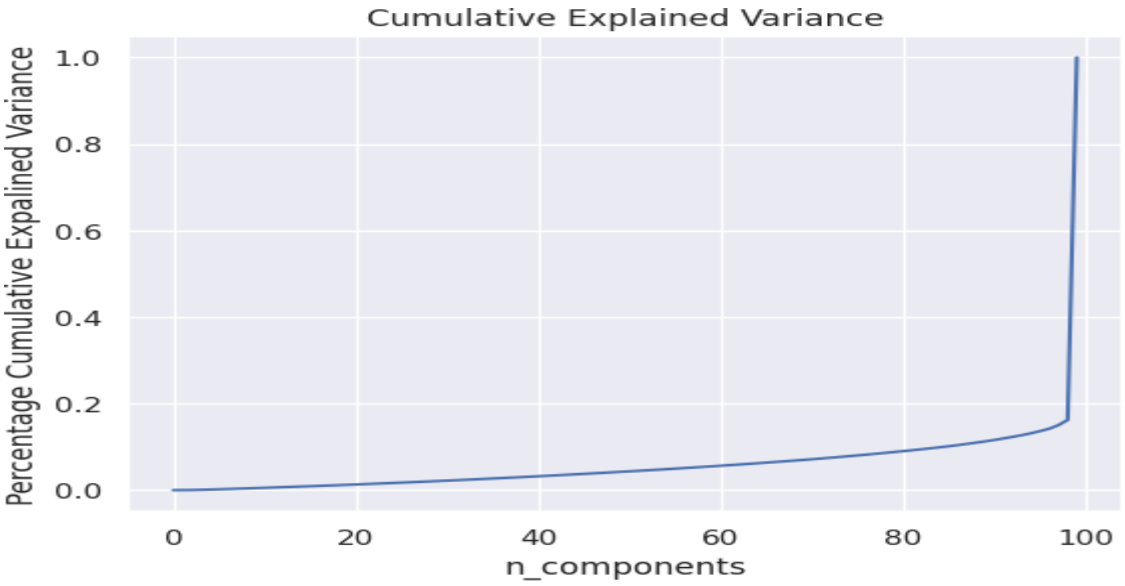


PCA_&_Auto_Encoder.ipynb

```
# Comparison with PCA
pca = PCA(n_components=386)
pca.fit(genes_train)
pca_error = mean_squared_error(genes_test,pca.inverse_transform(pca.transform(genes_test)))
print('PCA Reconstruction Error for Genes is ' + str(pca_error))
```

Snipped

PCA Reconstruction Error for Genes is 0.1629985903595509



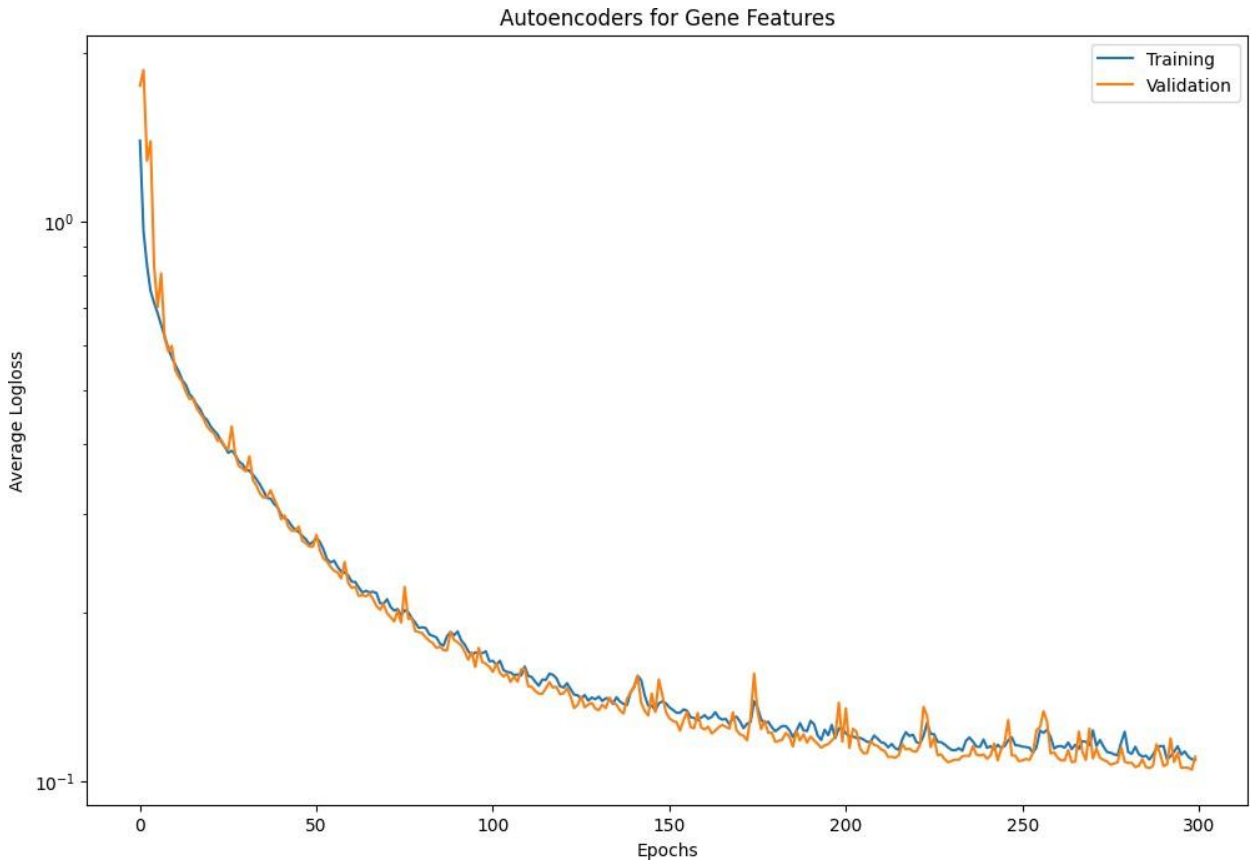
PCA_&_Auto_Encoder.ipynb

```
# Comparison with PCA
pca = PCA(n_components=50)
pca.fit(cells_train)
pca_error = mean_squared_error(cells_test,pca.inverse_transform(pca.transform(cells_test)))
print('PCA Reconstruction Error for Cells is ' + str(pca_error))
```

Snipped

PCA Reconstruction Error for Cells is 0.2175155904767999

- **Autoencoder Features:**



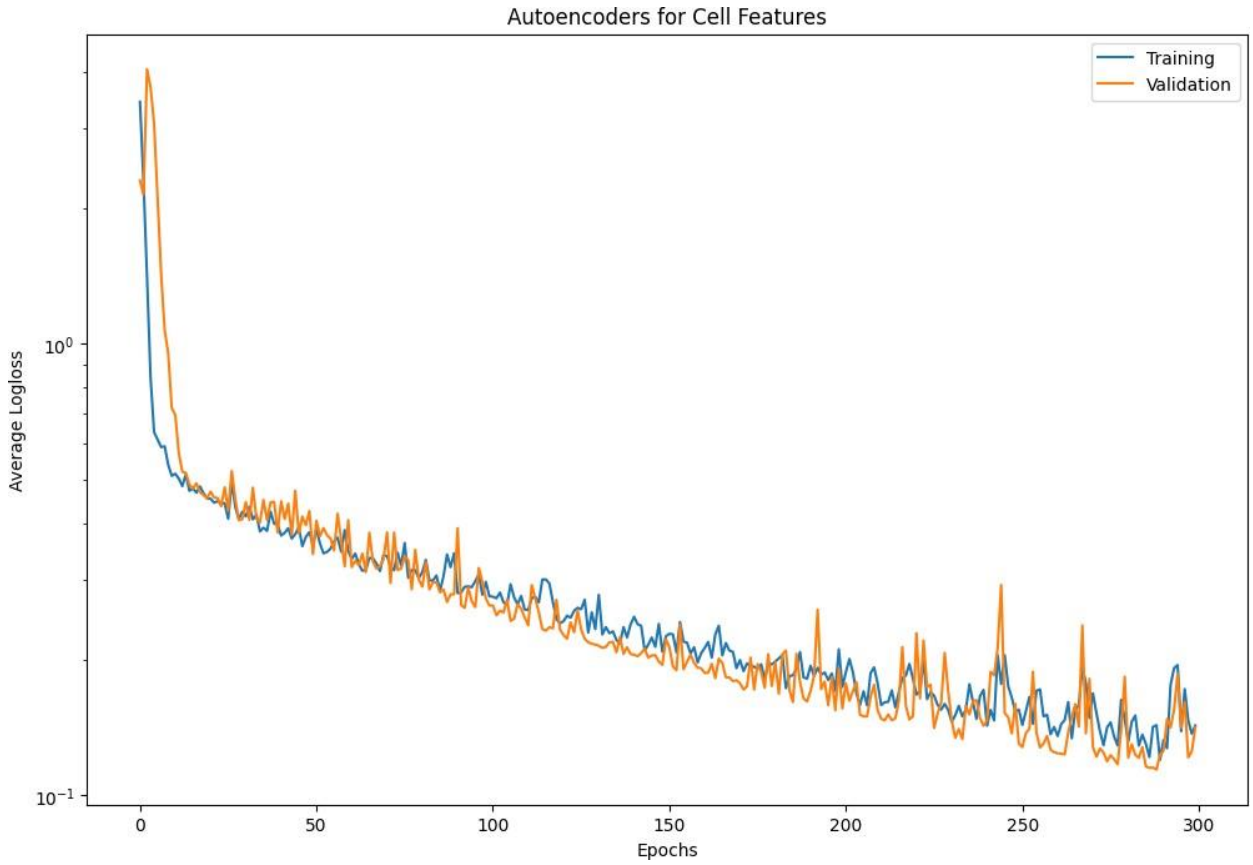
Autoencoders performs well for Gene Features than PCA

Epoch 299/300

47/47 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.1264

Epoch 300/300

47/47 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.1263



Autoencoders performs well for Gene Features than PCA

Epoch 299/300

47/47 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0966

Epoch 300/300

47/47 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0965

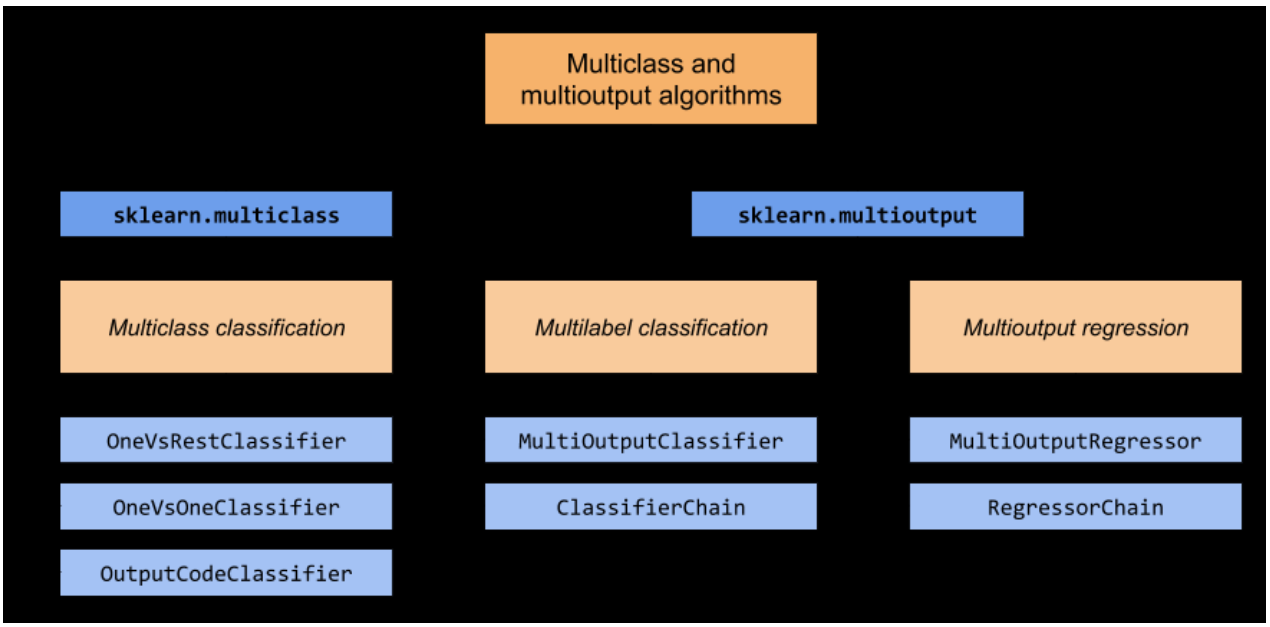
By applying Variance Threshold for feature selection and Autoencoders for feature extraction, we were able to enhance the quality and relevance of the features used in our machine learning models. This approach improved the performance and efficiency of our mechanism of action prediction model.

Machine Learning

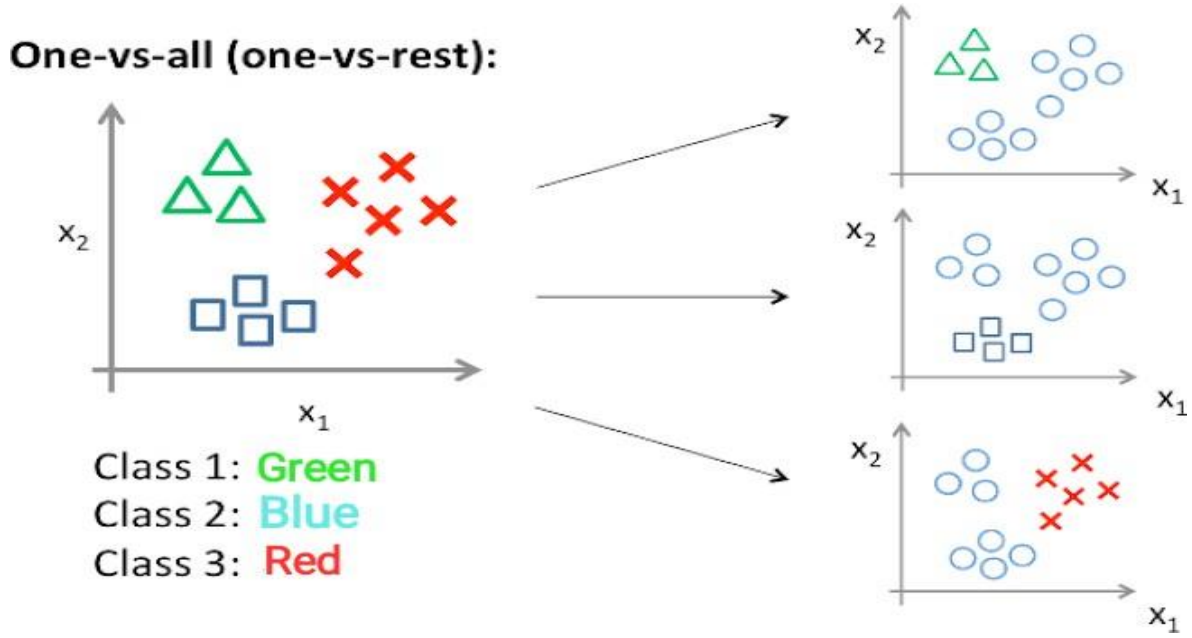
In this project, we employ machine learning and deep learning models to predict the mechanism of action (MoA) of various compounds based on the dataset provided by the Kaggle competition "Mechanisms of Action (MoA) Prediction." These models help classify compounds into different categories, assisting scientists and companies in drug discovery.

Machine Learning Models

Machine learning models identify patterns in the data and make predictions based on these patterns. We used several machine learning models, each with its own advantages and disadvantages. Here, we discuss these models, their working mechanisms, and their mathematical concepts.



1. One vs Rest Classifier:



The one-vs-rest classifier, or one-vs-all, splits a multi-class classification problem into several binary classification problems with a model for each class. For the popular orchid dataset, this would mean that each type of orchid would have a model. The classifier for class *i* is trained to predict if the label is label *i* or not and the final assignation output is given by the label of the most confident classifier. One popular instance of this type of classification is in email tagging, where an email can either be from work, your social networks, friends and family, or spam.

a. Logistic Regression:

i. <u>Advantages:</u> Simple, efficient, works well with small datasets.

ii. Disadvantages: Assumes a linear relationship, sensitive to outliers.

iii. How it Works: Uses the logistic function to model the probability of a binary outcome. The logistic function outputs a probability value between 0 and 1, which is used to predict the class.

iv. Mathematical Concept: The logistic function is defined as:

$$P(y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + ... + \beta_n X_n)}}$$

v. One vs Rest: In multi-class classification, a separate logistic regression model is trained for each class, where one class is treated as the positive class and all other classes as the negative class.

b. GaussianNB:

i. Advantages: Simple, fast, handles continuous and discrete data.

ii. Disadvantages: Assumes features are independent, which is rarely true in real life.

iii. How it Works: Uses Bayes' theorem with Gaussian distribution to model the likelihood of the features given the class.

iv. Mathematical Concept: Bayes' theorem is given by:

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)}$$

v. One vs Rest: Similar to logistic regression, separate models are trained for each class.

c. ExtraTreeClassifier:

i. Advantages: Handles both classification and regression tasks, fast and accurate.

ii. Disadvantages: Can be biased towards training data.

iii. How it Works: Constructs multiple decision trees during training and outputs the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

iv. Mathematical Concept: Nodes are split by randomly selecting a feature and a threshold, then partitioning the data based on these splits.

d. SGDClassifier:

i. Advantages: Efficient for large datasets, supports online learning.

ii. Disadvantages: Requires careful tuning of hyperparameters.

iii. How it Works: Optimizes a loss function iteratively using stochastic gradient descent, which updates the model's parameters based on a subset of the training data.

iv. Mathematical Concept: Weight updates are performed as follows:

$$w := w - \eta \nabla Q_i(w)$$

v. One vs Rest: Separate models are trained for each class.

e. LinearSVC:

i. Advantages: Effective in high-dimensional spaces, versatile.

ii. Disadvantages: Doesn't directly provide probability estimates.

iii. How it Works: Constructs a hyperplane in a high-dimensional space to separate the classes.

iv. Mathematical Concept: Optimization problem solved is:

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum \max(0, 1 - y_i(w \cdot x_i + b))$$

v. One vs Rest: Separate models are trained for each class.

2. Binary Relevance:

Transforms a multi-label classification problem with L labels into L single-label separate binary classification problems using the same base classifier provided in the constructor. The prediction output is the union of all per label classifiers

a. GaussianNB:

Used as mentioned before in One vs Rest Classifier.

3. Adapted Algorithms:

a. MLKNN:

MLkNN builds uses k-NearestNeighbors find nearest examples to a test class and uses Bayesian inference to select assigned labels.

i. Advantages: Simple, interpretable.
ii. Disadvantages: Performance can degrade with large datasets.
iii. How it Works: Combines k-nearest neighbors with a probabilistic approach to make predictions for multi-label data.
iv. Mathematical Concept: Uses maximum a posteriori (MAP) estimate to assign labels.

b. Classifier Chain:

i. Advantages: Captures label dependencies.

ii. Disadvantages: Computationally expensive, can suffer from error propagation.

iii. How it Works: Trains a chain of classifiers, where each classifier deals with a binary relevance problem and takes into account the previous classifiers' predictions.

4. **Label Powerset:**

   a. **Neighbors Classifier, ExtraTreeClassifier, SGDClassifier:**

      Models used as in One vs Rest Classifier, tailored for multi-label classification.

5. **MultiOutputClassifier:**

   a. **XGBoost:**

      i. Advantages: High performance, handles missing data well.

      ii. Disadvantages: Can be complex to tune.

      iii. How it Works: An ensemble method that builds additive trees in a sequential manner. Each tree corrects the errors of the previous ones.

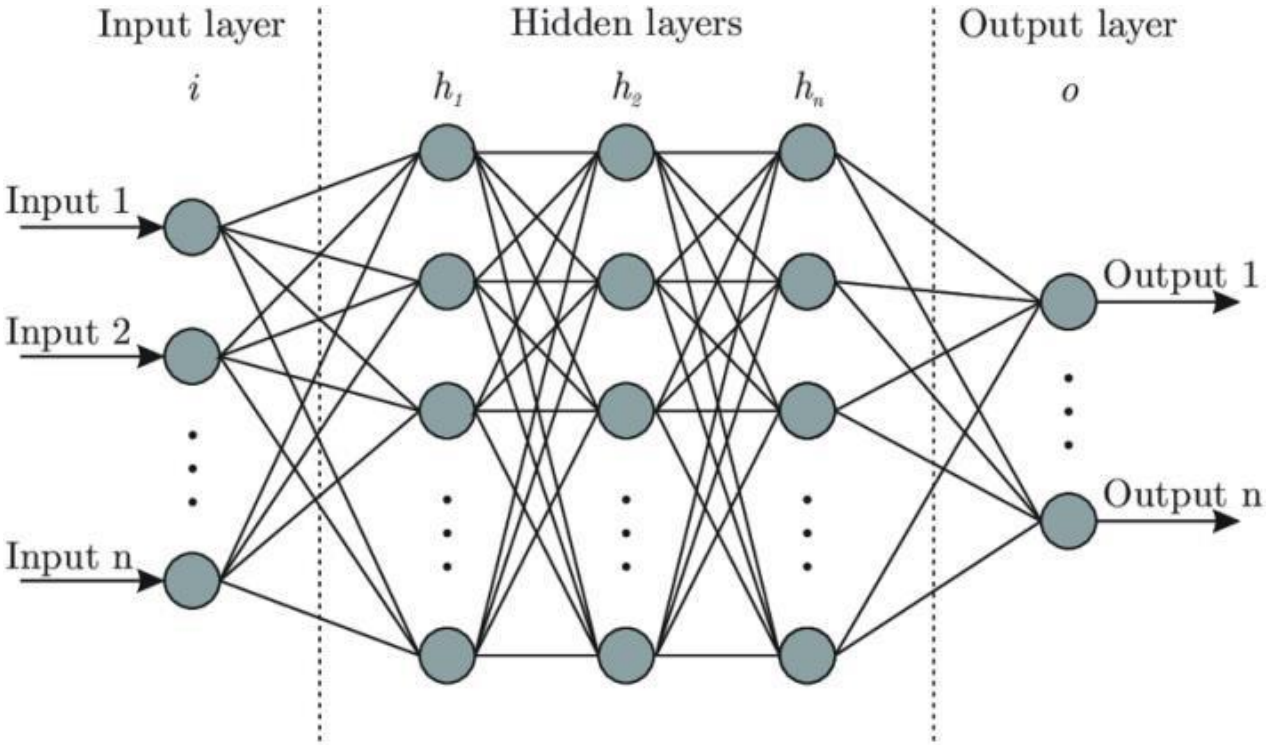
      iv. Mathematical Concept: Objective function is minimized as:

$$f(x) = \sum_{k=1}^{K} \alpha_k h_k(x)$$

Deep Learning Models

Deep learning models automatically learn representations from data and perform complex tasks. In this project, we used artificial neural networks (ANN) to enhance the predictive power of our model.

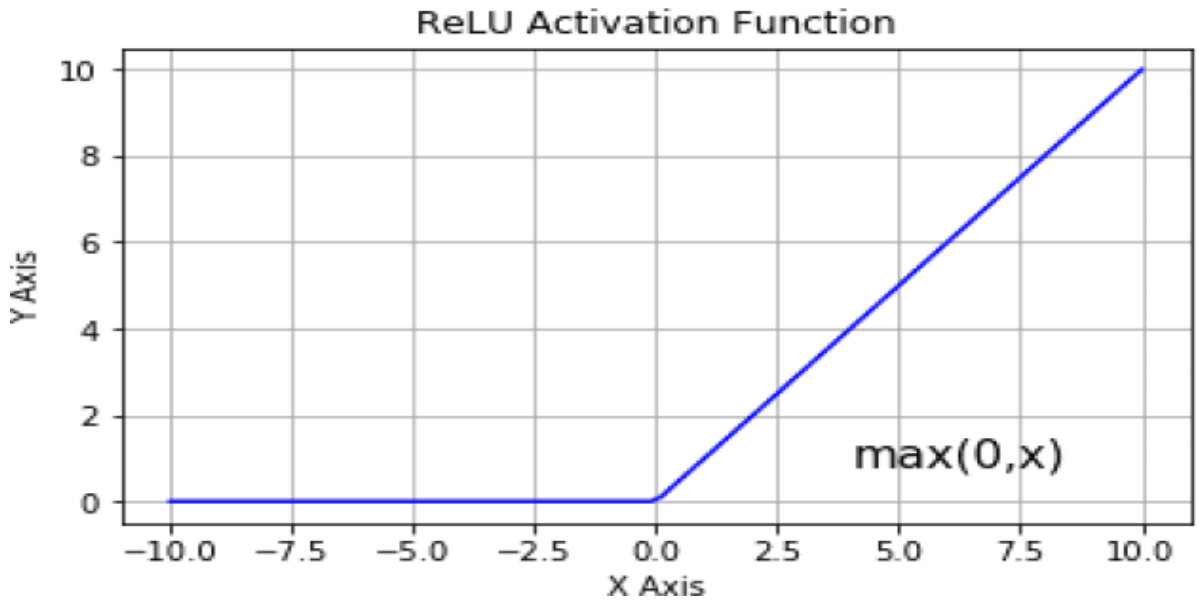
Artificial Neural Networks (ANN):



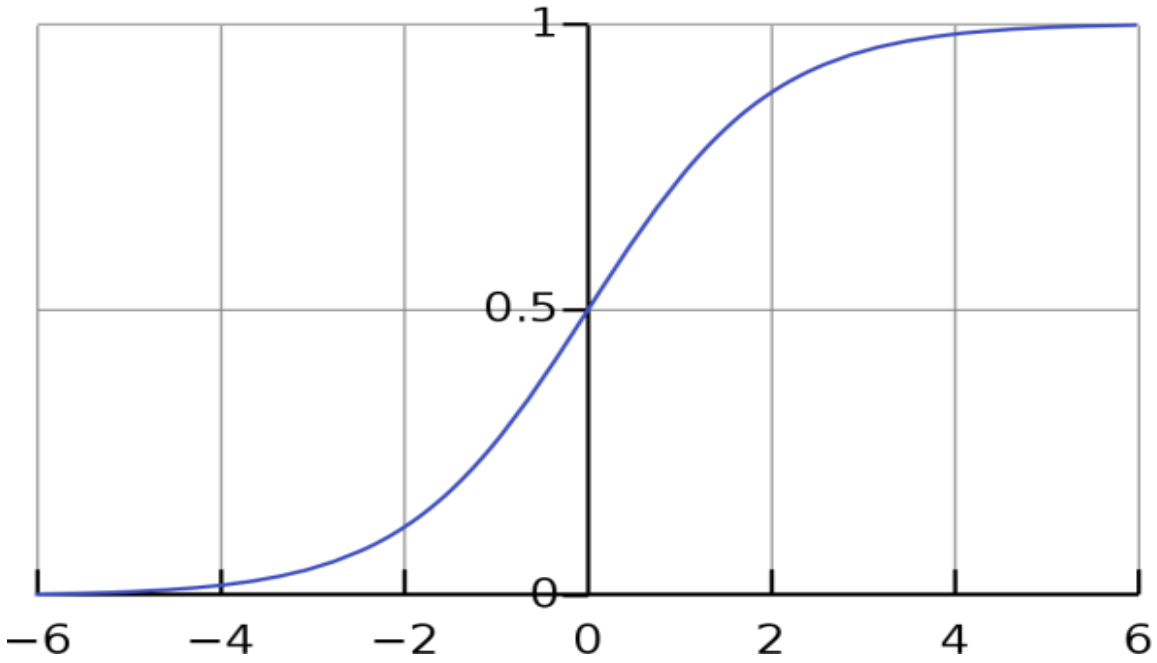
- **How it Works:** An ANN is composed of layers of interconnected nodes (neurons). Each connection has an associated weight. Input data is passed through the network, where each layer applies a linear transformation followed by a non-linear activation function.
- **Parts of ANN:**
    - Input Layer: Receives the input features.

- Hidden Layers: Perform non-linear transformations of the inputs.
- Output Layer: Produces the final output (predictions).

- **Activation Functions:**
  - ReLU (Rectified Linear Unit): Used in hidden layers to introduce non-linearity.



  - Sigmoid: Used in the output layer for binary classification problems.



- **Loss Function:**
  - Binary Cross-Entropy: Measures the performance of a classification model whose output is a probability value between 0 and 1.

$$L = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)]$$

- **Optimizer:**
  - Adam: Adaptive Moment Estimation, combines the advantages of AdaGrad and RMSProp.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

- **Training Techniques:**
  - Early Stopping: Stops training when the model's performance on a validation set stops improving.
  - LearningRateScheduler: Adjusts the learning rate during training based on predefined schedules.
- **Advantages of ANN:**
  - Can model complex relationships.
  - Handles large volumes of data.
  - Learns features automatically.
- **Disadvantages of ANN:**
  - Requires large amounts of data.
  - Computationally expensive.

- Can be a black box, making interpretation difficult.

Metrics and Validation

- **Log Loss:** Used in this competition to evaluate model performance. It measures the accuracy of probabilistic predictions.
  - **Mathematical Concept:**

$$L = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)]$$

- **Iterative Stratification:** Used to maintain the distribution of labels across the training and validation sets. Ensures each set is representative of the overall dataset, reducing bias during model evaluation.

By utilizing these machine learning and deep learning techniques, we were able to develop a robust and accurate model for predicting the mechanism of action of various compounds, contributing to advancements in drug discovery and pharmaceutical research.

Web Technologies

Frontend:

1. React: The core library for building the user interface. React's component-based architecture allows for modular and reusable UI components.
2. Next.js: A React framework that provides server-side rendering and static site generation, improving SEO and performance.
3. Material-UI: A popular React UI framework used for implementing the user interface components like Stepper, Backdrop, and Button with a consistent design.
4. Axios: A promise-based HTTP client for the browser, used for making HTTP requests to the backend.
5. Plotly.js: A graphing library for creating interactive charts and graphs, used for data visualization on the results page.
6. CSS Modules: Scoped CSS for each component to avoid style conflicts and ensure maintainable CSS code.

Backend:

1. Flask: A lightweight WSGI web application framework in Python. It provides the necessary tools to build and deploy the web application.
2. Flask-CORS: A Flask extension for handling Cross-Origin Resource Sharing (CORS), making the API accessible from different domains.
3. Pandas: A data manipulation and analysis library in Python, used for reading and processing the CSV files.

4. Numpy: A fundamental package for scientific computing with Python, used for handling arrays and numerical operations.
5. Scikit-learn: A machine learning library in Python, used for preprocessing and building the predictive models.
6. TensorFlow/Keras: A popular deep learning framework, used for loading and making predictions with the trained neural network models.
7. Plotly: Used in the backend for creating visualizations and generating JSON representations of plots.

Application Architecture

Overview:

The application follows a client-server architecture where the frontend interacts with the backend through RESTful APIs. The architecture is divided into two main components: the frontend and the backend.

Frontend:

Pages:

1. Prediction Page: Users upload their dataset (CSV file) here. The file is validated and then sent to the backend for processing.
2. Results Page: Displays the prediction results in three tabs: Preview, Insights, and Data.

3. Preview: Shows the first 10 rows of the prediction results and provides a download option for the full results.
4. Insights: Provides insights and visualizations of the prediction results.
5. Data: Displays insights and visualizations of the user's dataset.

Backend:

Endpoints:

1. /upload: Handles the file upload, processes the CSV, and makes predictions.
2. /preview: Provides a preview of the prediction results.
3. /dataset_details: Returns detailed information about the dataset.
4. /top_20_json: Returns the top 20 targets visualization data.
5. /lowest_20_json: Returns the lowest 20 targets visualization data.
6. /download: Endpoint for downloading the full prediction results.
7. /visualize: Endpoint for generating visualizations of the dataset based on a selected column.

Data Flow:

1. File Upload: Users upload a CSV file on the Prediction Page.
2. File Validation: The file is validated on the frontend for size and format.
3. Server Processing: The file is sent to the backend, where it is processed, and predictions are made using pre-trained models.
4. Results Generation: The backend generates prediction results and stores them.
5. Results Display: The frontend fetches the results and displays them in the respective tabs on the Results Page.

Chapter 5: Case study

Chapter 5: Case Study

In this chapter, we present a case study where the various parts of the project are connected and the idea is applied to a specific disease, which is diabetes, specifically type 2 diabetes.

5.1 Overview of Diabetes Mellitus

5.1.1 What is diabetes?

- Diabetes mellitus is a group of metabolic diseases characterized by chronic hyperglycemia resulting from defects in insulin secretion, insulin action, or both. This condition can lead to serious health complications if not managed properly.It is a chronic metabolic disorder characterized by high levels of blood glucose (hyperglycemia) resulting from defects in insulin production, insulin action, or both.

- Diabetes is a condition that happens when your blood sugar (glucose) is too high. It develops when your pancreas doesn't make enough insulin or any at all, or when your body isn't responding to the effects of insulin properly. Diabetes affects people of all ages. Most forms of diabetes are chronic (lifelong), and all forms are manageable with medications and/or lifestyle changes.

- Glucose (sugar) mainly comes from carbohydrates in your food

and drinks. It's your body's go-to source of energy. Your blood carries glucose to all your body's cells to use for energy.When glucose is in your bloodstream, it needs help — a "key" — to reach its final destination. This key is insulin (a hormone). If your pancreas isn't making enough insulin or your body isn't using it properly, glucose builds up in your bloodstream, causing high blood sugar (hyperglycemia).Over time, having consistently high blood glucose can cause health problems, such as heart disease, nerve damage and eye issues.

5.1.2 What is insulin?

- Insulin is a protein hormone that plays a critical role in regulating blood glucose levels in the body. It is produced by the beta cells of the pancreas and helps facilitate the uptake of glucose into cells, thereby lowering blood sugar levels. Insulin is not a gene itself, but it is encoded by a gene called the INS gene.

- In the case of the INS gene:
    - Exon 1: The sequence provided, 5'-ATG GCA CCA GTG ACG CCA GCC-3', is part of the coding region that will be transcribed and translated into the insulin protein.
    - Exon 2: The sequence provided, 5'-CTG TGG GTG GGC AGG TGG CTC-3', is another part of the coding region.
    - Intron: The intron is the non-coding sequence between exons 1 and 2, which is spliced out during mRNA

processing.

Exon 1: 5'-ATG GCA CCA GTG ACG CCA GCC-3'

Exon 2: 5'-CTG TGG GTG GGC AGG TGG CTC-3'

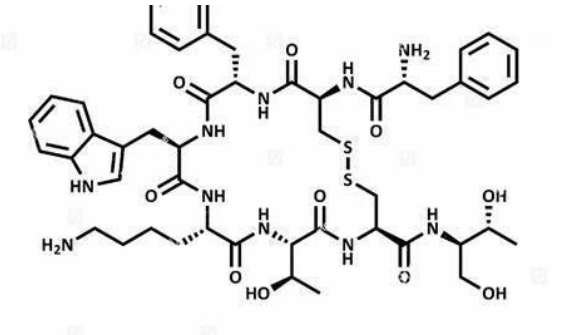


Fig.1 the chemical structure of insulin

5.1.3 Types of Diabetes:

- Type 1 Diabetes (T1D)

  Cause: Autoimmune destruction of the pancreatic beta cells that produce insulin.

  Onset: Typically develops in children and adolescents, though it can occur at any age.

  Symptoms: Rapid weight loss, frequent urination, excessive thirst, constant hunger, fatigue, and blurred vision.

  Management: Requires lifelong insulin therapy, blood glucose monitoring, and dietary management.

- Type 2 Diabetes (T2D)

  Cause: Insulin resistance combined with relative insulin deficiency.

  Onset: Usually occurs in adults over the age of 45, but increasing in younger populations due to rising obesity rates.

  Symptoms: Often develops gradually and may include fatigue, increased thirst, frequent urination, and slow-healing sores or infections.

  Management: Lifestyle changes (diet, exercise), oral medications, and sometimes insulin therapy.

- Gestational Diabetes (GDM)

  Cause: Insulin resistance during pregnancy.

  Onset: Occurs during pregnancy and usually resolves after childbirth.

  Symptoms: Typically asymptomatic and detected through routine screening.

  Management: Diet, exercise, and sometimes insulin or oral medications.

- Maturity-Onset Diabetes of the Young (MODY)

  Cause: Genetic mutations affecting insulin production.

Onset: Typically diagnosed in adolescents or young adults.

Symptoms: Mild, often similar to Type 2 diabetes.

Management: Depends on the specific genetic mutation; may include lifestyle changes, oral medications, or insulin.

5.1.4 Symptoms of Diabetes

- Increased thirst and hunger
- Frequent urination
- Unexplained weight loss
- Fatigue
- Blurred vision
- Slow-healing sores
- Frequent infections
- Tingling or numbness in hands or feet

5.1.5 What causes diabetes?

Too much glucose circulating in your bloodstream causes diabetes, regardless of the type. However, the reason why your blood glucose levels are high differs depending on the type of diabetes.

- Causes of diabetes include:

  1- Genetic factors

  - Insulin resistance: Type 2 diabetes mainly results from insulin resistance.
  - Autoimmune disease: Type 1 diabetes and LADA happen when your immune system attacks the insulin-producing cells in your pancreas.
  - Hormonal imbalances: During pregnancy, the placenta releases hormones that cause insulin resistance. can also cause Type 2 diabetes.
  - Pancreatic damage: Physical damage to your pancreas — from a condition, surgery or injury — can impact its ability to make insulin, resulting in Type 3c diabetes.
  - Genetic mutations: Certain genetic mutations can cause MODY and neonatal diabetes.

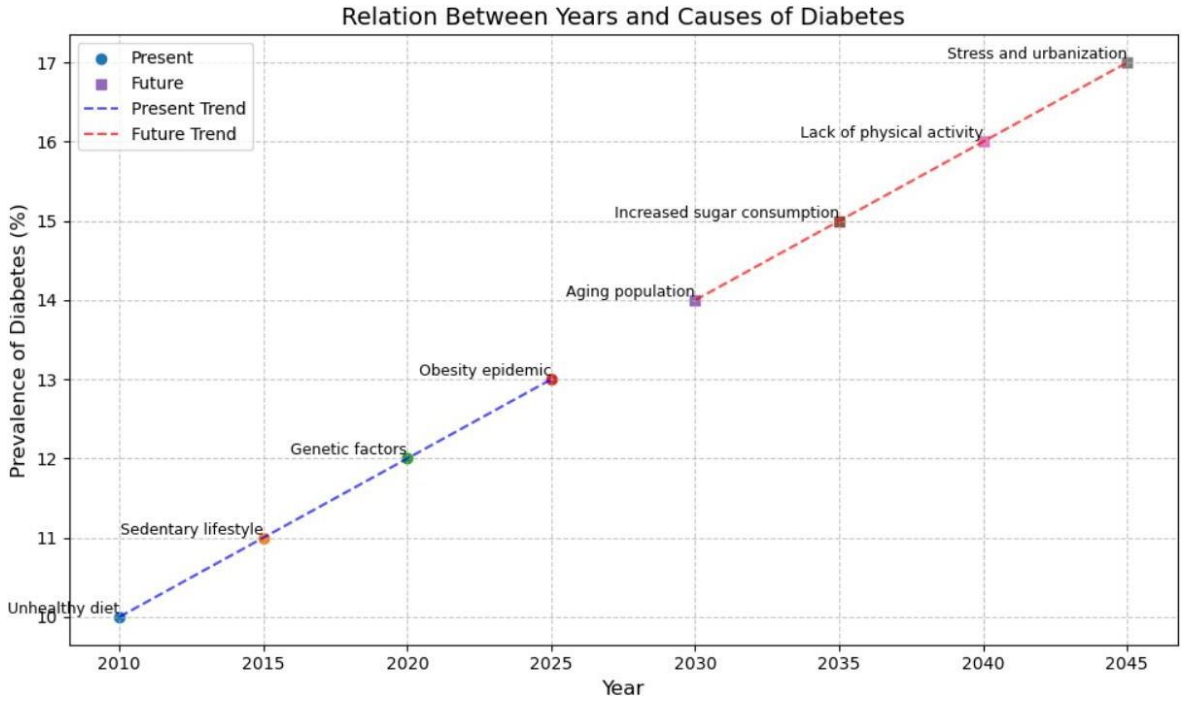
  2- Lifestyle factors (diet, physical activity)

  3- Obesity and overweight

  4- Age and ethnicity

Long-term use of certain medications can also lead to Type 2 diabetes, including HIV/AIDS medications and corticosteroids.

- Here is a line chart that shows the relation between Diabetes & its causes.



5.1.6 How common is diabetes?

- Diabetes is common. Approximately 37.3 million people in the United States have diabetes, which is about 11% of the population. Type 2 diabetes is the most common form, representing 90% to 95% of all diabetes cases.

- About 537 million adults across the world have diabetes. Experts predict this number will rise to 643 million by 2030 and 783 million by 2045.

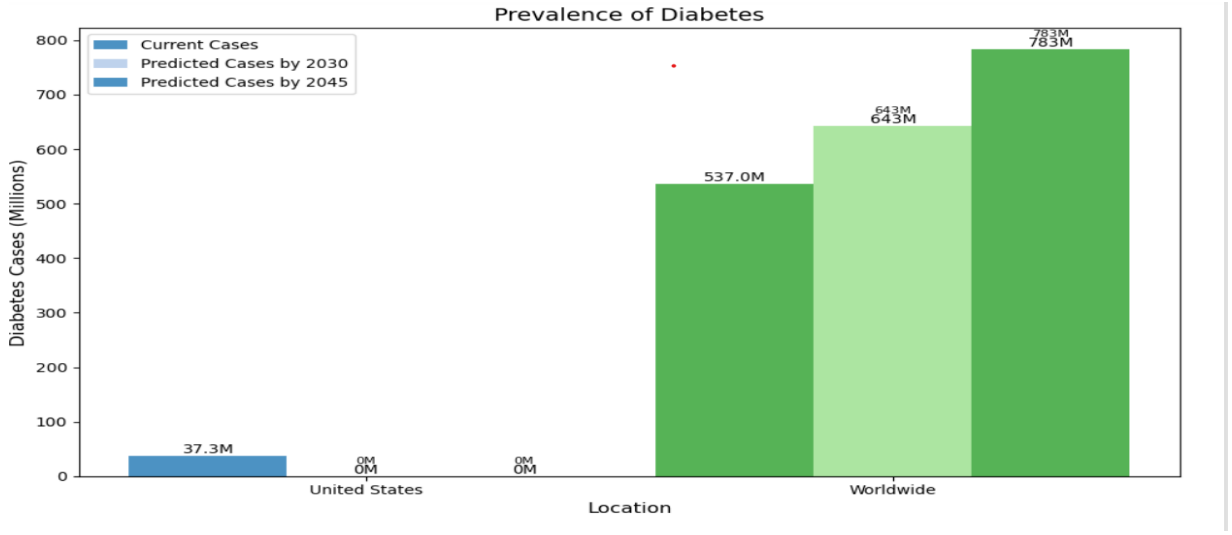


Fig.2 A statistical survey on diabetes in america

5.1.7 Diagnosis of Diabetes:

- Fasting Plasma Glucose (FPG): Measures blood sugar after an overnight fast.
- Oral Glucose Tolerance Test (OGTT): Measures blood sugar before and after drinking a sugary solution.
- Hemoglobin A1c Test: Reflects average blood sugar levels over the past 2-3 months.
- Random Plasma Glucose Test: Measures blood sugar without regard to when you last ate.

5.2 Type 2 Diabetes

5.2.1 Overview of Type 2 Diabetes

- Understanding the differences between type 2 diabetes and other forms of diabetes is crucial for effective management and treatment. Type 2 diabetes is primarily characterized by insulin resistance and relative insulin deficiency, often associated with lifestyle factors and obesity, whereas other types of diabetes have different etiologies and management approaches.

5.2.2 T2D Risk Factors

- **Genetics**: A family history of diabetes increases the risk.
- **Obesity**: Excess body fat, especially around the abdomen, contributes to insulin resistance.
- **Sedentary Lifestyle**: Lack of physical activity can exacerbate insulin resistance.
- **Age**: Risk increases with age, particularly after 45 years.
- **Diet**: Poor dietary choices, such as high intake of sugary and processed foods, contribute to the development of T2D.

5.2.3 T2D Pathophysiology

Type 2 diabetes (T2D) is a complex metabolic disorder that develops gradually over time. It is primarily characterized by two main pathological processes: insulin resistance and beta-cell dysfunction. Here's a detailed look at how T2D develops:

1. Insulin Resistance

Definition: Insulin resistance occurs when the body's cells (mainly muscle, fat, and liver cells) do not respond effectively to insulin, a hormone that facilitates the uptake of glucose from the blood into cells for energy.

Mechanism:

- Cellular Level: In insulin resistance, cells fail to respond to insulin's signal to absorb glucose. This occurs due to alterations in the insulin signaling pathway, which can be caused by factors such as obesity, inflammation, and lipid accumulation in cells.
- Insulin Receptor Substrate (IRS) Dysfunction: One of the early steps in insulin signaling is the phosphorylation of insulin receptor substrates (IRS). In insulin resistance, IRS proteins may be abnormally phosphorylated on serine residues instead of tyrosine residues, impairing their function.

- Inflammation: Adipose tissue, especially visceral fat, releases pro-inflammatory cytokines (e.g., TNF-α, IL-6) that interfere with insulin signaling, promoting insulin resistance.
- Free Fatty Acids (FFA): Elevated levels of FFAs in the blood, often due to obesity, contribute to insulin resistance by disrupting normal insulin signaling pathways.

Consequences:

- Hyperinsulinemia: In response to insulin resistance, the pancreas compensates by producing more insulin, leading to higher insulin levels in the blood.
- Hyperglycemia: Despite increased insulin production, glucose uptake by cells is impaired, leading to elevated blood glucose levels.

2. Beta-Cell Dysfunction

Definition: Beta-cell dysfunction refers to the inability of pancreatic beta cells to produce sufficient insulin to overcome insulin resistance and maintain normal blood glucose levels.

Mechanism:

- Compensatory Hyperinsulinemia: Initially, beta cells increase insulin production to compensate for insulin resistance. This is often sufficient to maintain normoglycemia in the early stages.

- Beta-Cell Exhaustion: Over time, chronic high demand for insulin production can lead to beta-cell exhaustion, reducing their ability to produce insulin.
- Glucotoxicity: Chronic hyperglycemia can directly damage beta cells, impairing their function and leading to apoptosis (cell death).
- Lipotoxicity: Elevated FFAs and triglycerides can also be toxic to beta cells, further impairing insulin secretion.
- Genetic Predisposition: Some individuals may have a genetic predisposition to beta-cell dysfunction, making them more susceptible to developing T2D when exposed to environmental risk factors like obesity and poor diet.

Consequences:

- Decreased Insulin Secretion: As beta cells become dysfunctional and die, insulin secretion decreases, exacerbating hyperglycemia.
- Progressive Hyperglycemia: The combination of insulin resistance and inadequate insulin secretion leads to sustained high blood glucose levels, the hallmark of T2D.

5.2.4 Complications of diabetes?

- Diabetes can lead to acute (sudden and severe) and long-term complications — mainly due to extreme or prolonged high blood sugar levels.

- **Acute diabetes complications**

  Acute diabetes complications that can be life-threatening include:

- Hyperosmolar hyperglycemic state (HHS): This complication mainly affects people with Type 2 diabetes. It happens when your blood sugar levels are very high (over 600 milligrams per deciliter or mg/dL) for a long period, leading to severe dehydration and confusion. It requires immediate medical treatment.
- Severe low blood sugar (hypoglycemia): Hypoglycemia happens when your blood sugar level drops below the range that's healthy for you. Severe hypoglycemia is very low blood sugar. It mainly affects people with diabetes who use insulin. Signs include blurred or double vision, clumsiness, disorientation and seizures. It requires treatment with emergency glucagon and/or medical intervention.

- Long-term diabetes complications

  Blood glucose levels that remain high for too long can damage your body's tissues and organs. This is mainly due to damage to your blood vessels and nerves, which support your body's tissues.

  Cardiovascular (heart and blood vessel) issues are the most common type of long-term diabetes complication. They include:

- Cardiovascular Disease: Increased risk of heart disease, stroke, and atherosclerosis.
- Neuropathy: Nerve damage that can cause pain, tingling, and loss of sensation.

Living with diabetes can also affect your mental health. People with diabetes are two to three times more likely to have depression than people without diabetes.

In Conclusion, The development of type 2 diabetes involves a progressive decline in the body's ability to use and produce insulin effectively. Initially driven by insulin resistance, often due to obesity and inflammation, the condition is exacerbated by the eventual dysfunction and loss of pancreatic beta cells. Understanding these processes highlights the importance of early intervention and lifestyle modifications to prevent or delay the onset of T2D.

5.2.4.1 Genetic Complexity of T2D

Here are the genetic malfunctions leading to T2D.It has a more complex genetic basis, Some of the key genes linked to T2D risk include:

(ignoring the environmental factors):

- TCF7L2: Variations in the TCF7L2 gene are among the most significant genetic risk factors for T2D. This gene plays a role in regulating insulin secretion and glucose production.
- PPARG: The PPARG gene is involved in fat cell development and insulin sensitivity. Mutations in PPARG can contribute to insulin resistance, a hallmark of T2D.
- FTO: The FTO gene is associated with increased body mass index (BMI) and risk of T2D. Variations in FTO influence appetite and energy expenditure, factors that impact obesity and diabetes.
- KCNJ11 and ABCC8: These genes encode subunits of the ATP-sensitive potassium channel in pancreatic beta cells, which is crucial for insulin release. Mutations in these genes can impair insulin secretion.

- Type 2 diabetes involves multiple genes affecting insulin secretion and action, such as TCF7L2, PPARG, and FTO.

- Understanding the genetic basis of diabetes helps in identifying individuals at risk and developing targeted therapies to manage and prevent the disease.

1- TCF7L2 (Transcription Factor 7-Like 2)

SNP: rs7903146

- **Location**: Intronic region of the TCF7L2 gene.

- **Alleles**: C/T (the T allele is associated with increased risk of T2D).

2- PPARG (Peroxisome Proliferator-Activated Receptor Gamma)

SNP: rs1801282 (Pro12Ala)

- **Location**: Exonic region.
- **Sequence Change**: Proline (C) to Alanine (G) at codon 12.
- **Alleles**: C/G (the G allele is associated with a decreased risk of T2D).

3- FTO (Fat Mass and Obesity-Associated Gene)

SNP: rs9939609

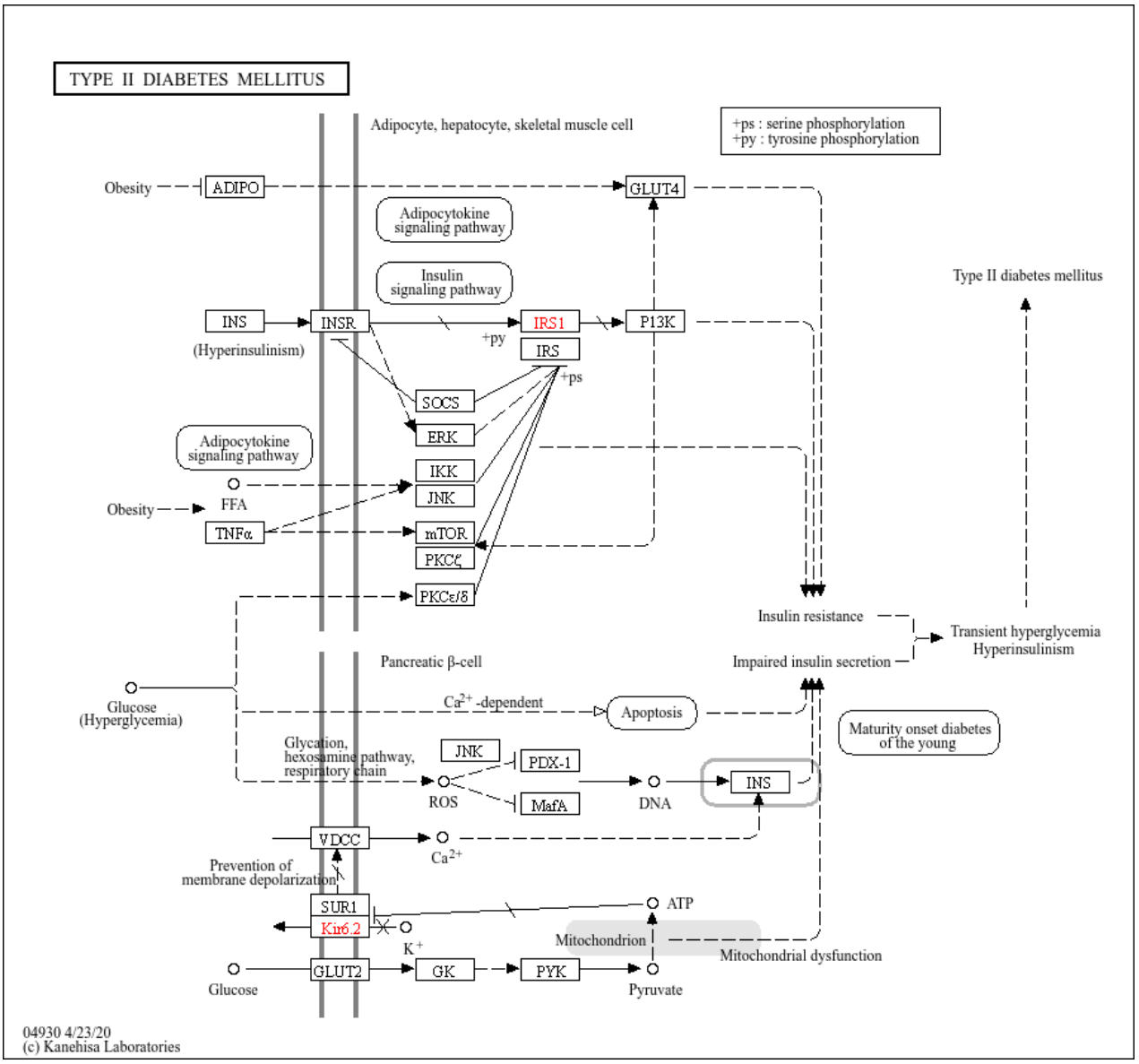
- **Location**: Intronic region.
- **Alleles**: T/A (the A allele is associated with increased BMI and T2D risk).

4- KCNJ11 (Potassium Inwardly Rectifying Channel, Subfamily J, Member 11)

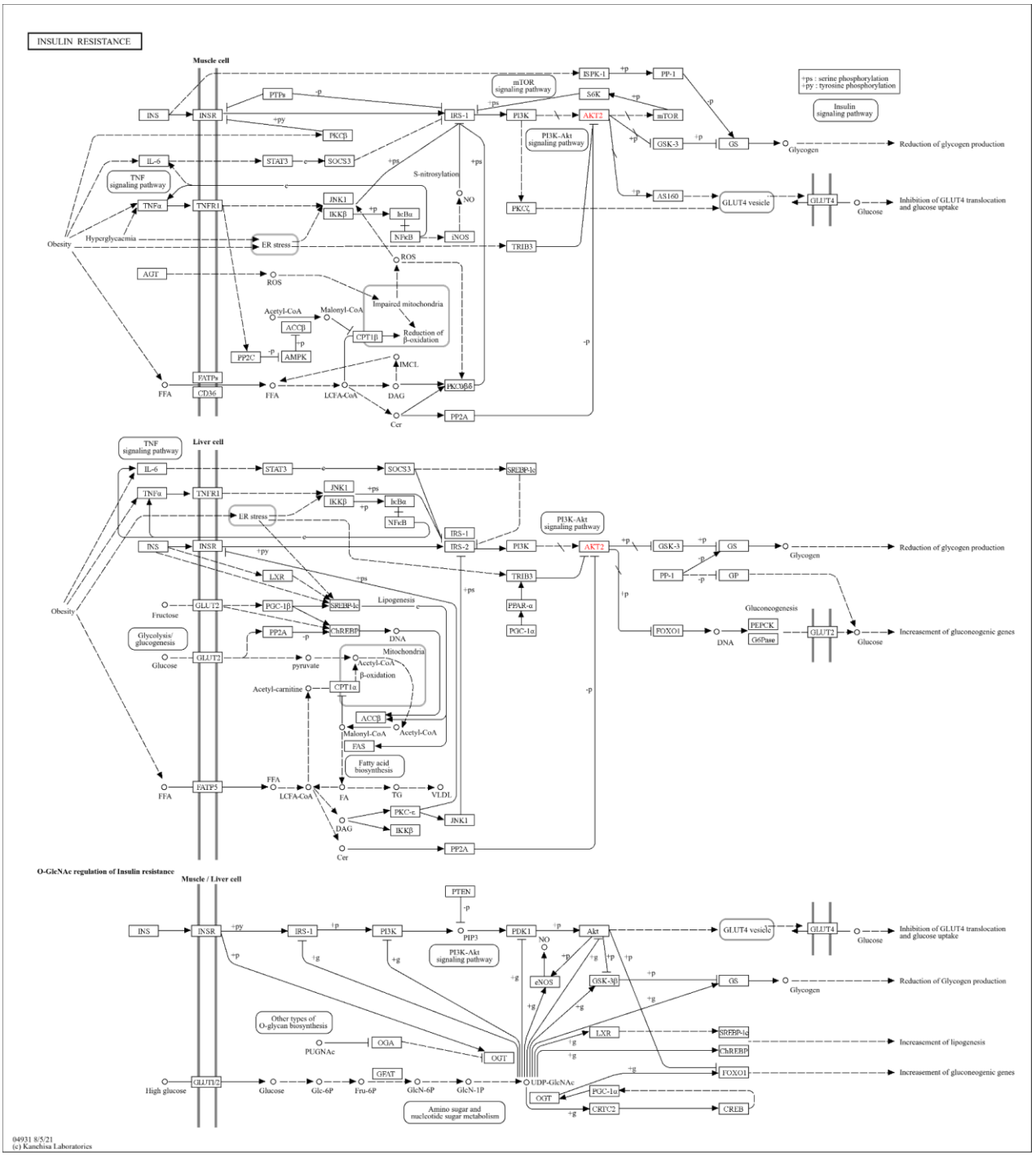
SNP: rs5219 (E23K)

- **Location**: Exonic region.
- **Sequence Change**: Glutamate (G) to Lysine (A) at codon 23.
- **Alleles**: G/A (the A allele is associated with increased risk of T2D).

Here is a diagram that shows the metabolic pathway of T2D:



Here is the metabolic pathway of the insulin resistance:



5.3 MoAs in Diabetes Treatment

5.3.1 Importance of MoAs in Diabetes

- Understanding the mechanisms of action (MoAs) of various diabetes treatments is crucial for effective management of the disease. MoAs describe how a drug or intervention works at the molecular, cellular, or physiological level to achieve its therapeutic effects. Here's why MoAs are important in managing diabetes, particularly type 2 diabetes (T2D):

  1. Personalized Treatment

  Tailoring Therapies: Different patients may respond differently to various treatments based on their specific pathophysiology, genetic makeup, and lifestyle factors. Understanding the MoAs allows healthcare providers to select the most appropriate therapy for each individual, enhancing efficacy and minimizing side effects.

  Combination Therapy: T2D is often managed with a combination of drugs that target different aspects of the disease. Knowing the MoAs of these drugs helps in designing effective combination therapies that can address multiple pathways simultaneously, providing better glycemic control.

2. Targeting Underlying Pathophysiology

Addressing Insulin Resistance: Drugs like metformin and thiazolidinediones (TZDs) work by improving insulin sensitivity in peripheral tissues. Understanding their MoAs helps in targeting insulin resistance, a core defect in T2D.

3. Improving Safety and Efficacy

Minimizing Side Effects: By understanding the MoAs, healthcare providers can anticipate and mitigate potential side effects. For instance, TZDs are effective in improving insulin sensitivity but can cause weight gain and fluid retention, so they should be used cautiously in patients with heart failure.

Managing Drug Interactions: Knowledge of MoAs helps in predicting and managing drug-drug interactions, ensuring that combined therapies do not negatively impact each other's effectiveness or safety.

Optimizing Dosing Regimens: Understanding how and when a drug acts can help in optimizing dosing schedules to maximize therapeutic effects and minimize side effects. For example, medications affecting postprandial glucose levels might be timed with meals.

4. Advancing Research and Development

Developing Personalized Medicine: As we learn more about the genetic and molecular basis of diabetes, MoAs can help in developing personalized medicine approaches, tailoring treatments to the individual characteristics of each patient's disease.

- Understanding the mechanisms of action of diabetes treatments is fundamental to effective disease management. It allows for personalized treatment plans, targeted interventions, improved safety and efficacy, and advances in research and development. By leveraging this knowledge, healthcare providers can better control diabetes, reduce complications, and enhance the quality of life for patients.

5.3.2 Common Mechanisms of Action

- **Insulin secretagogue**: These agents stimulate the pancreas to secrete more insulin. Examples include sulfonylureas and meglitinides. They are commonly used in the treatment of type 2 diabetes.
- **Insulin sensitizer**: These agents improve the sensitivity of body tissues to insulin. Examples include metformin and

thiazolidinediones. They are essential in the management of type 2 diabetes as they help the body utilize insulin more effectively.

- **Dipeptidyl peptidase inhibitor (DPP-4 inhibitor)**: These agents prevent the breakdown of incretin hormones, which help increase insulin secretion and decrease glucagon release. Examples include sitagliptin and saxagliptin. They are used in the treatment of type 2 diabetes.
- **Peroxisome proliferator-activated receptor (PPAR) agonist/antagonist**: PPAR agonists, particularly PPAR-γ agonists (such as thiazolidinediones), improve insulin sensitivity and are used in the treatment of type 2 diabetes. PPAR-α agonists (such as fibrates) are used to manage dyslipidemia but can also have effects on glucose metabolism.

5.4 Drugs and Medications for T2D

- Diabetes type 2 management typically involves several classes of medications, each with distinct mechanisms of action (MOAs). Here are some common classes of drugs used in the treatment of type 2 diabetes along with their MOAs:

  1- Biguanides (e.g., Metformin):

MOA: Decreases glucose production in the liver, improves insulin sensitivity in muscle tissue, and reduces glucose absorption in the intestines.

2- Sulfonylureas (e.g., Glimepiride, Glipizide):

MOA: Stimulates insulin secretion from pancreatic beta cells, thereby increasing insulin levels in the blood.

3- Meglitinides (e.g., Repaglinide, Nateglinide):

MOA: Stimulates rapid and short-lived insulin secretion from pancreatic beta cells in response to glucose.

4- Thiazolidinediones (TZDs or glitazones) (e.g., Pioglitazone, Rosiglitazone):

MOA: Improves insulin sensitivity in peripheral tissues (muscle and fat) by activating peroxisome proliferator-activated receptor gamma (PPAR-gamma).

5- Dipeptidyl Peptidase-4 (DPP-4) Inhibitors (e.g., Sitagliptin, Saxagliptin):

MOA: Inhibits the enzyme DPP-4, which degrades incretin hormones (GLP-1 and GIP). This leads to

increased insulin secretion and decreased glucagon secretion in a glucose-dependent manner.

6- SGLT-2 Inhibitors (e.g., Canagliflozin, Dapagliflozin):

MOA: Inhibits the sodium-glucose cotransporter 2 (SGLT-2) in the renal tubules, reducing renal glucose reabsorption and increasing urinary glucose excretion.

7- GLP-1 Receptor Agonists (e.g., Liraglutide, Exenatide):

MOA: Mimics the action of incretin hormones (GLP-1), stimulating insulin secretion in a glucose-dependent manner, inhibiting glucagon release, slowing gastric emptying, and promoting satiety.

- These medications are often used in combination to achieve optimal blood glucose control in patients with type 2 diabetes. Treatment plans are individualized based on factors such as the patient's overall health, kidney function, cardiovascular status, and preferences.

New and emerging drugs and targets for type 2 diabetes: reviewing the evidence

- Authors: BR Miller, H Nguyen, CJH Hu, C Lin

Summary: This review discusses various options for the treatment of type 2 diabetes mellitus, highlighting several

emerging drugs and their efficacy.

Read the full article

- Novel diabetes drugs and the cardiovascular specialist

  Authors: N Sattar, MC Petrie, B Zinman, JL Januzzi

  Summary: This paper reviews new diabetes drugs, particularly focusing on their impact on cardiovascular health.

  Read the full article

- Future glucose-lowering drugs for type 2 diabetes

  Authors: CJ Bailey, AA Tahrani, AH Barnett

  Summary: This paper reviews future developments in glucose-lowering drugs and their potential impact.Read the full article

This case study illustrates the practical application of the project in addressing a common and impactful chronic disease. By leveraging data analysis, computational modeling, and personalized treatment strategies, the system has the potential to significantly improve the management and outcomes of type 2 diabetes.

Chapter 6:

Related Work

Chapter 4: Related Work

4.1. Similar Projects

In this section, we explore several similar projects that have tackled multi-label classification problems using different datasets and models. These projects provide insights into various approaches and solutions that can be applied to our Mechanism of Action (MoA) prediction project.

Project 1: Toxic Comment Classification Challenge

- **Dataset:** The Jigsaw Toxic Comment Classification Challenge dataset, containing Wikipedia comments labeled with multiple toxicity types (e.g., toxic, severe toxic, obscene, threat, insult, identity hate).
- **Models Used:** Logistic Regression, Support Vector Machines (SVM), Gradient Boosting Machines (GBM), and Recurrent Neural Networks (RNN).
- **Solutions:**
    - **Feature Engineering:** Text preprocessing (tokenization, stop word removal) and TF-IDF vectorization.

- **Model Training:** Logistic Regression with One-vs-Rest strategy, SVM, GBM for boosting, and RNN with LSTM for capturing sequential dependencies in text.
- **Outcome:** Achieved competitive performance with a combination of logistic regression and RNNs, demonstrating the effectiveness of combining linear and deep learning models for multi-label text classification.

Project 2: Planet: Understanding the Amazon from Space

- **Dataset:** The Planet: Understanding the Amazon from Space dataset, containing satellite images labeled with multiple tags related to atmospheric conditions and land cover/land use.
- **Models Used:** Convolutional Neural Networks (CNN), Random Forest, and Multi-Label K-Nearest Neighbors (ML-KNN).
- **Solutions:**
  - **Feature Engineering:** Image augmentation (rotations, flips) and feature extraction using pre-trained CNNs (e.g., ResNet, VGG).
  - **Model Training:** Fine-tuned CNNs for multi-label image classification, Random Forest for capturing non-linear relationships, and ML-KNN for considering label dependencies.

- **Outcome:** Achieved high accuracy with CNNs and demonstrated the importance of image augmentation and pre-trained models in multi-label image classification tasks.

Project 3: YouTube-8M Video Understanding Challenge

- **Dataset:** The YouTube-8M dataset, containing millions of YouTube videos labeled with multiple video categories.
- **Models Used:** Deep Neural Networks (DNN), Recurrent Neural Networks (RNN), and Attention Mechanisms.
- **Solutions:**
  - **Feature Engineering:** Frame-level feature extraction using pre-trained models and aggregation techniques (mean, max pooling).
  - **Model Training:** DNNs for learning high-dimensional feature representations, RNNs with LSTM/GRU for capturing temporal dependencies, and attention mechanisms for focusing on important parts of the video.
  - **Outcome:** Achieved state-of-the-art results with a combination of DNNs, RNNs, and attention mechanisms, showcasing the power of deep learning in multi-label video classification.

Project 4: Multi-Label Bird Species Classification

- **Dataset:** The Cornell Birdcall Identification dataset, containing audio recordings labeled with multiple bird species.
- **Models Used:** Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM) networks, and Ensemble Methods.
- **Solutions:**
    - **Feature Engineering:** Audio preprocessing (spectrograms, mel-frequency cepstral coefficients), data augmentation (noise addition, pitch shifting).
    - **Model Training:** CNNs for extracting spatial features from spectrograms, LSTM networks for capturing temporal patterns in audio, and ensemble methods for combining predictions from multiple models.
    - **Outcome:** Achieved high accuracy with CNNs and LSTMs, emphasizing the importance of combining spatial and temporal features in multi-label audio classification.

Project 5: Emotion Recognition in Conversations (ERC)

- **Dataset:** The Emotion Recognition in Conversations (ERC) dataset, containing text-based conversations labeled with multiple emotions (e.g., happiness, sadness, anger, surprise).

- **Models Used:** Transformer-based models (e.g., BERT), Bi-Directional LSTM (Bi-LSTM), and Graph Neural Networks (GNN).
- **Solutions:**
    - **Feature Engineering:** Text preprocessing (tokenization, embedding generation), context modeling using conversation history.
    - **Model Training:** Fine-tuned transformer models for capturing context and semantics, Bi-LSTM for modeling sequential data, and GNN for capturing relational dependencies between conversation participants.
    - **Outcome:** Achieved state-of-the-art performance with transformer models and GNNs, demonstrating the effectiveness of advanced NLP techniques in multi-label emotion recognition.

Summary

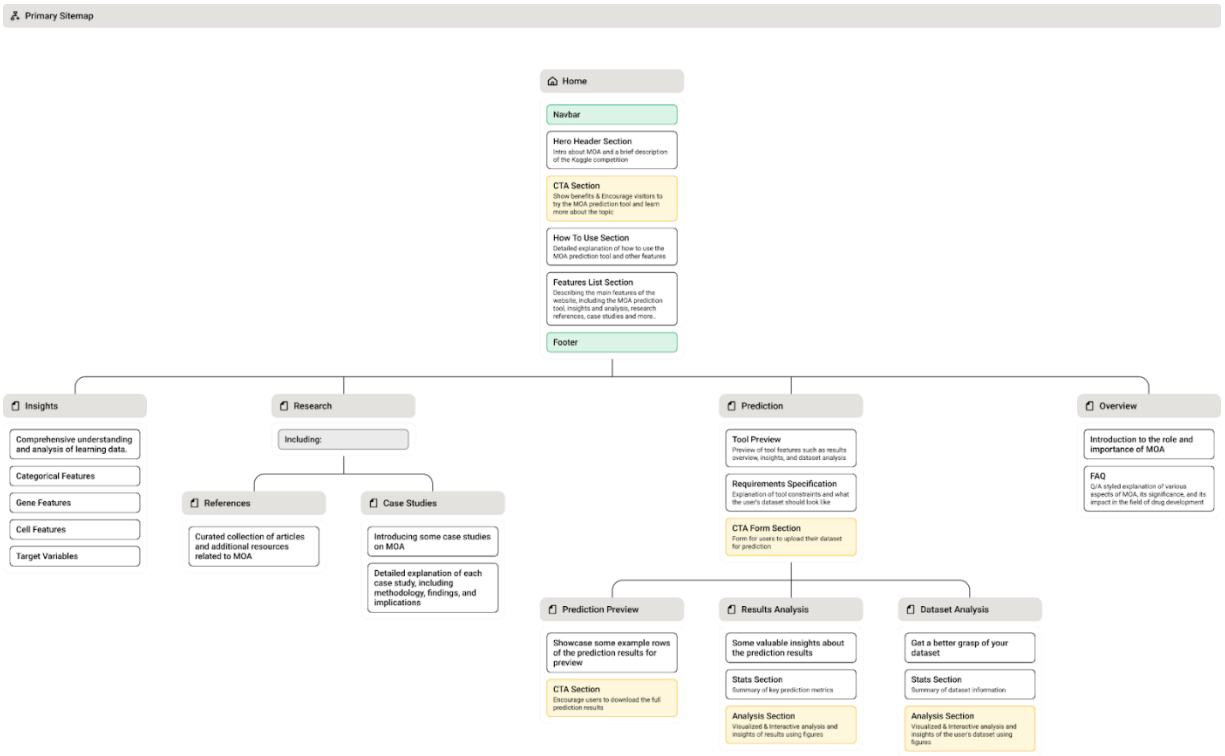
These projects illustrate various approaches to multi-label classification across different domains, including text, image, video, audio, and conversational data. By leveraging a combination of traditional machine learning models and advanced deep learning architectures, these projects achieved significant performance improvements, offering valuable insights and techniques applicable to our Mechanism of Action prediction project.

Chapter 7:

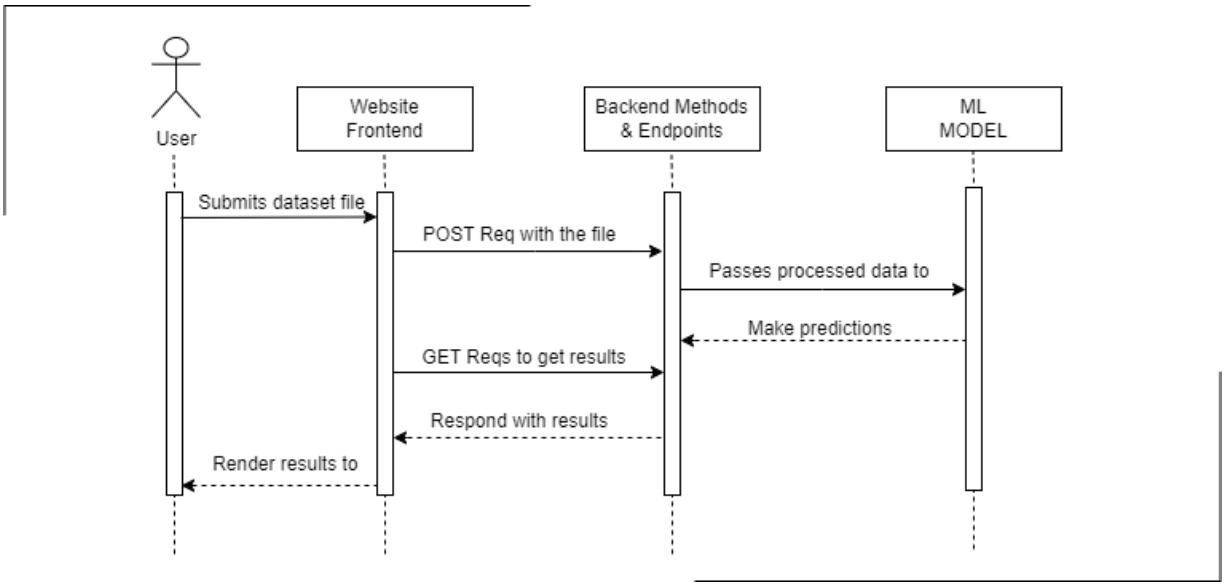
Proposed Solutions

Chapter 7: Proposed Solutions

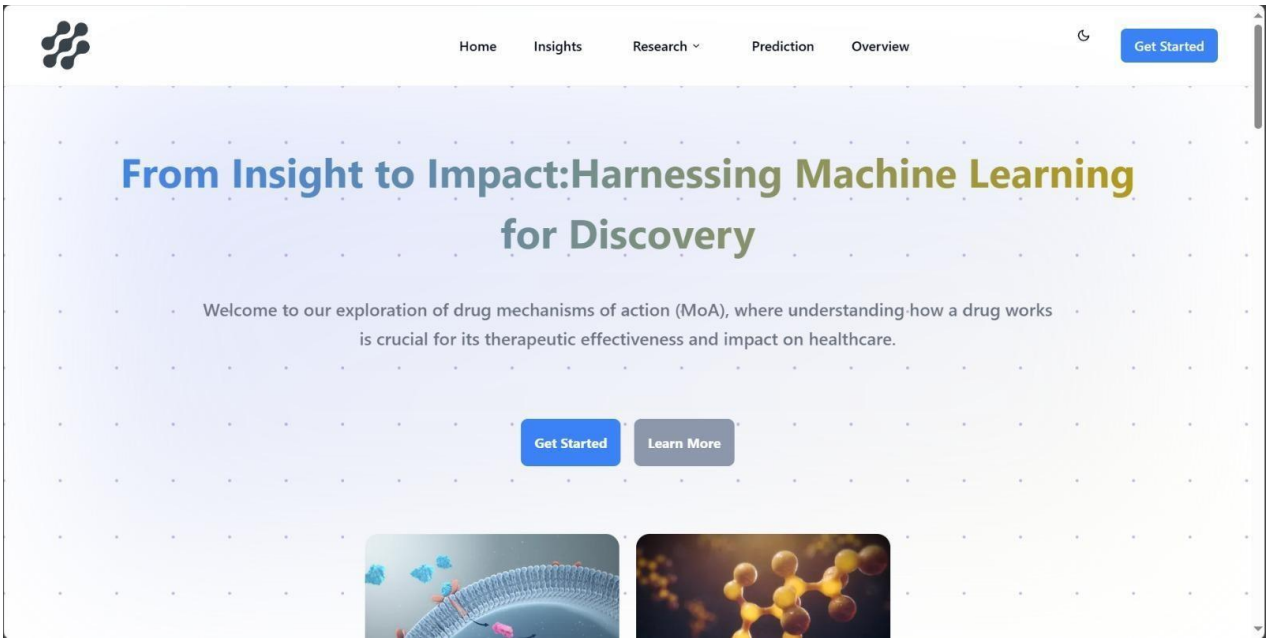
Sitemap

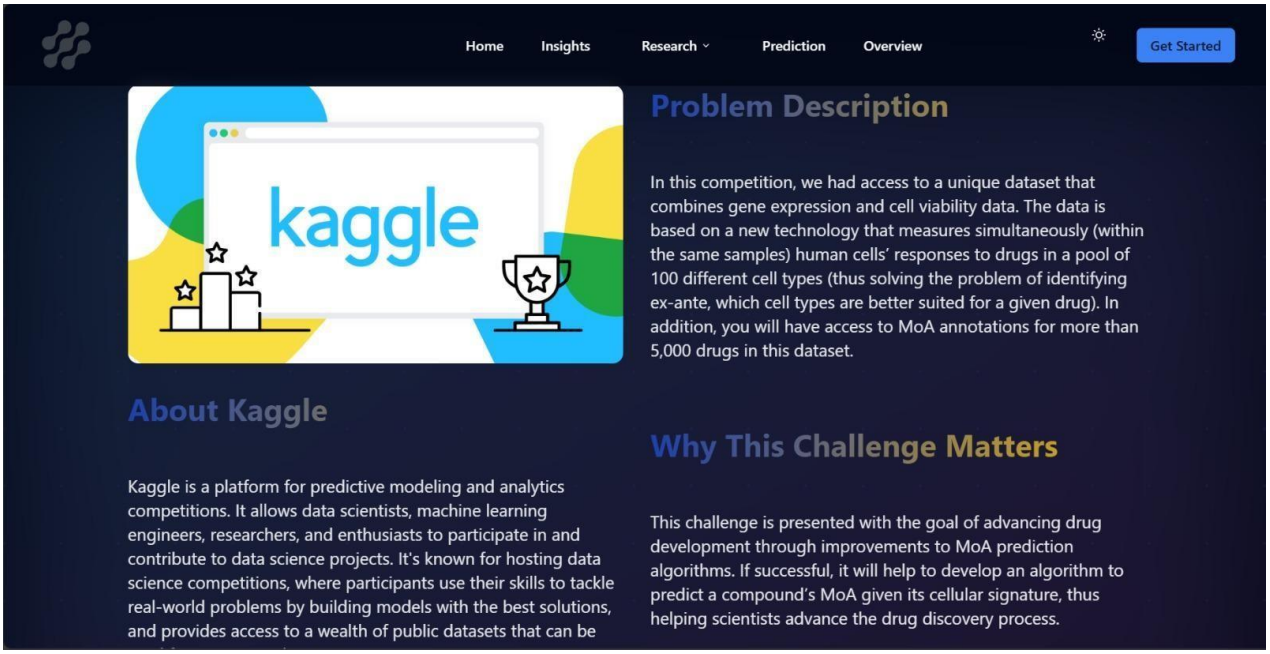


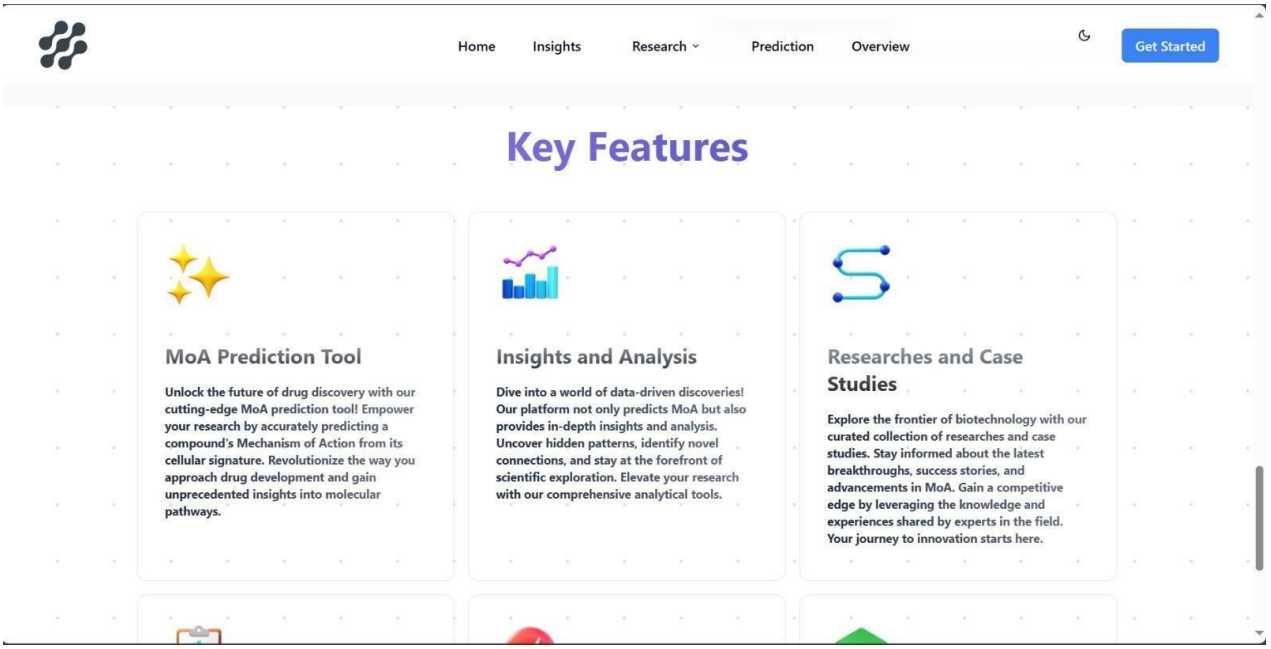
Sequence Diagram

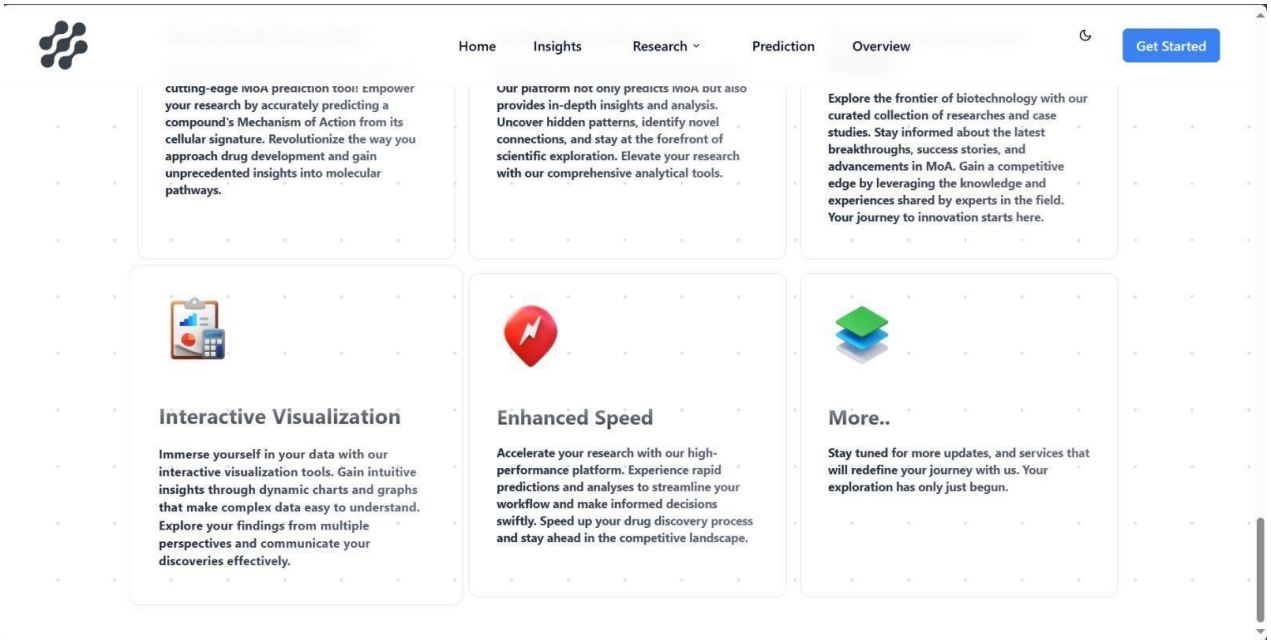


Web application wireframe

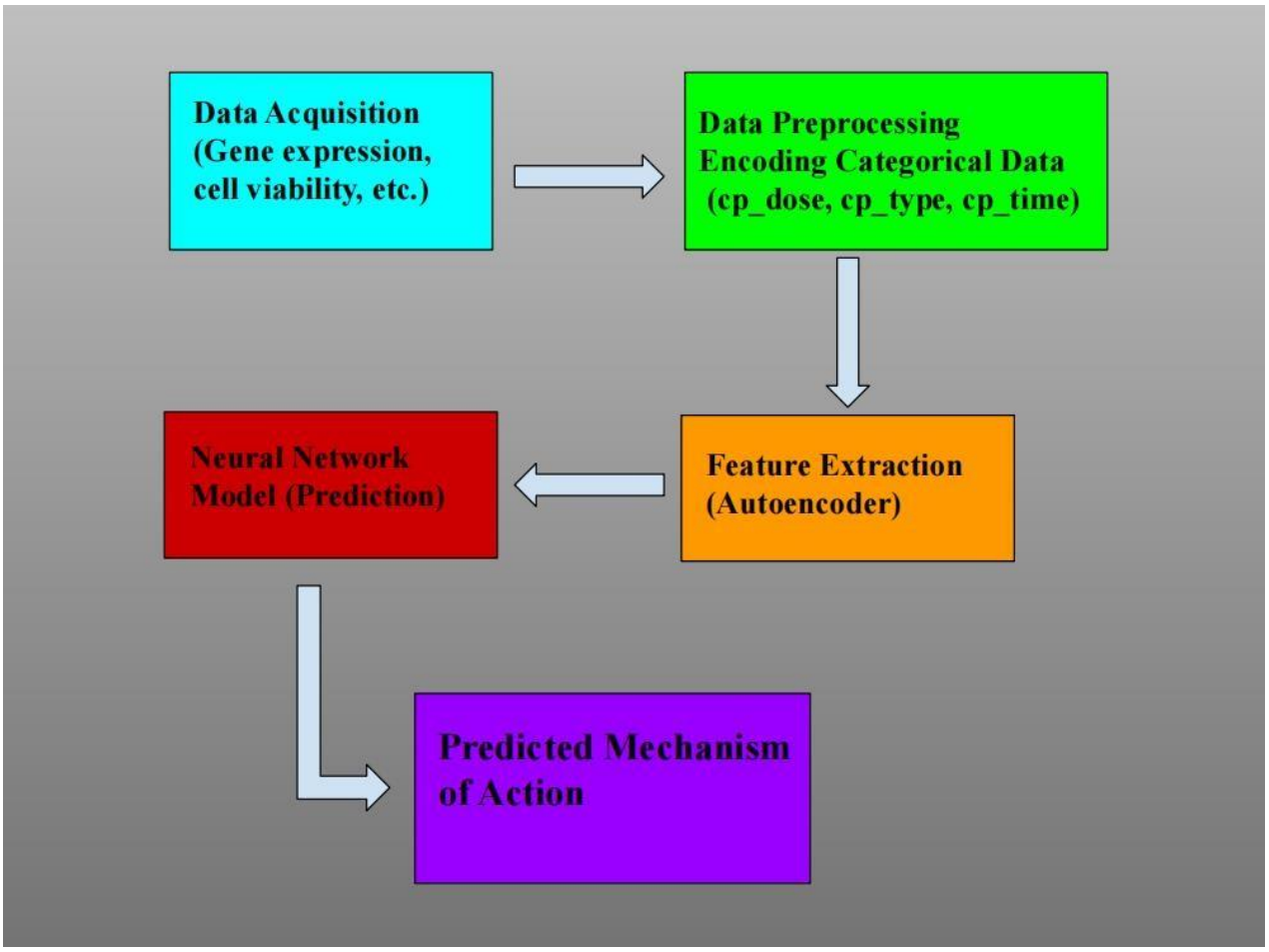





Home
Insights
Research
Prediction
Overview
Get Started
Key Features
MoA Prediction Tool
Unlock the future of drug discovery with our cutting-edge MoA prediction tool! Empower your research by accurately predicting a compound's Mechanism of Action from its cellular signature. Revolutionize the way you approach drug development and gain unprecedented insights into molecular pathways.
Insights and Analysis
Dive into a world of data-driven discoveries! Our platform not only predicts MoA but also provides in-depth insights and analysis. Uncover hidden patterns, identify novel connections, and stay at the forefront of scientific exploration. Elevate your research with our comprehensive analytical tools.
Researches and Case Studies
Explore the frontier of biotechnology with our curated collection of researches and case studies. Stay informed about the latest breakthroughs, success stories, and advancements in MoA. Gain a competitive edge by leveraging the knowledge and experiences shared by experts in the field. Your journey to innovation starts here.


Home
Insights
Research
Prediction
Overview
Get Started
Interactive Visualization
Immerse yourself in your data with our interactive visualization tools. Gain intuitive insights through dynamic charts and graphs that make complex data easy to understand. Explore your findings from multiple perspectives and communicate your discoveries effectively.
Enhanced Speed
Accelerate your research with our high-performance platform. Experience rapid predictions and analyses to streamline your workflow and make informed decisions swiftly. Speed up your drug discovery process and stay ahead in the competitive landscape.
More..
Stay tuned for more updates, and services that will redefine your journey with us. Your exploration has only just begun.

system block diagram



Machine Learning Process

The proposed solution for predicting the mechanism of action (MoA) of drugs involves several key steps, as illustrated in the system block diagram above:

1. **Data Acquisition**:

- In this initial phase, raw data is gathered, which includes gene expression profiles, cell viability metrics, and other relevant biological data.
- The dataset used for this project was obtained from the Kaggle competition "Mechanisms of Action (MoA) Prediction."

2. **Data Preprocessing**:
   - The acquired data undergoes preprocessing to ensure it is in a suitable format for analysis.
   - Specifically, categorical data such as cp_dose, cp_type, and cp_time are encoded using appropriate techniques to convert them into numerical representations.
3. **Feature Extraction**:
   - Following preprocessing, the features of the data are extracted to capture the most informative aspects.
   - An Autoencoder, a type of neural network, is employed for feature extraction. This method reduces the dimensionality of the data while retaining essential information, making it easier for the model to learn from the data.
4. **Neural Network Model**:
   - The extracted features are then fed into a neural network model designed to predict the mechanism of action.

- The model is trained on the processed dataset to learn the complex patterns and relationships between the features and the target labels (MoA).

5. **Predicted Mechanism of Action**:
   - Finally, the neural network model provides predictions on the mechanism of action of the drugs.
   - These predictions can then be utilized by researchers and pharmaceutical companies to better understand the effects of various compounds and assist in drug discovery efforts.

The combination of robust data preprocessing, advanced feature extraction techniques, and a well-tuned neural network model ensures that our approach is capable of accurately predicting the mechanism of action, thereby contributing valuable insights to the field of drug discovery and development.

Dataset

Here is the information about the datasets from the Kaggle "Mechanisms of Action (MoA) Prediction" competition, presented in a table format with columns for the name of the dataset, its structure, and detailed information.

Name	Structure	Details
`train_features.csv`	Rows: 23814, Columns: 876	Contains the features for the training set. Each row corresponds to a compound, and columns include gene expression (`g-` prefix) and cell viability (`c-` prefix) features.
`train_targets_scored.csv`	Rows: 23814, Columns: 207	Contains the target labels for the training set. Each row corresponds to a compound, with binary values indicating the presence (1) or absence (0) of each MoA.
`test_features.csv`	Rows: 3982, Columns: 876	Contains the features for the test set. Similar to `train_features.csv`, but without the target labels.
`sample_submission.csv`	Rows: 3982, Columns: 207	A sample submission file showing the required format for the predictions. Each row corresponds to a compound in the test set, with columns for each MoA.
`train_targets_nonscored.csv`	Rows: 23814, Columns: 402	Contains additional target labels that are not scored in the competition. Similar to `train_targets_scored.csv` but includes more MoA labels.

Separating Cell Features and Gene Features for Feature Extraction

First, we separate the gene and cell features from the training dataset for feature extraction using autoencoders.

```
# Separating cell features and gene features
gene_features = []
cell_features = []

# Identify columns that start with 'g-' as gene features and 'c-' as cell features
for i in train_features.columns:
    if i.startswith('g-'):
        gene_features.append(i)
    if i.startswith('c-'):
        cell_features.append(i)

print('the length of gene features', len(gene_features))
print('the length of cell features', len(cell_features))

# Merge the training features with the target scores on 'sig_id'
train_merge = pd.merge(train_features, train_targets_scored, on='sig_id', how='left')
print('the length of all inputs and target dataset', train_merge.shape)
```

This part of the code separates the gene and cell features based on their prefixes and merges the training features with the target scores for further processing.

Encoding Categorical Data

Next, we encode the categorical data (cp_type, cp_time, cp_dose) using label encoding to convert categorical labels into numerical values.

```
from sklearn.preprocessing import LabelEncoder

# Create a copy of the merged dataset
x = train_merge.copy()
le = LabelEncoder()

# Apply label encoding to categorical columns
x['cp_type'] = le.fit_transform(x['cp_type'])
x['cp_time'] = le.fit_transform(x['cp_time'])
x['cp_dose'] = le.fit_transform(x['cp_dose'])

# Apply label encoding to test features as well
test_features['cp_type'] = le.fit_transform(test_features['cp_type'])
test_features['cp_time'] = le.fit_transform(test_features['cp_time'])
test_features['cp_dose'] = le.fit_transform(test_features['cp_dose'])
```

Feature Extraction with Autoencoder

We then apply autoencoders to both gene features and cell features to reduce their dimensionality and extract significant features.

Autoencoder for Gene Features

```python
# Autoencoder for gene features
import tensorflow as tf
from tensorflow.keras.layers import Input, Dense, BatchNormalization
from tensorflow.keras import optimizers

# Set random seed for reproducibility
tf.random.set_seed(42)

# Define the autoencoder model for gene features
inputs = Input(shape=(772,))
encoder_1 = Dense(512, activation='relu')(inputs)
batch_norm = BatchNormalization()(encoder_1)
encoder_2 = Dense(420, activation='relu')(batch_norm)

# Decoder part
decoder_1 = Dense(420, activation='relu')(encoder_2)
batch_norm = BatchNormalization()(decoder_1)
decoder_2 = Dense(512, activation='relu')(batch_norm)
batch_norm = BatchNormalization()(decoder_2)
decoder_3 = Dense(772)(encoder_2)

model = tf.keras.Model(inputs=inputs, outputs=decoder_3)
model.compile(optimizer=optimizers.Adam(), loss='mse')

# Train the autoencoder
model.fit(x[gene_features], x[gene_features], batch_size=512, epochs=300, verbose=1)

# Save the encoder part of the autoencoder
encoder = tf.keras.Model(inputs=inputs, outputs=encoder_2)
encoder.save('encoders_gene_features.h5')
```

Autoencoder for Cell Features

```
# Autoencoder for cell features
tf.random.set_seed(42)

# Define the autoencoder model for cell features
inputs = Input(shape=(100,))
encoder_1 = Dense(90, activation='relu')(inputs)
batch_norm = BatchNormalization()(encoder_1)
encoder_2 = Dense(75, activation='relu')(batch_norm)

# Decoder part
decoder_1 = Dense(75, activation='relu')(encoder_2)
batch_norm = BatchNormalization()(decoder_1)
decoder_2 = Dense(90, activation='relu')(batch_norm)
batch_norm = BatchNormalization()(decoder_2)
decoder_3 = Dense(100)(encoder_2)

model = tf.keras.Model(inputs=inputs, outputs=decoder_3)
model.compile(optimizer='adam', loss='mse')

# Train the autoencoder
model.fit(x[cell_features], x[cell_features], batch_size=512, epochs=300, verbose=1)

# Save the encoder part of the autoencoder
encoder = tf.keras.Model(inputs=inputs, outputs=encoder_2)
encoder.save('encoders_cell_features.h5')
```

Load the Models and Apply Feature Extraction

```
from tensorflow.keras.models import load_model

# Load and apply autoencoder for gene features
encoder = load_model('encoders_gene_features.h5')
train_gene_features = encoder.predict(x[gene_features])
test_gene_features = encoder.predict(test_features[gene_features])

# Load and apply autoencoder for cell features
encoder = load_model('encoders_cell_features.h5')
train_cell_features = encoder.predict(x[cell_features])
test_cell_features = encoder.predict(test_features[cell_features])
```

```
# Combine the data after encoding and feature extraction
x_1_train = np.hstack((x['cp_type'].values.reshape(-1, 1),
                       x['cp_time'].values.reshape(-1, 1),
                       x['cp_dose'].values.reshape(-1, 1),
                       train_gene_features,
                       train_cell_features))

x_1_test = np.hstack((test_features['cp_type'].values.reshape(-1, 1),
                      test_features['cp_time'].values.reshape(-1, 1),
                      test_features['cp_dose'].values.reshape(-1, 1),
                      test_gene_features,
                      test_cell_features))

print(x_1_train.shape, x_1_test.shape)
```

Neural Network Model for Prediction

Finally, we create and train a neural network model to predict the mechanism of action of drugs.

This code defines, compiles, and trains a neural network model using cross-validation. The model predicts the mechanism of action of drugs based on the combined features from the autoencoders and encoded categorical data.

```python
# Create a Neural Network Model

num_of_labels = y.shape[1]
num_test_samples = test_features.shape[0]
y = y.to_numpy()

# Training settings
n_seeds = 3
n_folds = 7
hists = []
oof = tf.constant(0.0)
y_pred = np.zeros((num_test_samples, num_of_labels))
bias = tf.keras.initializers.constant(np.log(y.mean(axis=0)))

# Training loop
seeds = [34, 9, 18]
for seed in seeds:
    fold = 0
    stratifier = IterativeStratification(n_splits=7, order=1)
    for train_idx, test_idx in stratifier.split(x_1_train, y):
        X_train = x_1_train[train_idx]
        X_test = x_1_train[test_idx]
        y_train = y[train_idx]
        y_test = y[test_idx]

        # Define the neural network model
        model = Sequential([
            Dropout(0.3),
            Dense(2048, activation='relu'),
            BatchNormalization(),
            Dropout(0.5),
            Dense(1024, activation='relu'),
            BatchNormalization(),
            Dropout(0.5),
            Dense(512, activation='relu'),
            BatchNormalization(),
            Dropout(0.3),
            Dense(num_of_labels, activation='sigmoid', bias_initializer=bias)
        ])
        # Compile the model
        model.compile(optimizer=optimizers.Adam(learning_rate=1e-5),
                      loss=losses.BinaryCrossentropy(label_smoothing=0.001),
                      metrics=['binary_crossentropy', logloss])
        early_stopping = callbacks.EarlyStopping(monitor='val_logloss', patience=5, mode='min')

        def scheduler(epoch, lr):
            if epoch % 16 < 9:
                lr += np.exp(-int(epoch / 16)) * (0.5 * 1e-2) / 16
            else:
                lr -= np.exp(-int(epoch / 16)) * (0.5 * 1e-2) / 16
            return lr
        lr_scheduler = callbacks.LearningRateScheduler(scheduler)

        # Train the model
        history = model.fit(X_train, y_train, batch_size=128, epochs=192, verbose=1,
                            validation_data=(X_test, y_test), callbacks=[lr_scheduler, early_stopping])

        # Compile and fit the model again without the scheduler
        model.compile(optimizer=optimizers.Adam(learning_rate=1e-5),
                      loss=losses.BinaryCrossentropy(label_smoothing=0.001),
                      metrics=['binary_crossentropy', logloss])
        model.fit(X_train, y_train, batch_size=128, epochs=192, verbose=1,
                  validation_data=(X_test, y_test), callbacks=[early_stopping])
        hists.append(history)

        # Save the model
        model.save(f'AutoEncoded_seed_{seed}_fold_{fold}.h5')
        y_valid = model.predict(X_test)
        oof += logloss(tf.constant(y_test, dtype=tf.float32),
                       tf.constant(y_valid, dtype=tf.float32)) / (n_folds * n_seeds)
        y_pred += model.predict(x_1_test) / (n_folds * n_seeds)
        fold += 1
```

Generate the Final Submission File

After training the model, we generate the final predictions and save them to a CSV file for submission.

```
p_min = 0.0005
p_max = 0.9995

# Generate submission file, Clip Predictions
sub = pd.read_csv('/kaggle/input/lish-moa/sample_submission.csv')
sub.iloc[:, 1:] = np.clip(y_pred, p_min, p_max)

# Save Submission
sub.to_csv('submission.csv', index=False)
sub.head()
```

This final part reads the sample submission file, updates it with the model's predictions, clips the predictions to ensure they fall within the specified range, and saves the final submission file.

By organizing and illustrating each part of the code, we provide a clear and detailed documentation of the proposed solution for predicting the mechanism of action of drugs using autoencoders and neural networks.

Model Deployment

```python
app = Flask(__name__)
CORS(app)

encoderGene = None
encoderCell = None
sub = None
preview_data = None
dataset = None
dataset_details = None

p_min = 0.0005
p_max = 0.9995
prediction = pd.DataFrame()

def visualizeData(name_column):
    global dataset

    gene_features = []
    cell_features = []
    for i in dataset.columns:
        if i.startswith('g-'):
            gene_features.append(i)
        if i.startswith('c-'):
            cell_features.append(i)

    if name_column == 'cp_type':
        cp_type_percentages = (dataset['cp_type'].value_counts()*100.0 /len(dataset))

        colors = ['#1f77b4', '#ff7f0e']
        data=[
            go.Bar(name='cp_type', x=cp_type_percentages.index, y=cp_type_percentages.values,
                marker_color=colors, text=[f'{val:.1f}%' for val in cp_type_percentages.values],
                textposition='auto')
        ]
        fig = go.Figure(data)
        fig.update_layout(
            title_text='Type of Treatment',
            xaxis_title='cp_type',
            yaxis_title='% Drug',
            barmode='stack'
        )
        return to_json(fig, engine="orjson")

    elif name_column == 'cp_time':
        cp_time_percentages = dataset['cp_time'].value_counts()*100.0 /len(dataset)

        colors = ['#1f77b4', '#ff7f0e', '#2ca02c']
        data=[
            go.Bar(name='cp_time', x=cp_time_percentages.index, y=cp_time_percentages.values,
                text=[f'{val:.2f}%' for val in cp_time_percentages.values],
                textposition='auto', marker_color=colors)
        ]
        fig = go.Figure(data)
        fig.update_layout(
            title_text='Time Duration of Treatment',
            xaxis_title='Time',
            yaxis_title='% Treatment',
            barmode='stack'
        )
        return to_json(fig, engine="orjson")

    elif name_column == 'cp_dose':
        cp_dose_percentages = dataset['cp_dose'].value_counts()*100.0 /len(dataset)

        colors = ['#1f77b4', '#ff7f0e']
        data=[
            go.Bar(name='cp_dose', x=cp_dose_percentages.index, y=cp_dose_percentages.values,
                text=[f'{val:.2f}%' for val in cp_dose_percentages.values],
                textposition='auto', marker_color=colors)
        ]
        fig = go.Figure(data)
        fig.update_layout(
            title_text='Doses of Drugs',
            xaxis_title='Dose',
            yaxis_title='% Treatment',
            barmode='stack'
        )
        return to_json(fig, engine="orjson")
```

```
    elif name_column == 'gene_expression':
        data_list = [dataset[feature] for feature in gene_features]

        colors = ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd', '#8c564b', '#e377c2',
'#7f7f7f', '#bcbd22', '#17becf']
        fig = go.Figure()
        for i, feature in enumerate(gene_features):
            fig.add_trace(go.Histogram(x=data_list[i], name=feature, marker_color=colors[i %
len(colors)]))
        fig.update_layout(
            title_text='Distribution of all Gene Features',
            xaxis_title_text='Value',
            yaxis_title_text='Count',
            bargap=0.2,
            bargroupgap=0.1,
            barmode='overlay',
        )
        fig.update_traces(opacity=0.75)
        return to_json(fig, engine="orjson")

    elif name_column == 'cell_viability':
        data_list = [dataset[feature] for feature in cell_features]

        colors = ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd', '#8c564b', '#e377c2',
'#7f7f7f', '#bcbd22', '#17becf']
        fig = go.Figure()
        for i, feature in enumerate(cell_features):
            fig.add_trace(go.Histogram(x = data_list[i], name = feature, marker_color = colors[i %
len(colors)]))
        fig.update_layout(
            title_text='Distribution of all Cell Features',
            xaxis_title_text='Value',
            yaxis_title_text='Count',
            bargap=0.2,
            bargroupgap=0.1,
            barmode='overlay',
        )
        fig.update_traces(opacity=0.75)
        return to_json(fig, engine="orjson")

def generate_top_20(pred_results):
    global prediction

    x_axis = list(prediction.columns.values)
    sig_id_values = x_axis[1:]
    count_of_target = prediction.iloc[:, 1:].sum().values
    dct = dict(zip(sig_id_values, count_of_target))
    sorted_dict = dict(sorted(dct.items(), key=lambda i: i[1], reverse=True))

    num_bars = 20
    color_palette = [
        f'rgb(0, 0, {int(255 * (i / num_bars))})' for i in range(num_bars)
    ]

    data=[
        go.Bar(y=list(sorted_dict.keys())[:num_bars], x=list(sorted_dict.values())[:num_bars],
orientation='h', marker_color = color_palette)
    ]
    fig1 = go.Figure(data)
    fig1.update_layout(
        yaxis=dict(autorange="reversed"),
        autosize=False,
        width=900,
        height=600,
        plot_bgcolor='white',
        paper_bgcolor='rgb(247, 250, 252)',
        # font=dict(color= color)
    )

    return to_json(fig1, engine="orjson")

def generate_lowest_20(pred_results):
    global prediction

    x_axis = list(prediction.columns.values)
    sig_id_values = x_axis[1:]
    count_of_target = prediction.iloc[:, 1:].sum().values
    dct = dict(zip(sig_id_values, count_of_target))
    sorted_dict = dict(sorted(dct.items(), key=lambda i: i[1], reverse=True))

    num_bars = 20
    color_palette = [
        f'rgb(0, 0, {int(255 * (i / num_bars))})' for i in range(num_bars)
    ]

    data=[
        go.Bar(y=list(sorted_dict.keys())[-num_bars:], x=list(sorted_dict.values())[-num_bars:],
orientation='h', marker_color = color_palette)
    ]
    fig2 = go.Figure(data)
    fig2.update_layout(
        yaxis=dict(autorange="reversed"),
        autosize=False,
        width=900,
        height=600,
        plot_bgcolor='white',
        paper_bgcolor='rgb(247, 250, 252)',
    )

    return to_json(fig2, engine="orjson")
```

```python
def load_models():
    global encoderGene, encoderCell, sub
    if encoderGene is None:
        encoderGene = load_model('./models/NN/encoders_gene_features.h5')
    if encoderCell is None:
        encoderCell = load_model('./models/NN/encoders_cell_features.h5')
    if sub is None:
        sub = pd.read_csv('./models/NN/sample_submission.csv')

def preprocessData(inputData):
    load_models()

    le = LabelEncoder()
    inputData['cp_type'] = le.fit_transform(inputData['cp_type'])
    inputData['cp_time'] = le.fit_transform(inputData['cp_time'])
    inputData['cp_dose'] = le.fit_transform(inputData['cp_dose'])

    gene_features = []
    cell_features = []
    for i in inputData.columns:
        if i.startswith('g-'):
            gene_features.append(i)
        if i.startswith('c-'):
            cell_features.append(i)

    test_gene_features = encoderGene.predict(inputData[gene_features])
    test_cell_features = encoderCell.predict(inputData[cell_features])

    x_1_test = np.hstack((inputData['cp_type'].values.reshape(-1,1),
inputData['cp_time'].values.reshape(-1, 1), inputData['cp_dose'].values.reshape(-1, 1),
                          test_gene_features, test_cell_features))

    return x_1_test

@app.route('/upload', methods=['POST'])
def upload_file():
    global preview_data, dataset_details, dataset, prediction, p_max, p_min
    if request.method == 'POST':
        file = request.files['file']

    if not file:
        print("No file 🚩🚩🚩🚩")
        return "No file"

    dataset_details = [
        {"Item": "Dataset Name", "Value": file.filename},
        {"Item": "Data Source", "Value": "CSV File"},
    ]

    data = pd.read_csv(file, delimiter=',')
    dataset = data.copy()

    dataset_details.extend([
        {"Item": "Total Rows", "Value": data.shape[0]},
        {"Item": "Total Columns", "Value": data.shape[1]},
        {"Item": "Total Data Points", "Value": data.shape[0] * data.shape[1]}
    ])

    start_time = time.time()

    new_data = preprocessData(data)

    directory = r'.\models\NN\seed'
    file_paths = []

    for filename in os.listdir(directory):
        file_path = os.path.join(directory, filename)
        file_paths.append(file_path)

    y_pred = np.zeros((new_data.shape[0], 206))
    for i in file_paths:
        model = load_model(i)
        y_pred += model.predict(new_data) / (21)

    print('DONE PREDICTION')
    end_time = time.time()
    print('TIME TAKEN TO PREDICT IS : {}'.format(end_time - start_time))

    sub.iloc[ : new_data.shape[0],1:] = np.clip(y_pred,p_min,p_max)
    prediction = sub.iloc[ : new_data.shape[0], : ]
    prediction['sig_id'] = data['sig_id']

    # Storing preview data temporarily
    preview_data = {
        "data": prediction.to_dict(orient='records'),
        "columns": list(prediction.columns)  # Include the column order
    }

    return jsonify({"file_ready": True})

@app.route('/dataset_details', methods=['GET'])
def get_dataset_details():
    global dataset_details
    if dataset_details is None:
        return jsonify({"message": "No dataset details available"})
    else:
        return jsonify(dataset_details)
```

```python
@app.route('/download', methods=['GET'])
def download_file():
    global prediction
    csv_output = StringIO()
    prediction.to_csv(csv_output, index=False)
    csv_output.seek(0)

    return Response(
        csv_output.getvalue(),
        mimetype="text/csv",
        headers={"Content-disposition": "attachment; filename=prediction.csv"}
    )

@app.route('/preview', methods=['GET'])
def get_preview_data():
    global preview_data
    if preview_data is None:
        return jsonify({"message": "No preview data available"})
    else:
        return jsonify(preview_data)

@app.route('/top_20_json', methods=['GET'])
def get_top_20_json():
    global prediction
    json_data = generate_top_20(prediction)
    return jsonify({"json": json_data})

@app.route('/lowest_20_json', methods=['GET'])
def get_lowest_20_json():
    global prediction
    json_data = generate_lowest_20(prediction)
    return jsonify({"json": json_data})

@app.route('/visualize', methods=['GET'])
def visualize():
    name_column = request.args.get('name_column')
    graph_json = None
    graph_json = visualizeData(name_column)
    if graph_json:
        return jsonify({"json": graph_json})
    else:
        return jsonify({'error': 'Column not found'}), 404

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

Implementation

Prediction Page Flow:

1. File Input: Users select a CSV file using an input element.

Validation: The frontend checks if the file meets the requirements (size and columns).

1. Form Submission: On form submission, the file is sent to the backend using Axios.
2. Loading States: A loading state with a stepper is shown to the user while the backend processes the file.
3. Redirection: On successful processing, the user is redirected to the Results Page.

Results Page Flow:

1. Data Fetching: On component mount, the frontend fetches preview data and dataset details from the backend.
2. Tab Navigation: Users can navigate between the Preview, Insights, and Data tabs.
3. Insights and Data Visualization: The frontend fetches additional data for the Insights and Data tabs when these tabs are active.

4. Download: Users can download the full prediction results by clicking the download button, which triggers a request to the backend.

Request/Response Cycle:

1. Prediction Request:
    a. Request: A POST request is made to /upload with the CSV file.
    b. Processing: The backend processes the file, makes predictions, and stores the results.
    c. Response: A JSON response indicating whether the file processing was successful.

1. Preview Data Request:
    a. Request: A GET request is made to /preview.
    b. Processing: The backend retrieves the preview data from the stored results.
    c. Response: A JSON response with the preview data and column names.

1. Dataset Details Request:
    a. Request: A GET request is made to /dataset_details.

b. Processing: The backend retrieves and formats the dataset details.

c. Response: A JSON response with the dataset details.

1. Download Request:

a. Request: A GET request is made to /download.

b. Processing: The backend retrieves the full prediction results and sends them as a CSV file.

c. Response: The CSV file is sent as a downloadable attachment.

1. Visualization Request:

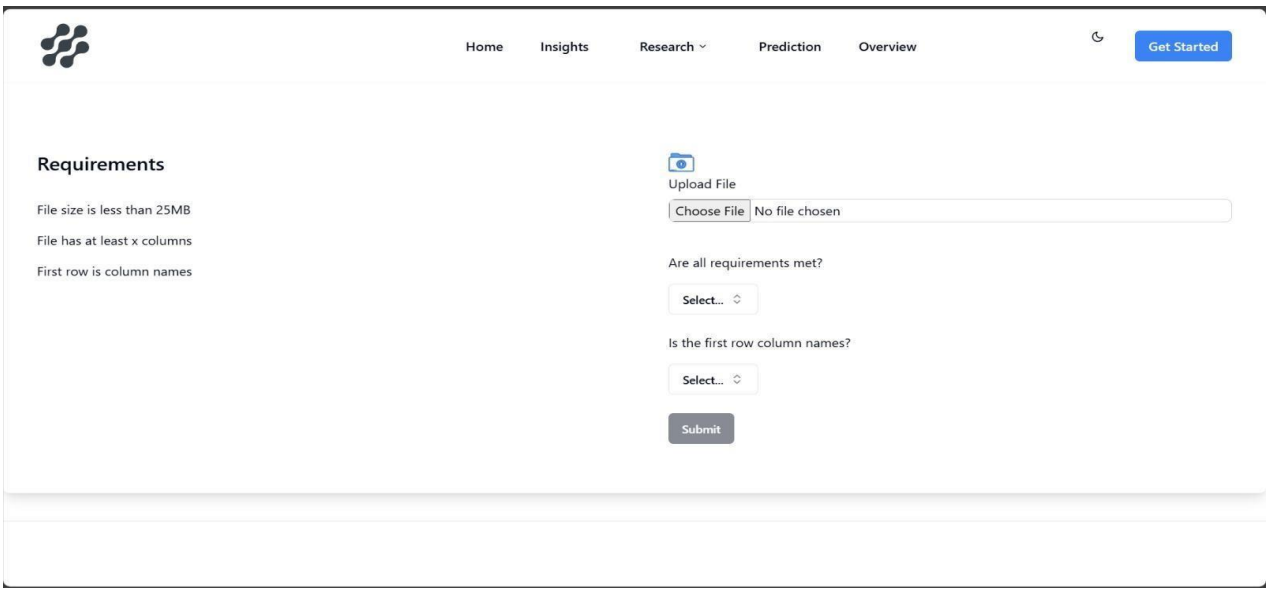
a. Request: A GET request is made to /visualize with a query parameter specifying the column name.

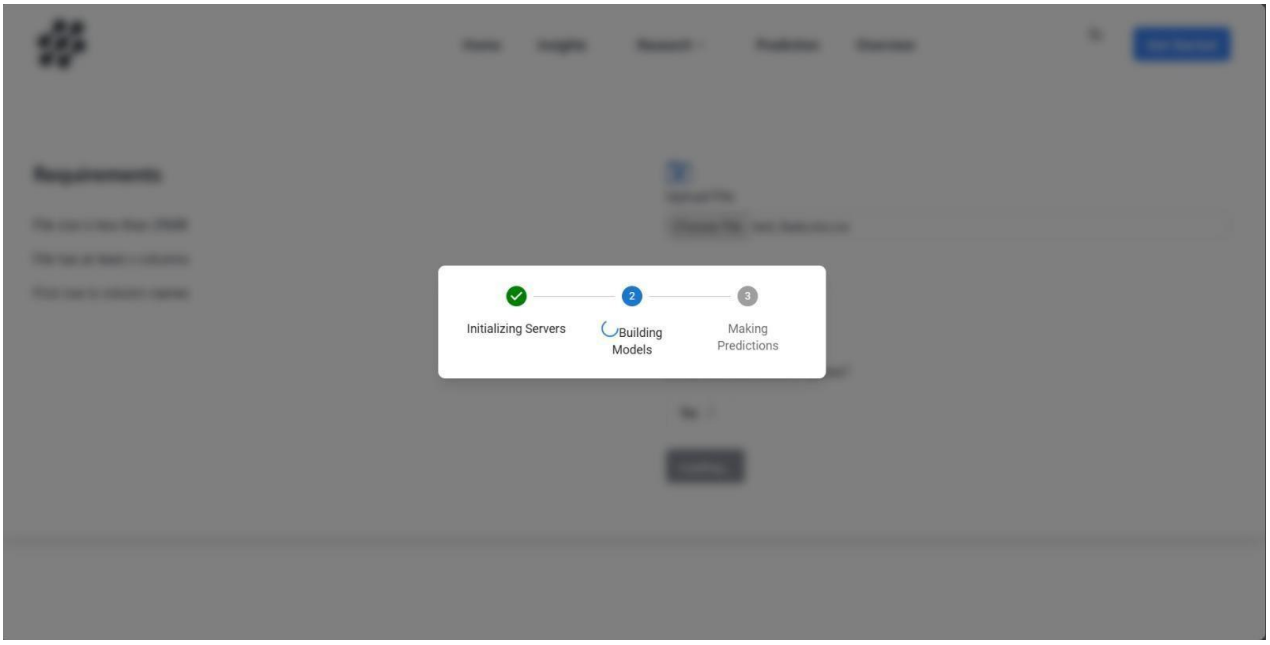
b. Processing: The backend generates a visualization for the specified column and returns it as JSON.

c. Response: A JSON response with the visualization data.

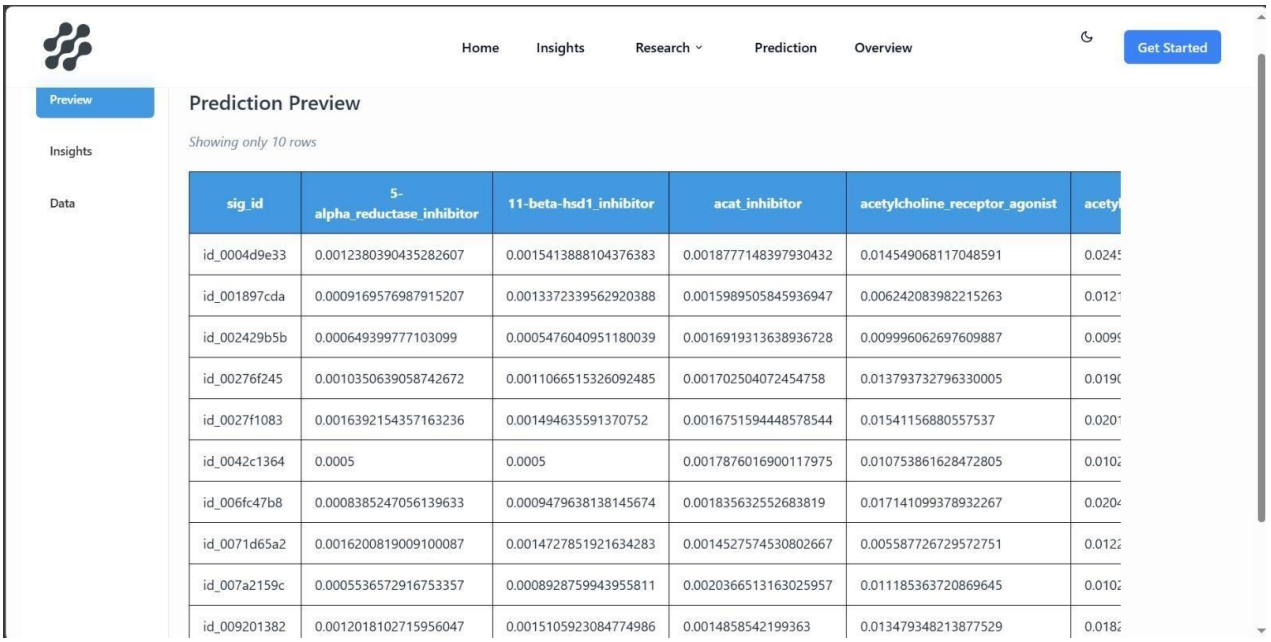
Screens

Prediction page



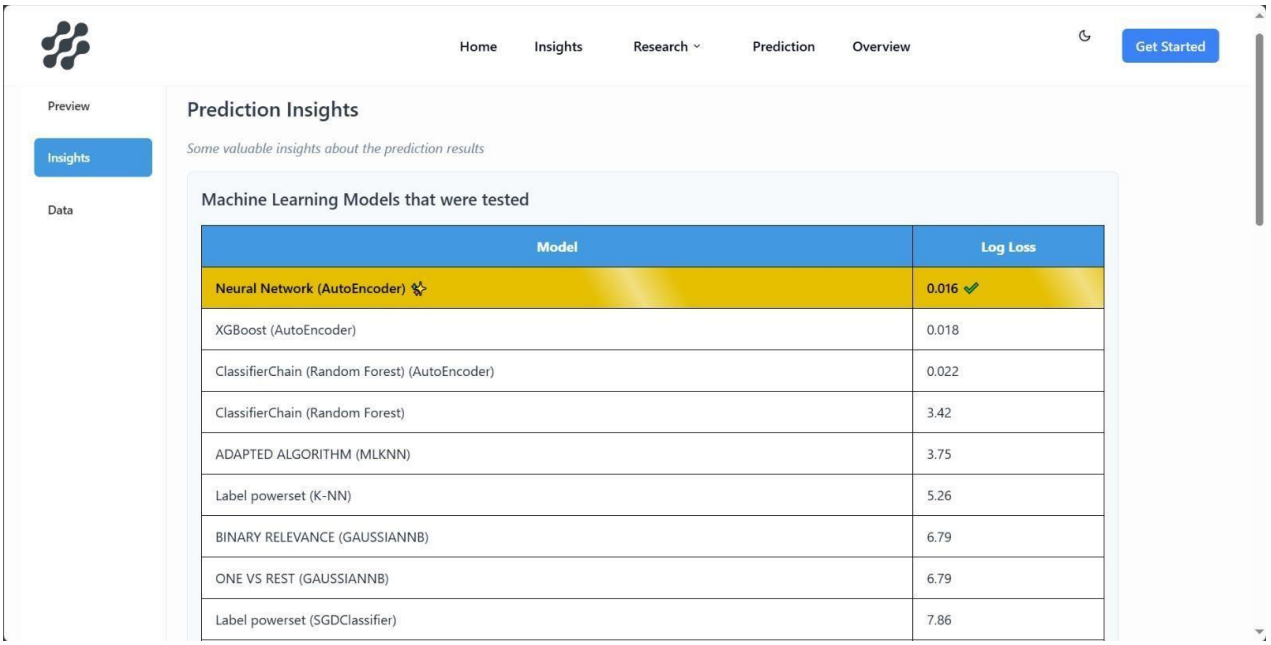


Prediction Preview



sig_id	5-alpha_reductase_inhibitor	11-beta-hsd1_inhibitor	acat_inhibitor	acetylcholine_receptor_agonist	acety
id_0004d9e33	0.0012380390435282607	0.0015413888104376383	0.0018777148397930432	0.014549068117048591	0.0245
id_001897cda	0.0009169576987915207	0.0013372339562920388	0.0015989505845936947	0.006242083982215263	0.0121
id_002429b5b	0.000649399777103099	0.0005476040951180039	0.0016919313638936728	0.009996062697609887	0.0095
id_00276f245	0.0010350639058742672	0.0011066515326092485	0.001702504072454758	0.013793732796330005	0.0190
id_0027f1083	0.0016392154357163236	0.001494635591370752	0.0016751594448578544	0.01541156880557537	0.0201
id_0042c1364	0.0005	0.0005	0.0017876016900117975	0.010753861628472805	0.0102
id_006fc47b8	0.0008385247056139633	0.0009479638138145674	0.001835632552683819	0.017141099378932267	0.0204
id_0071d65a2	0.0016200819009100087	0.0014727851921634283	0.0014527574530802667	0.005587726729572751	0.0122
id_007a2159c	0.0005536572916753357	0.0008928759943955811	0.0020366513163025957	0.011185363720869645	0.0102
id_009201382	0.0012018102715956047	0.0015105923084774986	0.0014858542199363	0.013479348213877529	0.0182

Prediction Insights

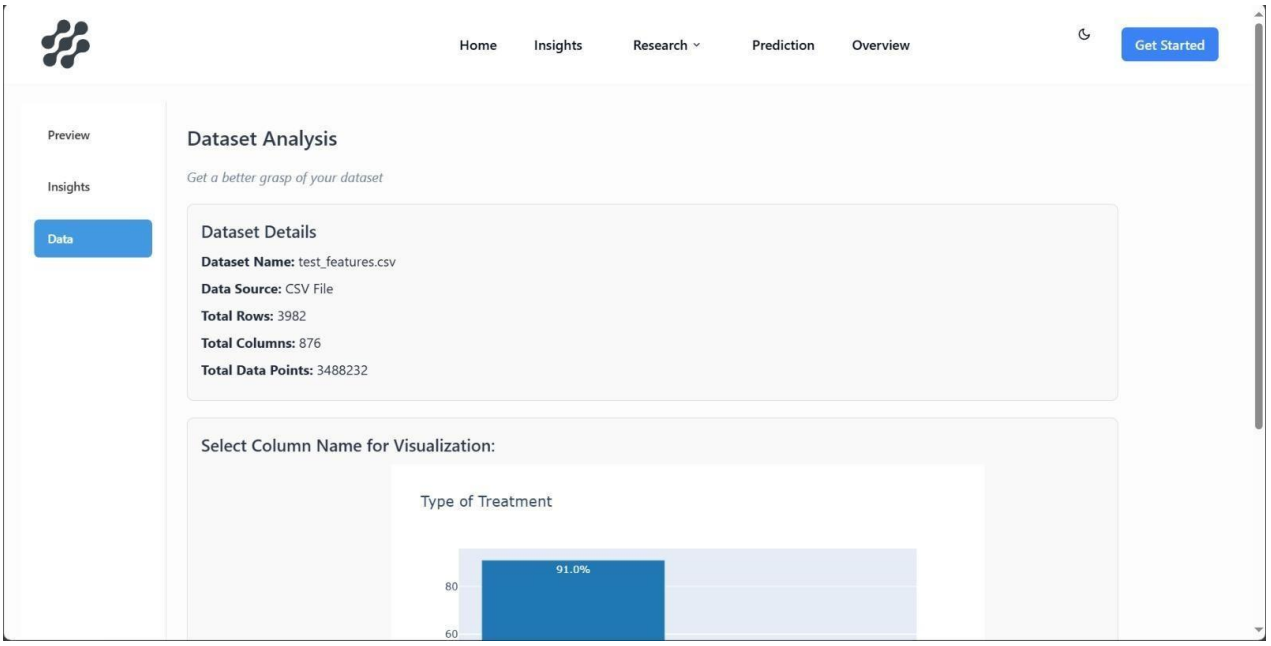


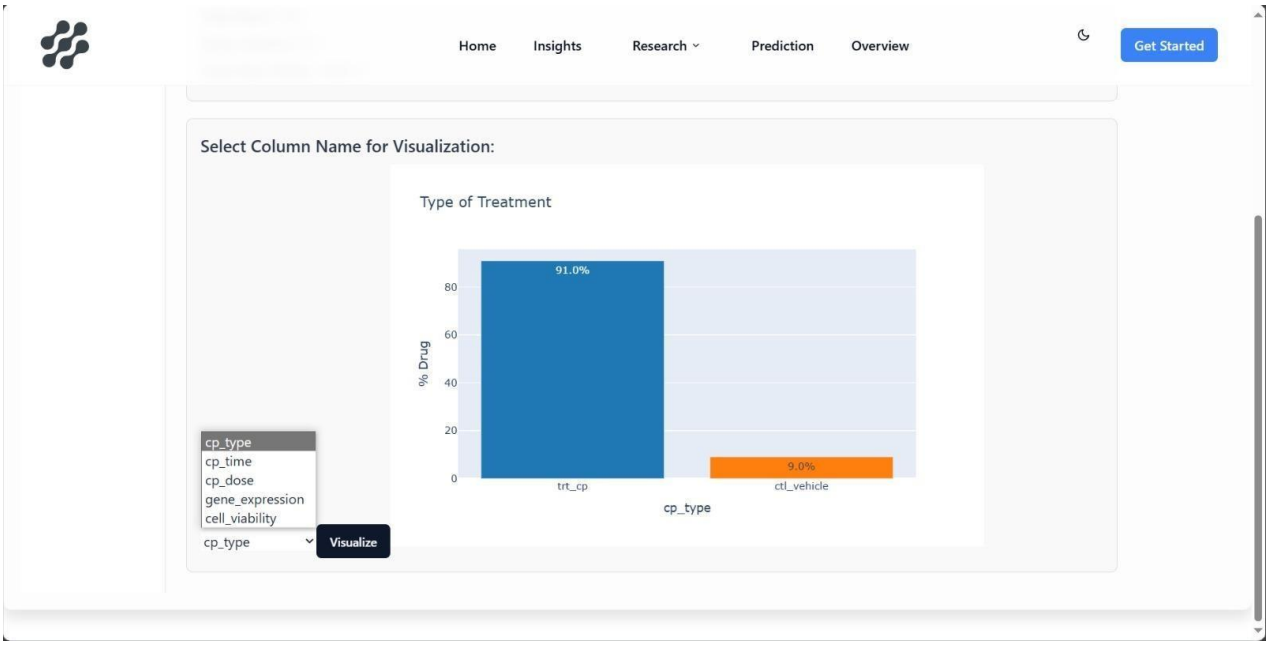
Model	Log Loss
Neural Network (AutoEncoder)	0.016
XGBoost (AutoEncoder)	0.018
ClassifierChain (Random Forest) (AutoEncoder)	0.022
ClassifierChain (Random Forest)	3.42
ADAPTED ALGORITHM (MLKNN)	3.75
Label powerset (K-NN)	5.26
BINARY RELEVANCE (GAUSSIANNB)	6.79
ONE VS REST (GAUSSIANNB)	6.79
Label powerset (SGDClassifier)	7.86


Home
Insights
Research
Prediction
Overview
Get Started
Top 20 Targets
nfkb_inhibitor
proteasome_inhibitor
cyclooxygenase_inhibitor
dopamine_receptor_antagonist
serotonin_receptor_antagonist
glutamate_receptor_antagonist
adrenergic_receptor_antagonist
dna_inhibitor
acetylcholine_receptor_antagonist
tubulin_inhibitor
adrenergic_receptor_agonist
calcium_channel_blocker
phosphodiesterase_inhibitor
sodium_channel_inhibitor
serotonin_receptor_agonist
histamine_receptor_antagonist
glucocorticoid_receptor_agonist
acetylcholine_receptor_agonist
bacterial_cell_wall_synthesis_inhibitor
egfr_inhibitor
0
50
100
150


Home
Insights
Research
Prediction
Overview
Get Started
Lowest 20 Targets
elastase_inhibitor
autotaxin_inhibitor
steroid
coagulation_factor_inhibitor
calcineurin_inhibitor
tropomyosin_receptor_kinase_inhibitor
ubiquitin_specific_protease_inhibitor
retinoid_receptor_antagonist
antiarrhythmic
nicotinic_receptor_agonist
diuretic
atm_kinase_inhibitor
laxative
nitric_oxide_production_inhibitor
aldehyde_dehydrogenase_inhibitor
lxr_agonist
protein_phosphatase_inhibitor
bacterial_membrane_integrity_inhibitor
erbb2_inhibitor
atp-sensitive_potassium_channel_antagonist
0
0.5
1
1.5
2
2.5
3

Dataset Analysis





Chapter 8:

Data Analysis & visualization

8.1 Introduction

Overview of Drug Data Analysis and Visualization

- Our project focuses on analyzing a comprehensive dataset of drugs, aiming to uncover key insights and patterns. Analyzing drug data is crucial as it helps identify the mechanisms of action, optimize dosage timings, and improve drug efficacy and safety. This analysis is essential for enhancing patient care and guiding future pharmaceutical research.

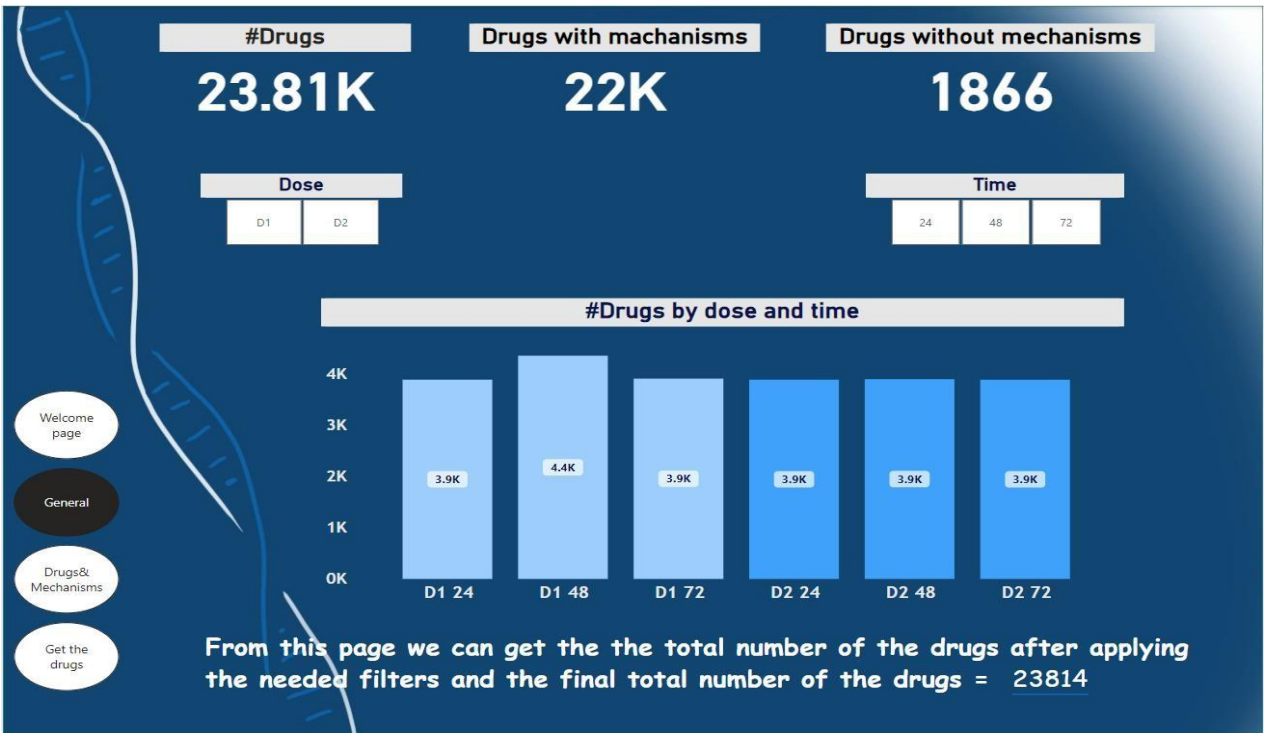
- We utilized PowerBI as our primary tool for data visualization due to its robust capabilities in handling large datasets and creating interactive and dynamic visual representations. PowerBI enables us to effectively present our findings, making complex data more accessible and understandable for stakeholders.

In the next points , we will show you some analysis and visualizations of the data we used in our project using powerbi tool .

8.2 General Overview

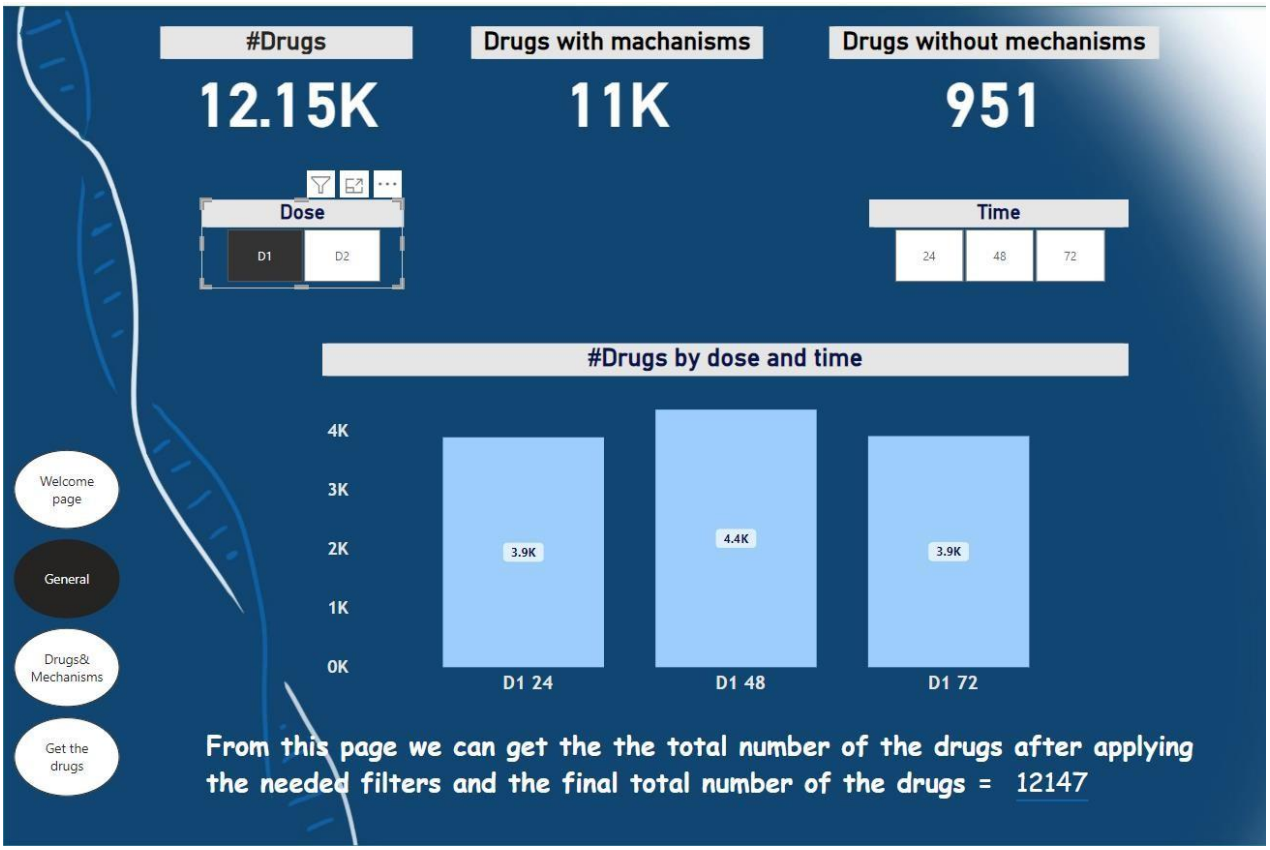
General Statistics

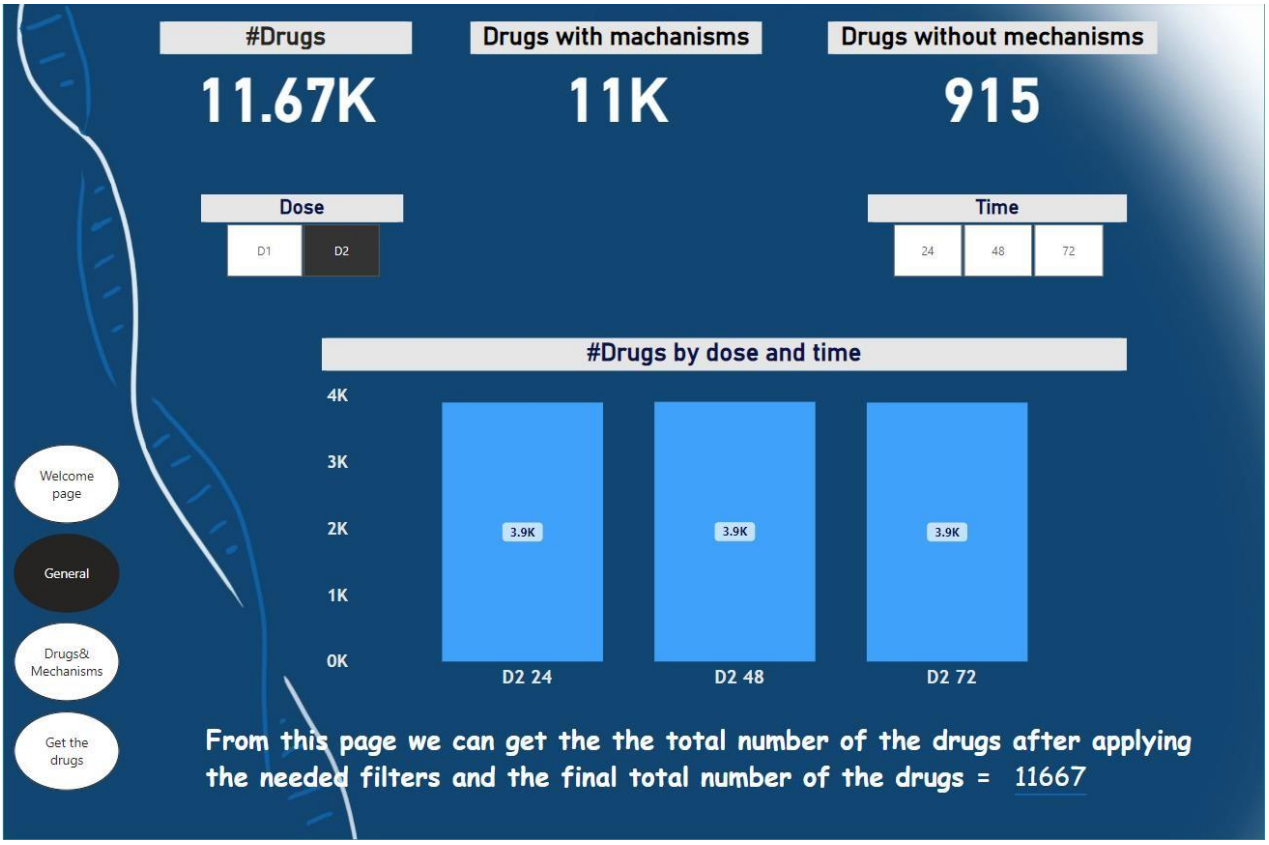
-In the first page with name "General" , we show the total number of the drugs which equal 23.18 k drugs , the total number of the drugs which have at least on mechanism of action which equal 22 k drugs and the total number of the drugs which have no mechanism of action which equal 1866 drugs . we also use a filter based on the dose either one dose or 2 doses which help us to specify the drugs we want to show , we also use a filter based on the timing of the doses either each 24 hours , 48 hours or 72 hours . It helps us to specify the drugs we want to show . we use a column chart to show the number of drugs with doses and timing of doses. In the next page we will show you the visualization related to it .

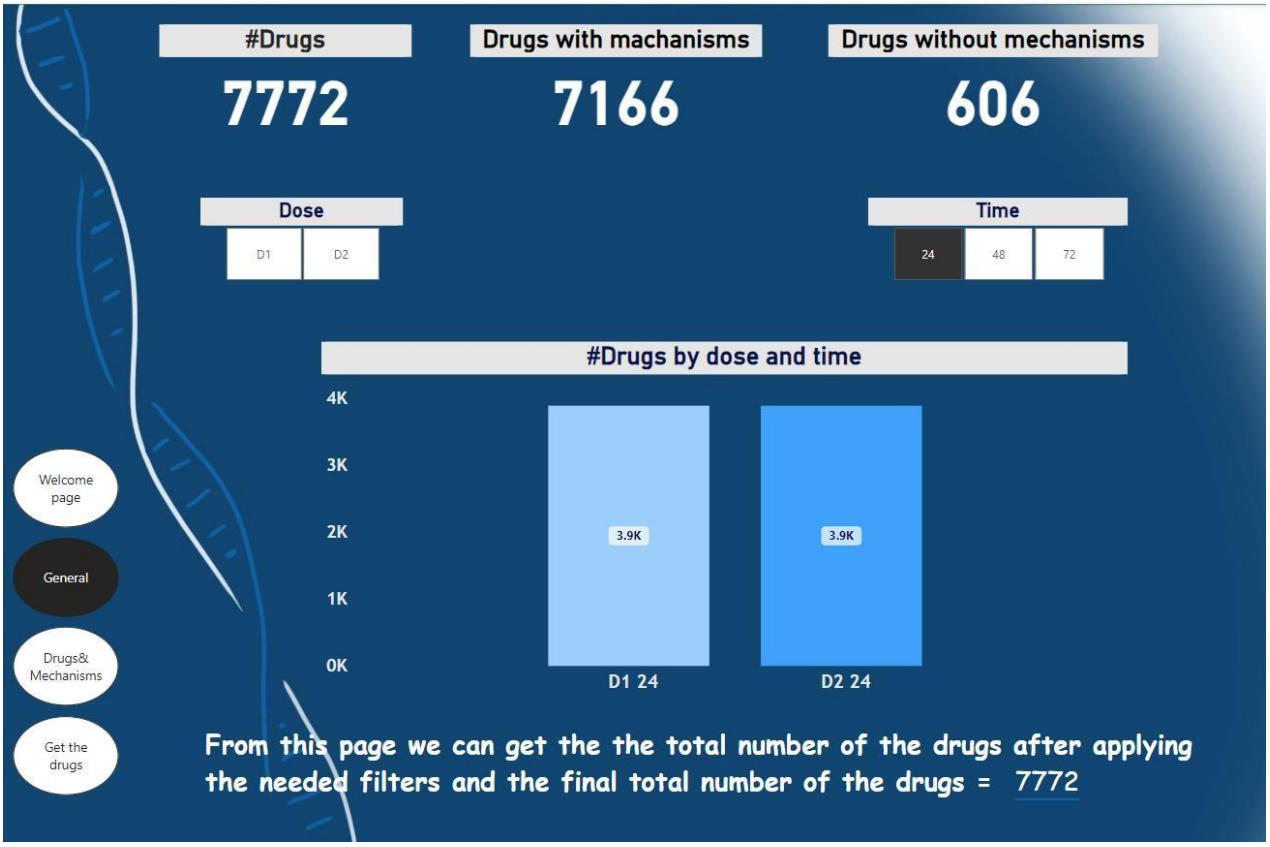


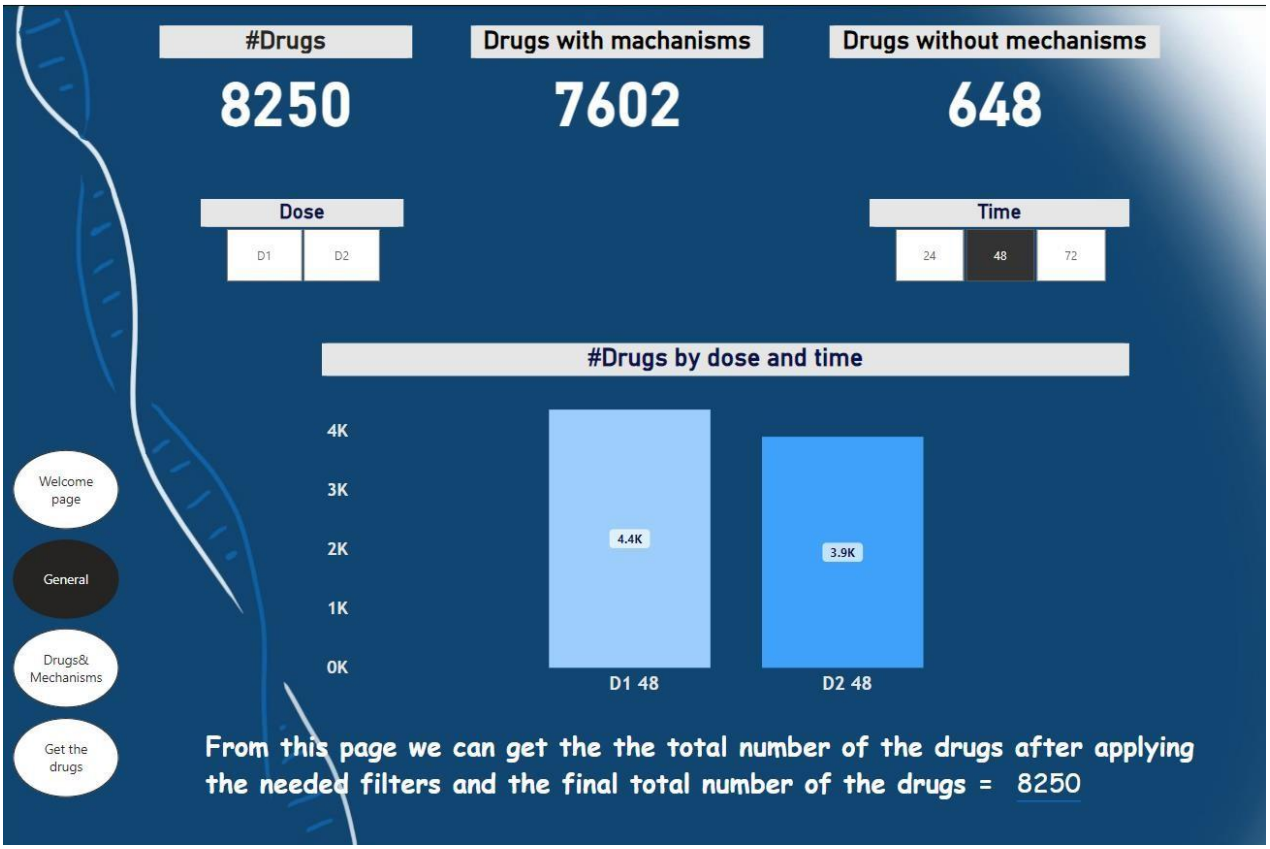
As we say , we can use the two filters to specify the drugs we wanted to show .

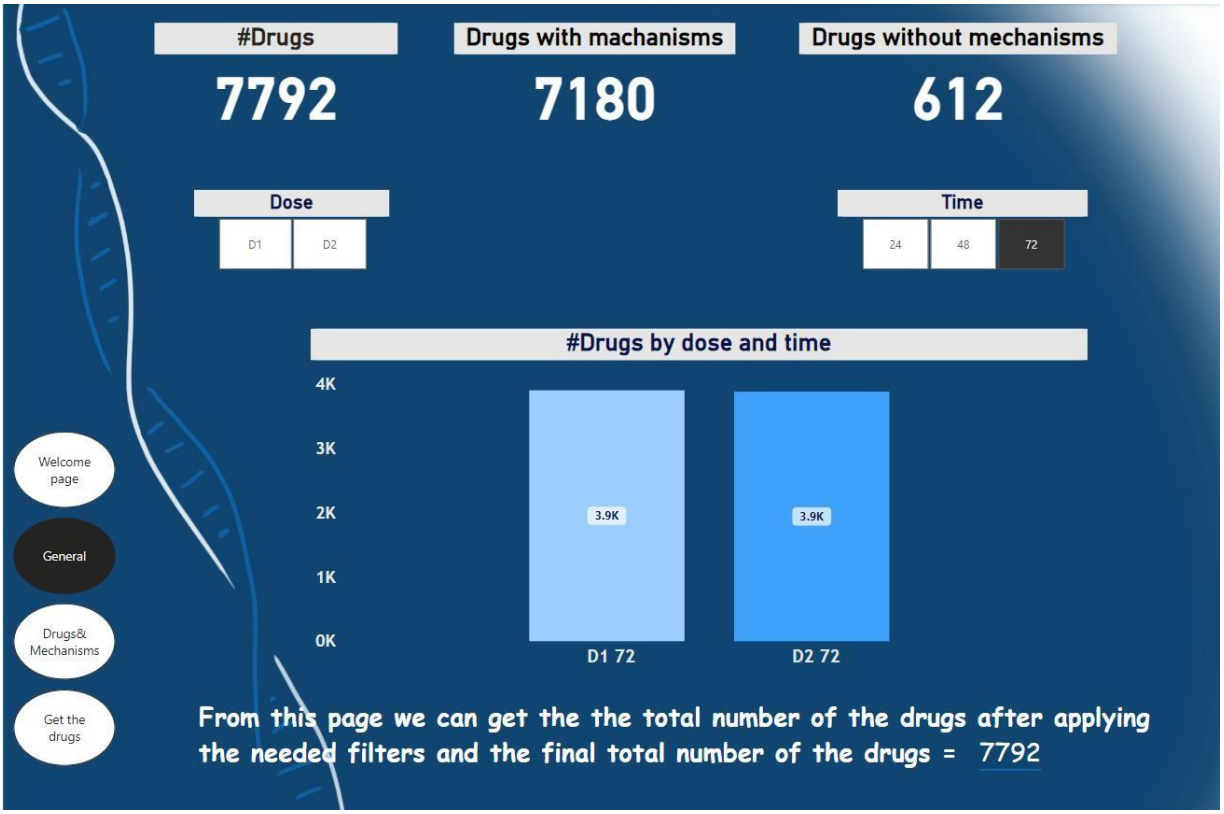
Let’s see some examples in the next pages:

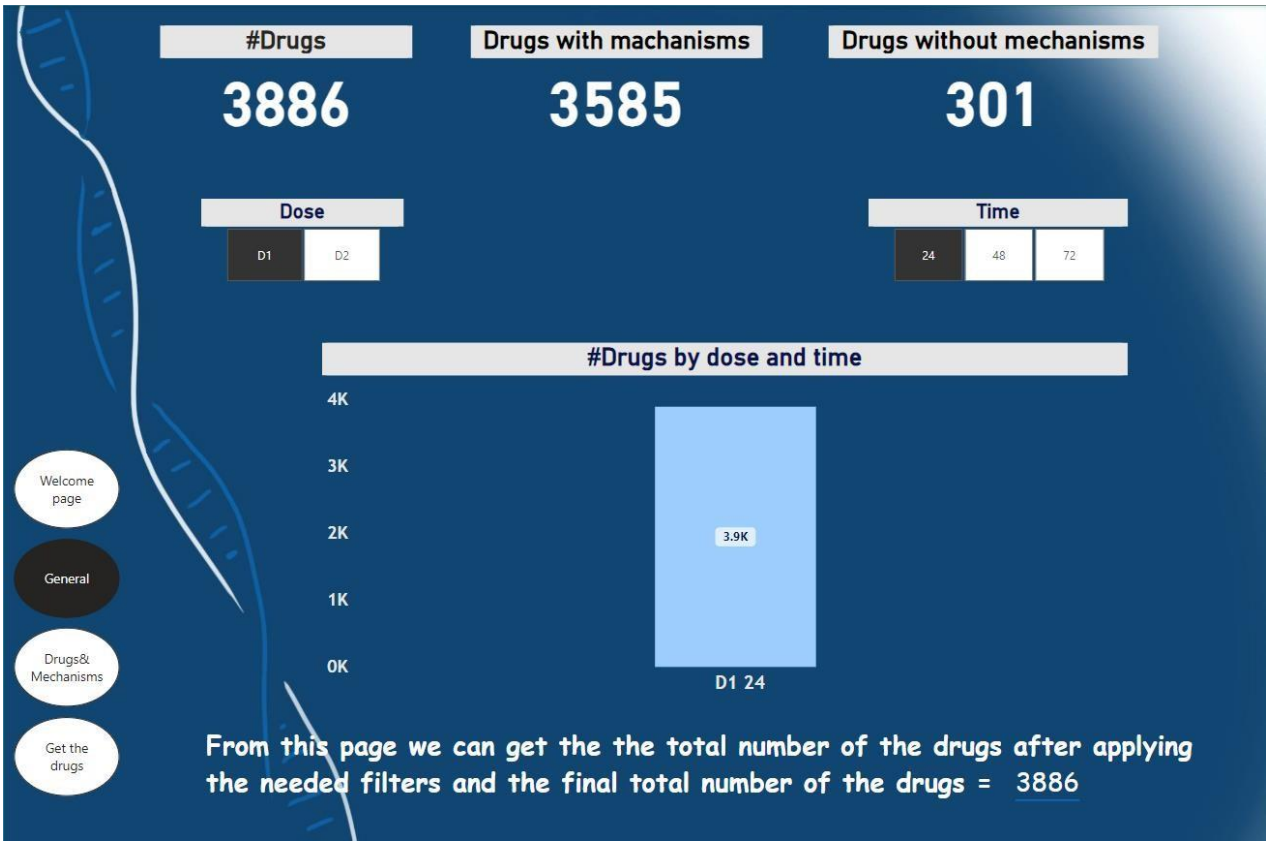

#Drugs
12.15K
Drugs with machanisms
11K
Drugs without mechanisms
951
Dose
D1
D2
Time
24
48
72
#Drugs by dose and time
4K
3K
2K
1K
0K
3.9K
4.4K
3.9K
D1 24
D1 48
D1 72
Welcome page
General
Drugs& Mechanisms
Get the drugs
From this page we can get the the total number of the drugs after applying the needed filters and the final total number of the drugs = 12147

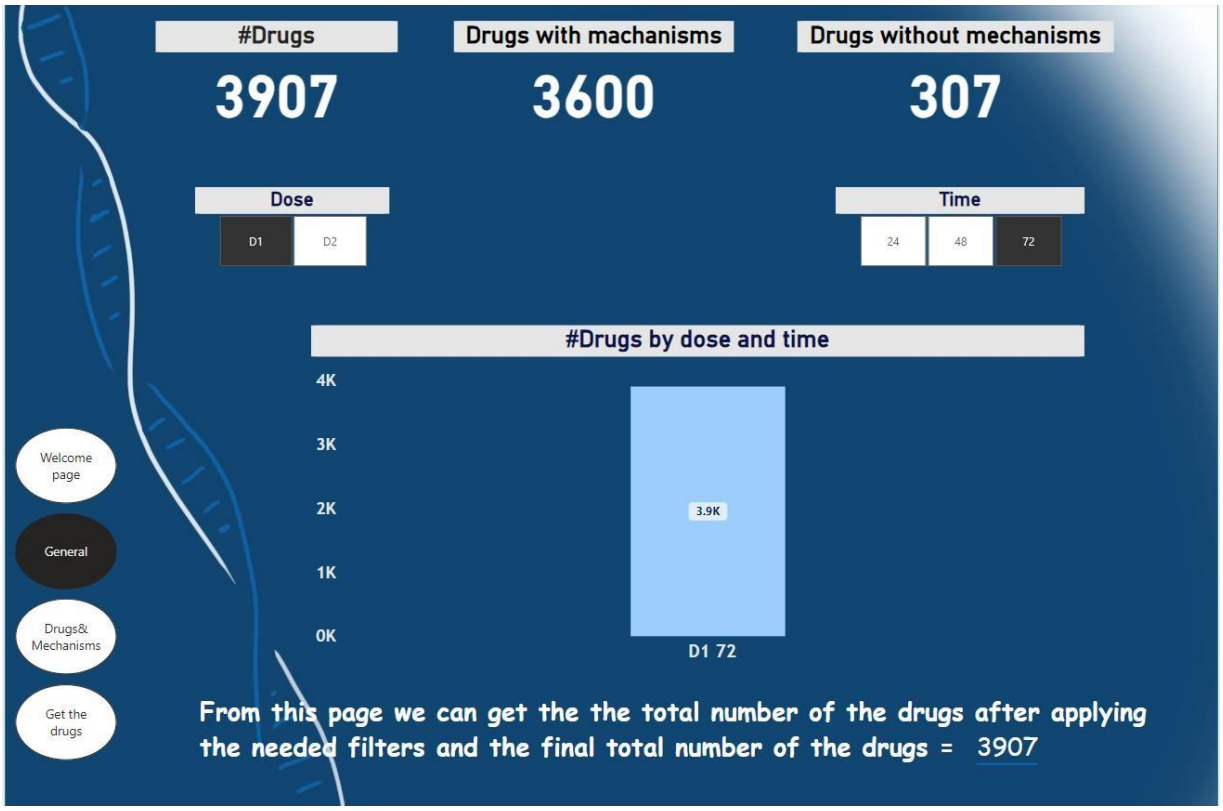

#Drugs
11.67K
Drugs with machanisms
11K
Drugs without mechanisms
915
Dose
D1
D2
Time
24
48
72
#Drugs by dose and time
4K
3K
2K
1K
0K
3.9K
3.9K
3.9K
D2 24
D2 48
D2 72
Welcome page
General
Drugs& Mechanisms
Get the drugs
From this page we can get the the total number of the drugs after applying the needed filters and the final total number of the drugs = 11667

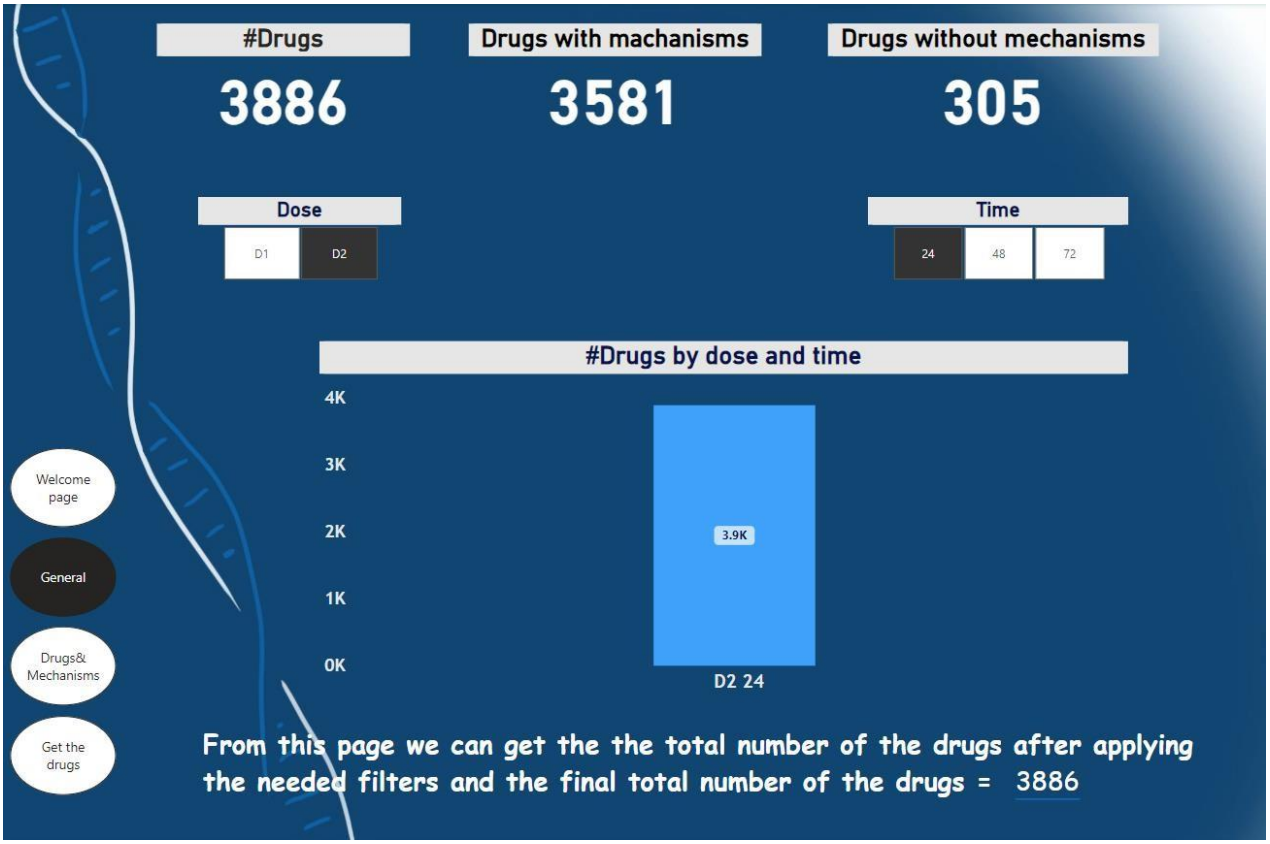

#Drugs
7772
Drugs with machanisms
7166
Drugs without mechanisms
606
Dose
D1
D2
Time
24
48
72
#Drugs by dose and time
4K
3K
2K
1K
0K
3.9K
3.9K
D1 24
D2 24
Welcome page
General
Drugs& Mechanisms
Get the drugs
From this page we can get the the total number of the drugs after applying the needed filters and the final total number of the drugs = 7772

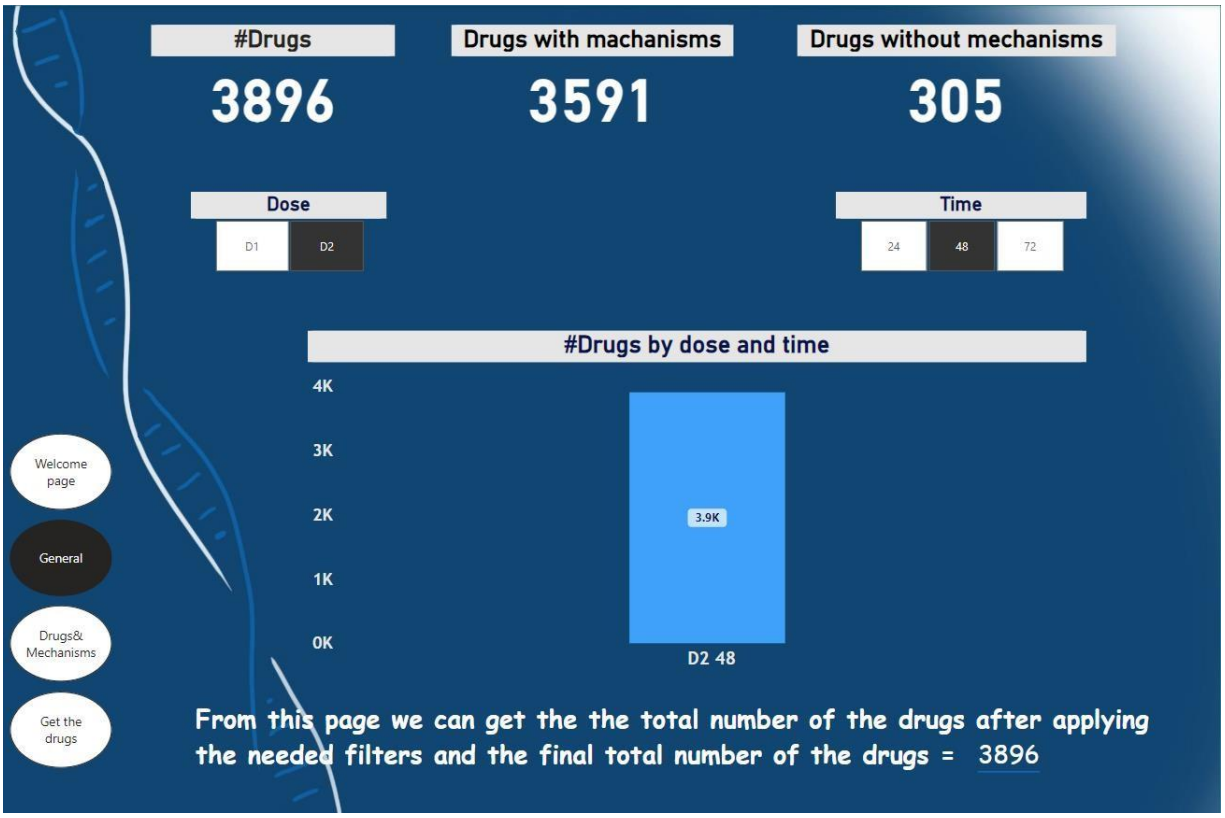

#Drugs
8250
Drugs with machanisms
7602
Drugs without mechanisms
648
Dose
D1
D2
Time
24
48
72
#Drugs by dose and time
4K
3K
2K
1K
0K
4.4K
3.9K
D1 48
D2 48
Welcome page
General
Drugs& Mechanisms
Get the drugs
From this page we can get the the total number of the drugs after applying the needed filters and the final total number of the drugs = 8250

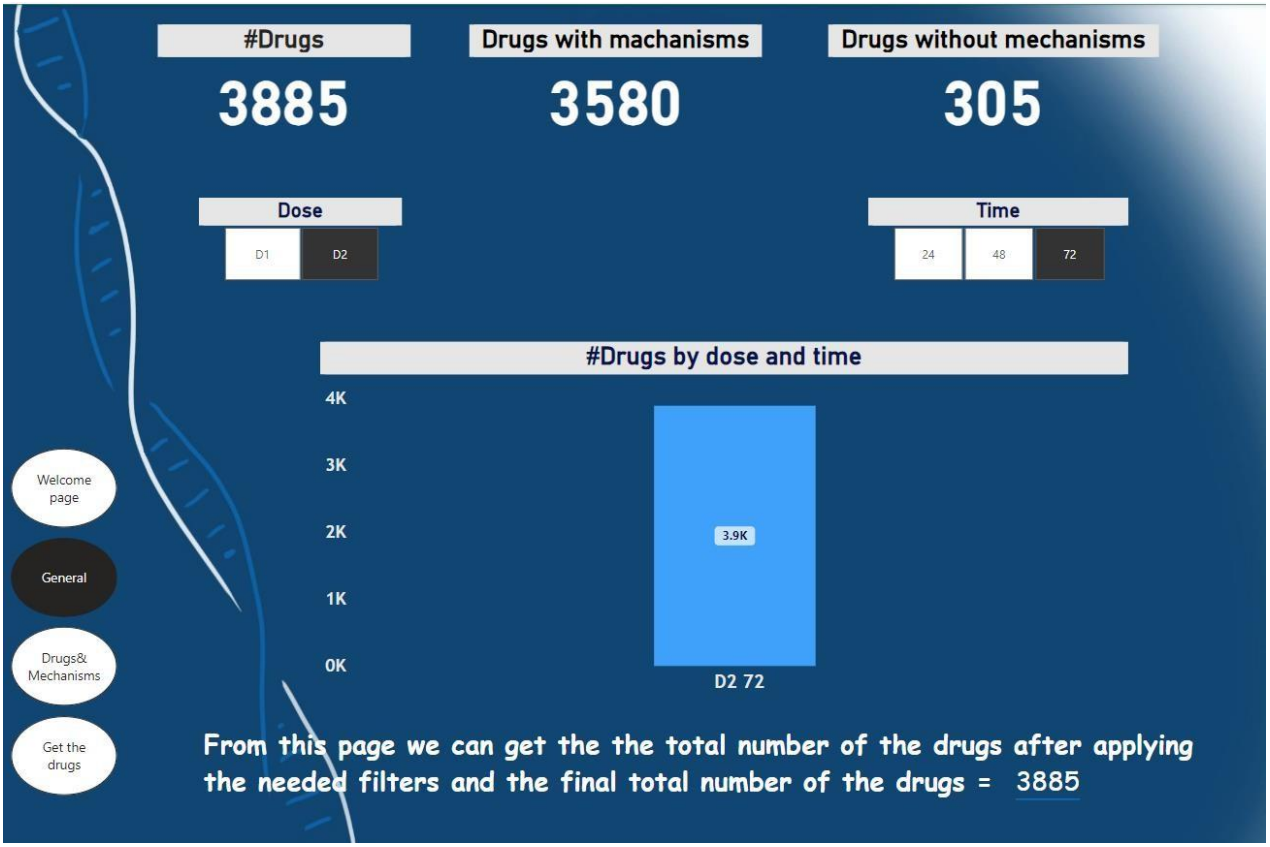

#Drugs
7792
Drugs with machanisms
7180
Drugs without mechanisms
612
Dose
D1
D2
Time
24
48
72
#Drugs by dose and time
4K
3K
2K
1K
0K
3.9K
3.9K
D1 72
D2 72
Welcome page
General
Drugs& Mechanisms
Get the drugs
From this page we can get the the total number of the drugs after applying the needed filters and the final total number of the drugs = 7792

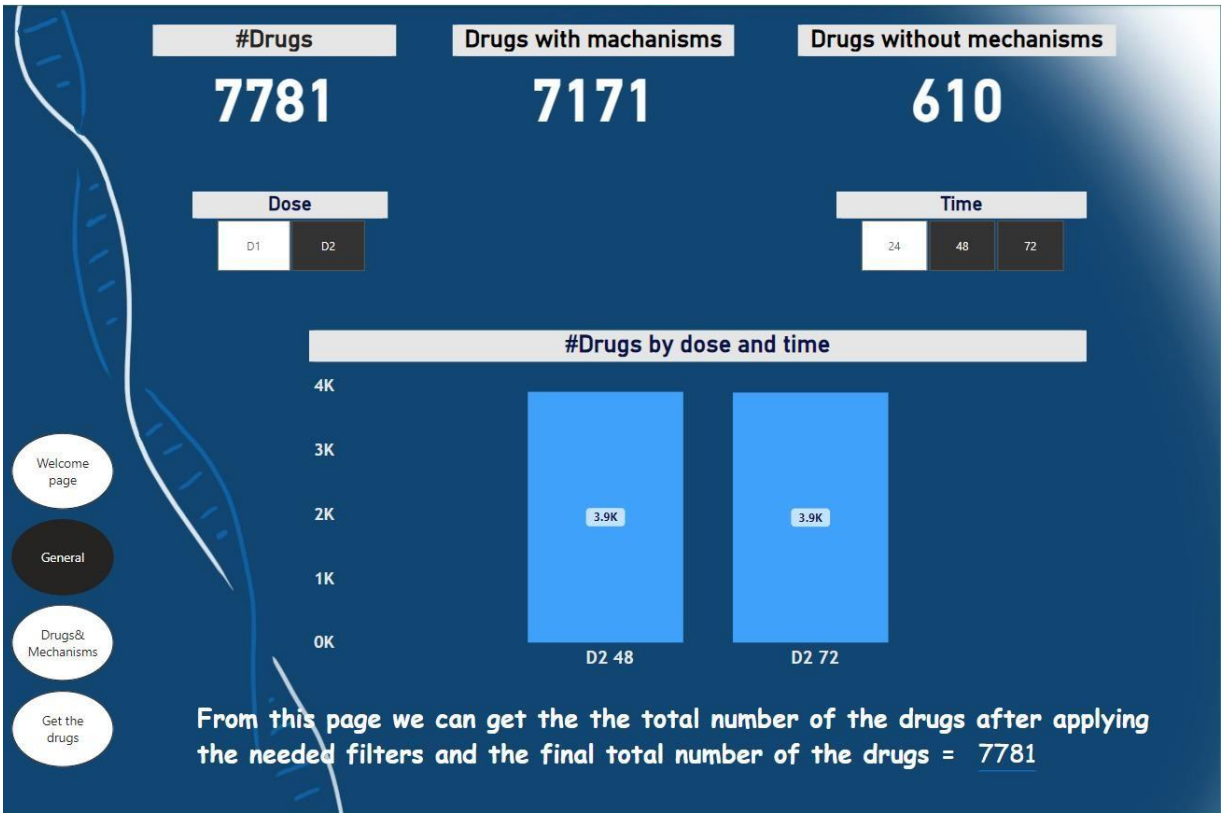

#Drugs
3886
Drugs with machanisms
3585
Drugs without mechanisms
301
Dose
D1
D2
Time
24
48
72
#Drugs by dose and time
4K
3K
2K
1K
0K
3.9K
D1 24
Welcome page
General
Drugs& Mechanisms
Get the drugs
From this page we can get the the total number of the drugs after applying the needed filters and the final total number of the drugs = 3886

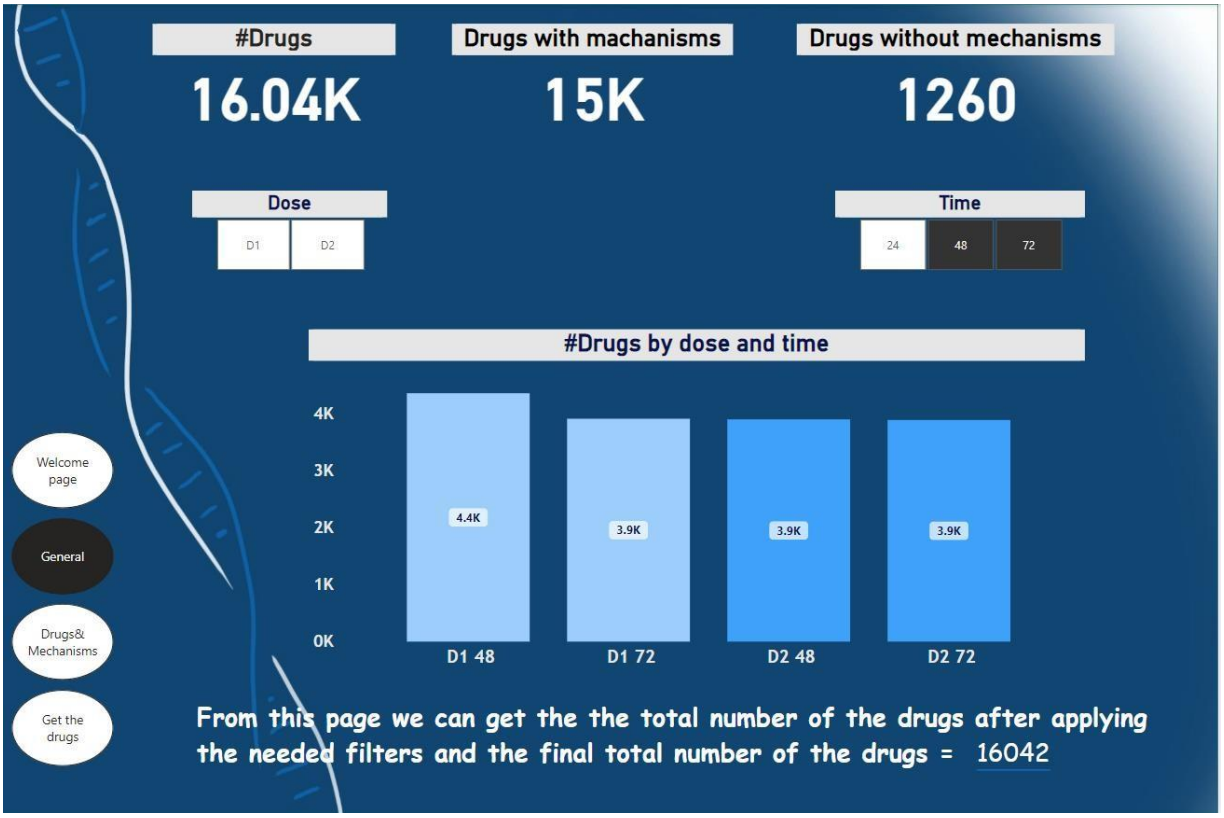

#Drugs
3907
Drugs with machanisms
3600
Drugs without mechanisms
307
Dose
D1
D2
Time
24
48
72
#Drugs by dose and time
4K
3K
2K
1K
0K
3.9K
D1 72
Welcome page
General
Drugs& Mechanisms
Get the drugs
From this page we can get the the total number of the drugs after applying the needed filters and the final total number of the drugs = 3907


#Drugs
3886
Drugs with machanisms
3581
Drugs without mechanisms
305
Dose
D1
D2
Time
24
48
72
#Drugs by dose and time
4K
3K
2K
1K
0K
3.9K
D2 24
Welcome page
General
Drugs& Mechanisms
Get the drugs
From this page we can get the the total number of the drugs after applying the needed filters and the final total number of the drugs = 3886


#Drugs
3896
Drugs with machanisms
3591
Drugs without mechanisms
305
Dose
D1
D2
Time
24
48
72
#Drugs by dose and time
4K
3K
2K
1K
0K
3.9K
D2 48
Welcome page
General
Drugs& Mechanisms
Get the drugs
From this page we can get the the total number of the drugs after applying the needed filters and the final total number of the drugs = 3896


#Drugs
3885
Drugs with machanisms
3580
Drugs without mechanisms
305
Dose
D1
D2
Time
24
48
72
#Drugs by dose and time
4K
3K
2K
1K
0K
3.9K
D2 72
Welcome page
General
Drugs& Mechanisms
Get the drugs
From this page we can get the the total number of the drugs after applying the needed filters and the final total number of the drugs = 3885


#Drugs
7781
Drugs with machanisms
7171
Drugs without mechanisms
610
Dose
D1
D2
Time
24
48
72
#Drugs by dose and time
4K
3K
2K
1K
0K
3.9K
3.9K
D2 48
D2 72
Welcome page
General
Drugs& Mechanisms
Get the drugs
From this page we can get the the total number of the drugs after applying the needed filters and the final total number of the drugs = 7781



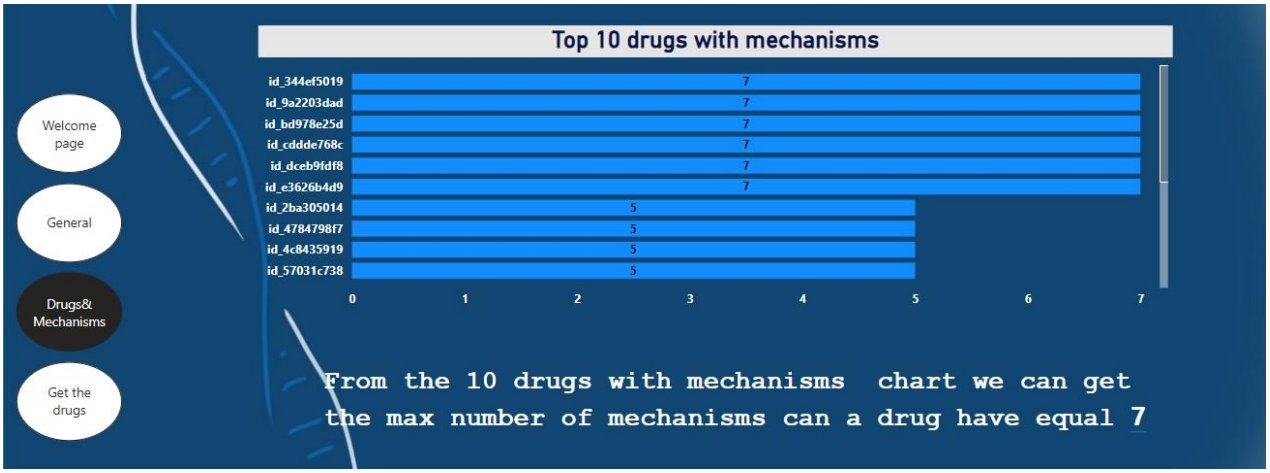
8.3 Drugs & Mechanisms

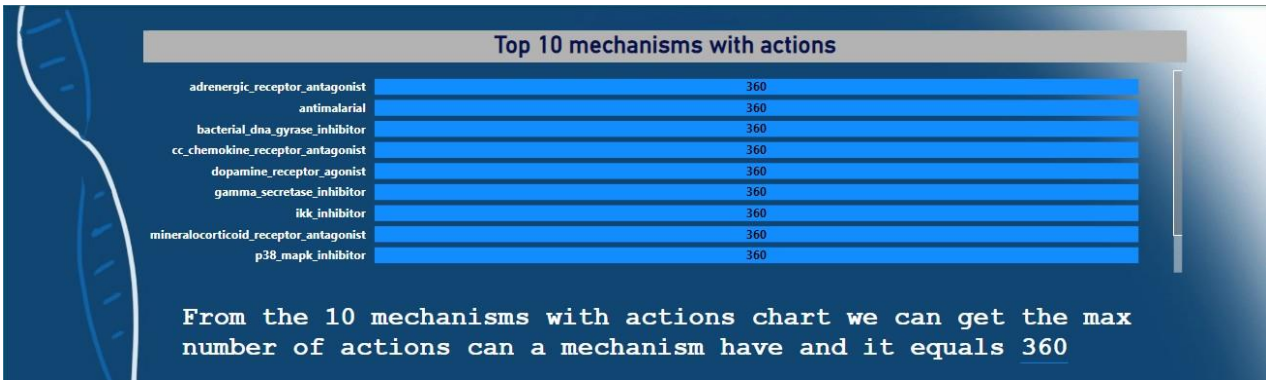
Analysis of Drugs and Mechanisms

-In the second page with name "Drugs & Mechanisms", we use bar charts to show some analysis for the drugs . In the bar chart with name "Top 10 mechanisms with actions", we show the top 10 mechanisms which achieved by the drugs and we noticed that the max number of mechanisms which is achieved = 360 mechanisms .

In the bar chart with name "Top 10 drugs with mechanisms", we show the top 10 drugs which achieve the max number of the mechanism of actions and we noticed that the max number of

the mechanism of actions which can be achieved by one drug = 7 mechanisms . In the next page we will show you the visualization related to it .





8.4 Detailed Filters and Data

Detailed Drug Data with Filters

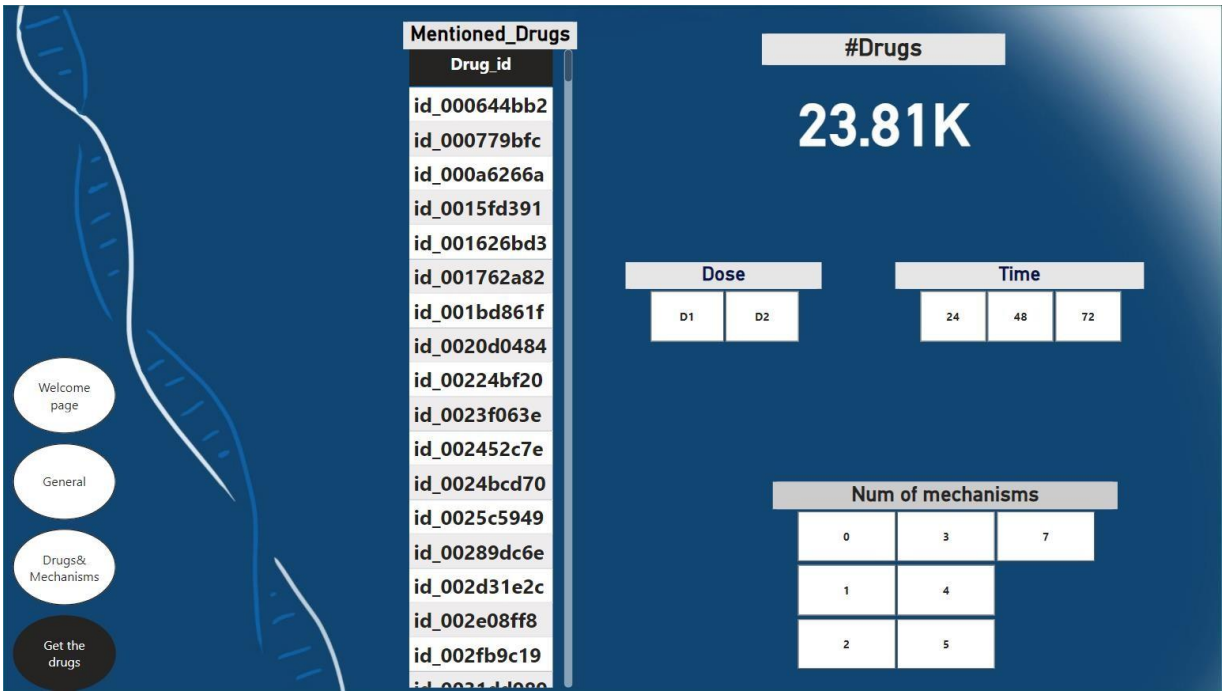
-In the third page with name "Get the drugs" , we have three filters one based on the doses , one based on the timing of the

doses and the last one based on the number of the mechanisms which be achieved by the drug and also I show a table which contains the the ids of all drugs .In the next page we will show you the visualization related to it .

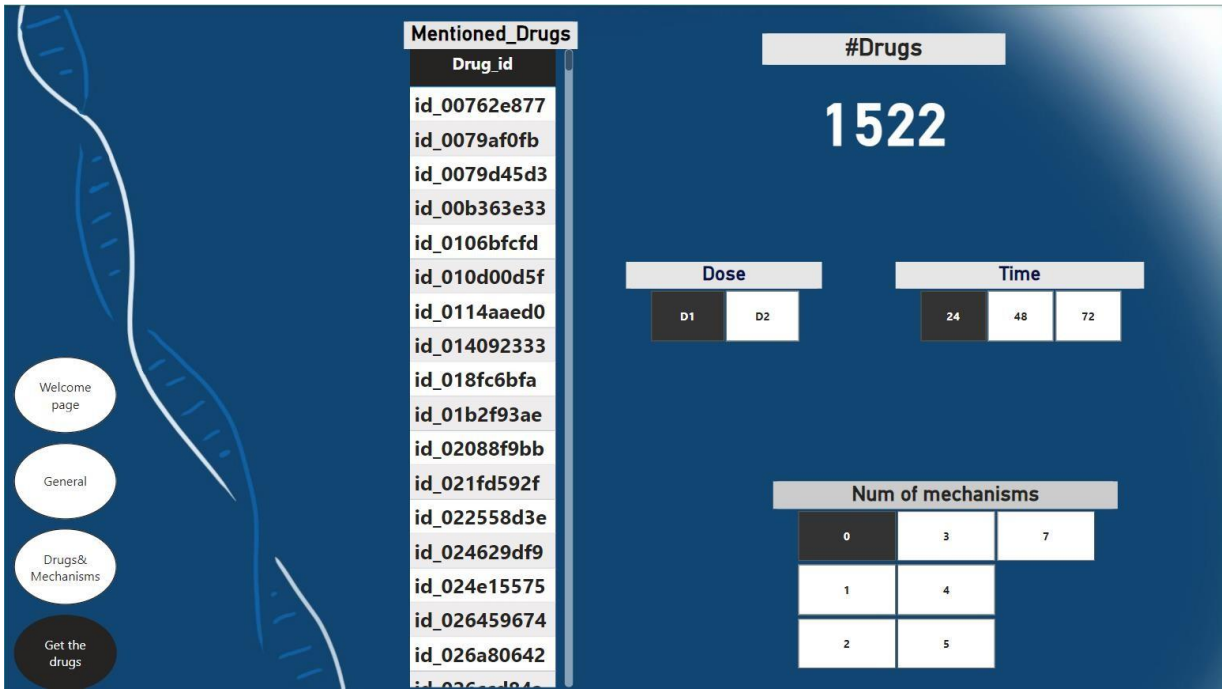
We can use these filters to specify the drugs we want to show .

For examples :

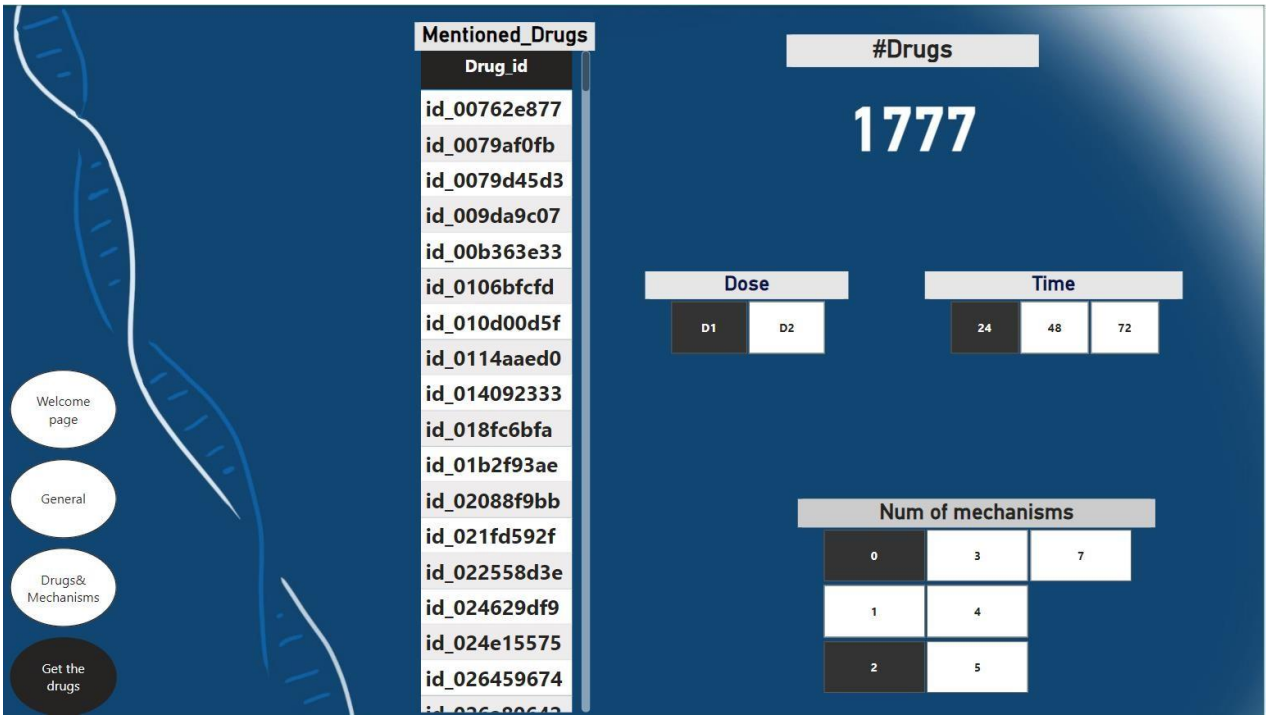
(1) we show all drugs without any specifications



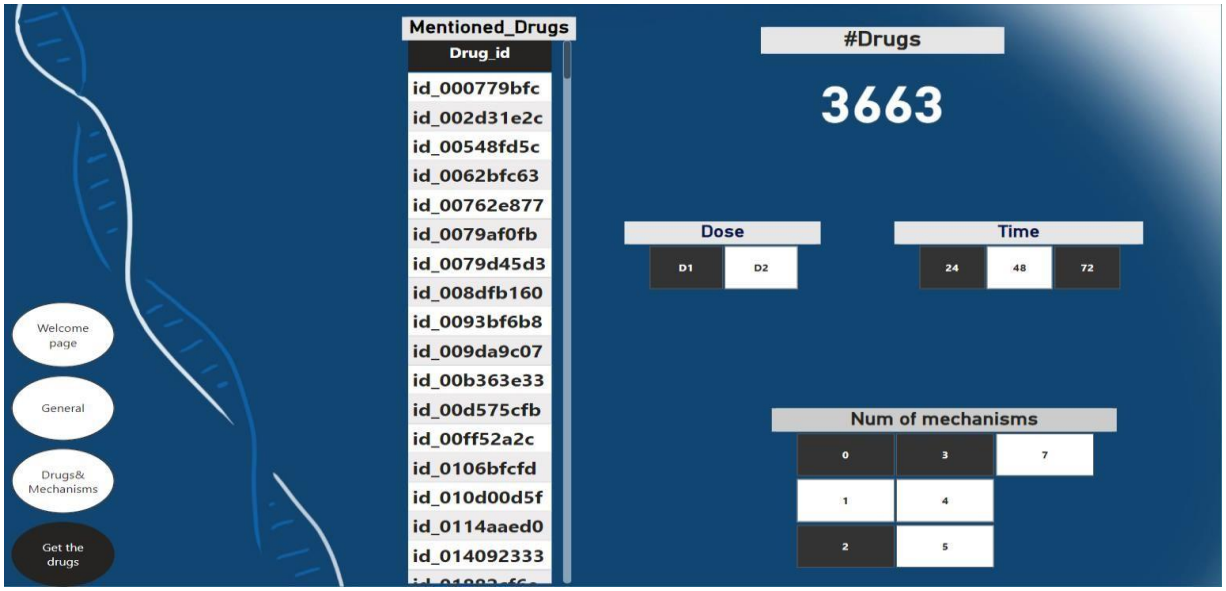
(2) We want to show the drugs with one dose each 24 hours and have no mechanism of action .



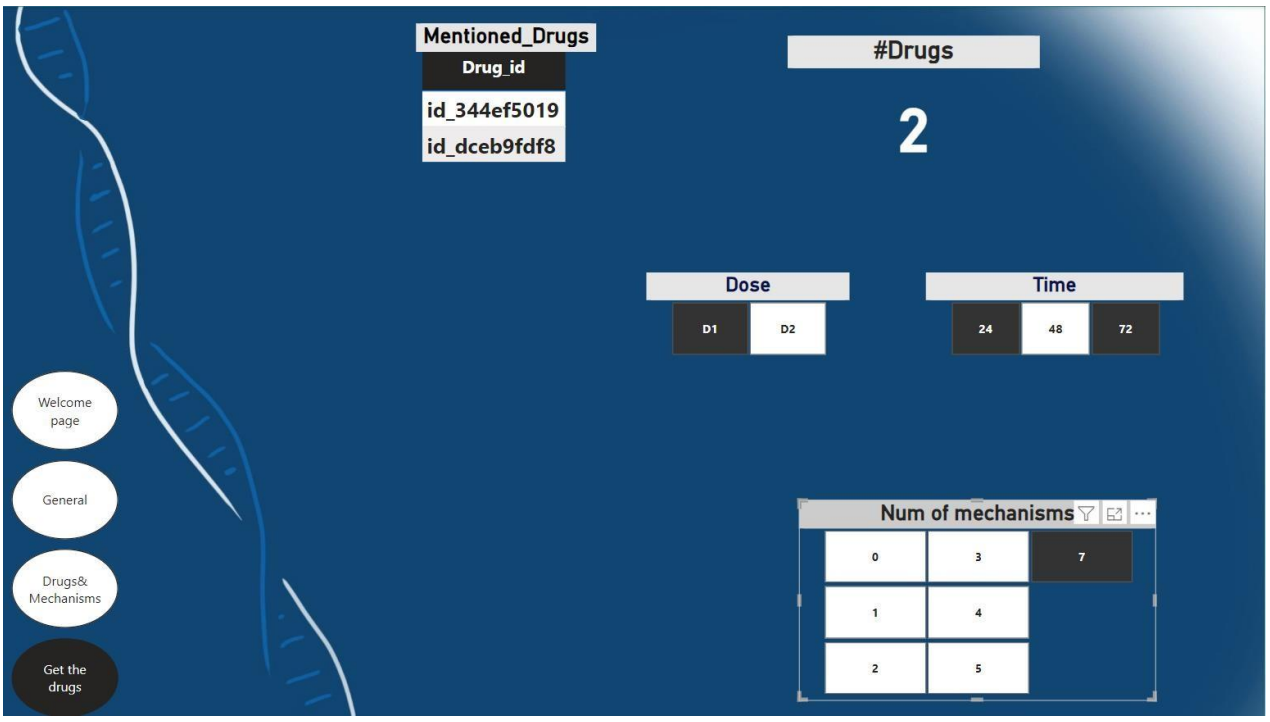
(3) We want to show the drugs with one dose each 24 hours and have no mechanism of action or have 2 mechanisms of action.



(3) We want to show the drugs with one dose each either 24 or 72 hours and have either no mechanism of actions , 3 mechanism of actions or 2 mechanism of actions.



(5) We want to show the drugs with one dose each either 24 or 72 hours and have 7 mechanism of action.



Chapter 9: Testing

Chapter 9: Testing

Importance of the Testing Phase

The testing phase is a critical component in the machine learning pipeline. Its primary objective is to evaluate the performance and generalizability of trained models on unseen data. This phase ensures that the model not only performs well on the training data but also can make accurate predictions on new, previously unseen data. Evaluating models during the testing phase provides insights into their robustness, reliability, and potential issues, guiding further improvements and tuning.

Evaluating Performance of Different Models

In this chapter, we explore the performance metrics of various models applied during the testing phase. The main metric used for evaluation is log loss, which measures the accuracy of probabilistic predictions. A lower log loss indicates better performance, as it reflects a smaller difference between the predicted probabilities and the actual outcomes.

Below, we present a summary of the models evaluated, along with their performance on the training, validation, and test datasets.

Performance Metrics of Various Models

One-vs-Rest Classifier

Model	Training Data Log Loss	Validation Data Log Loss	Test Data Log Loss
Logistic Regression	-	7.99	7.97
GaussianNB	-	6.70	6.35
Extra Tree Classifier	-	12.31	12.06
SGDClassifier	-	7.96	7.98
LinearSVC	-	10.55	10.77

Binary Relevance

Model	Training Data Log Loss	Validation Data Log Loss	Test Data Log Loss
GaussianNB	-	6.35	6.79

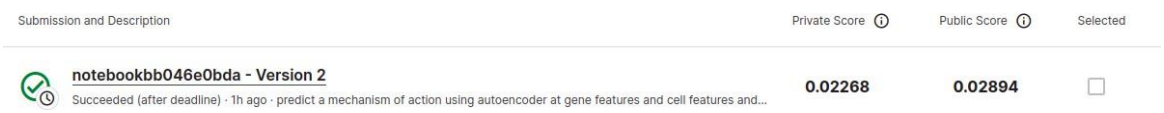
Adapted Algorithms

Model	K Value	Training Data Log Loss	Validation Data Log Loss	Test Data Log Loss
MLKNN	20	-	3.64	3.75
MLKNN	10	-	3.83	3.92
Classifier Chain				
- GaussianNB		-	25.54	25.17
- Random Forest		-	3.42	3.45

Classifier Chain with Feature Extraction by Autoencoder

Model	Training Data Log Loss	Validation Data Log Loss	Test Data Log Loss
Random Forest	-	0.02894	0.02268

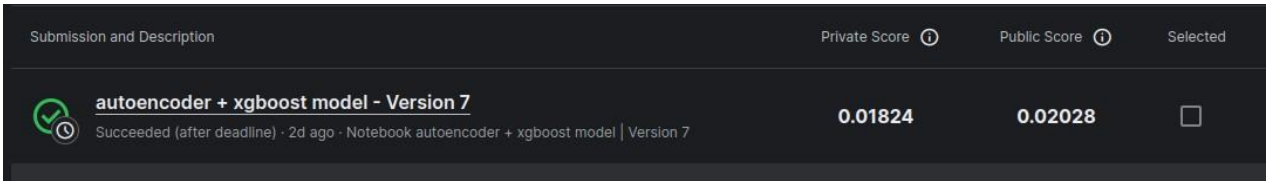
Score on kaggle



MultiOutputClassifier

Model	Training Data Log Loss	Validation Data Log Loss	Test Data Log Loss
XGBoost Classifier	-	0.0169	0.0169
XGBoost Classifier with Autoencoder Features	-	0.0173	0.0173

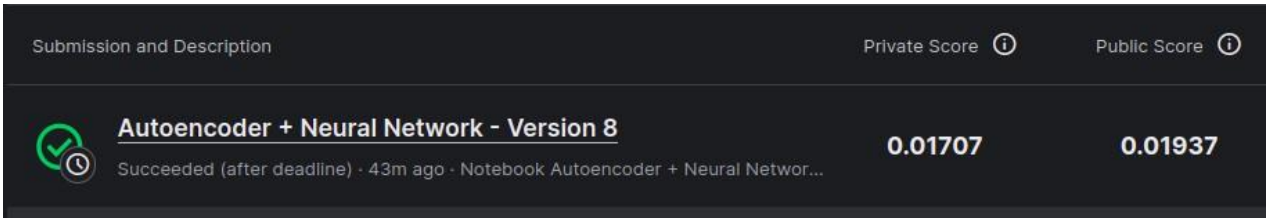
Score on kaggle

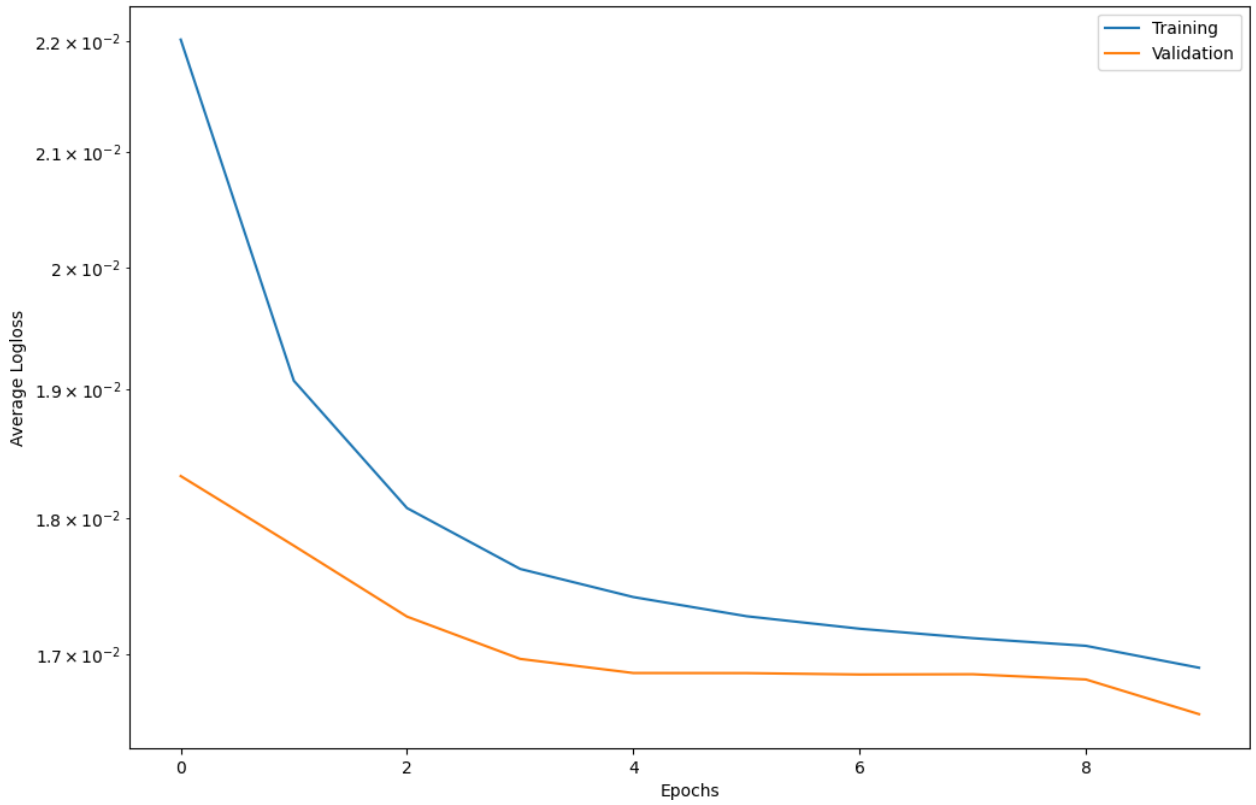


Final Model: Artificial Neural Network (ANN)

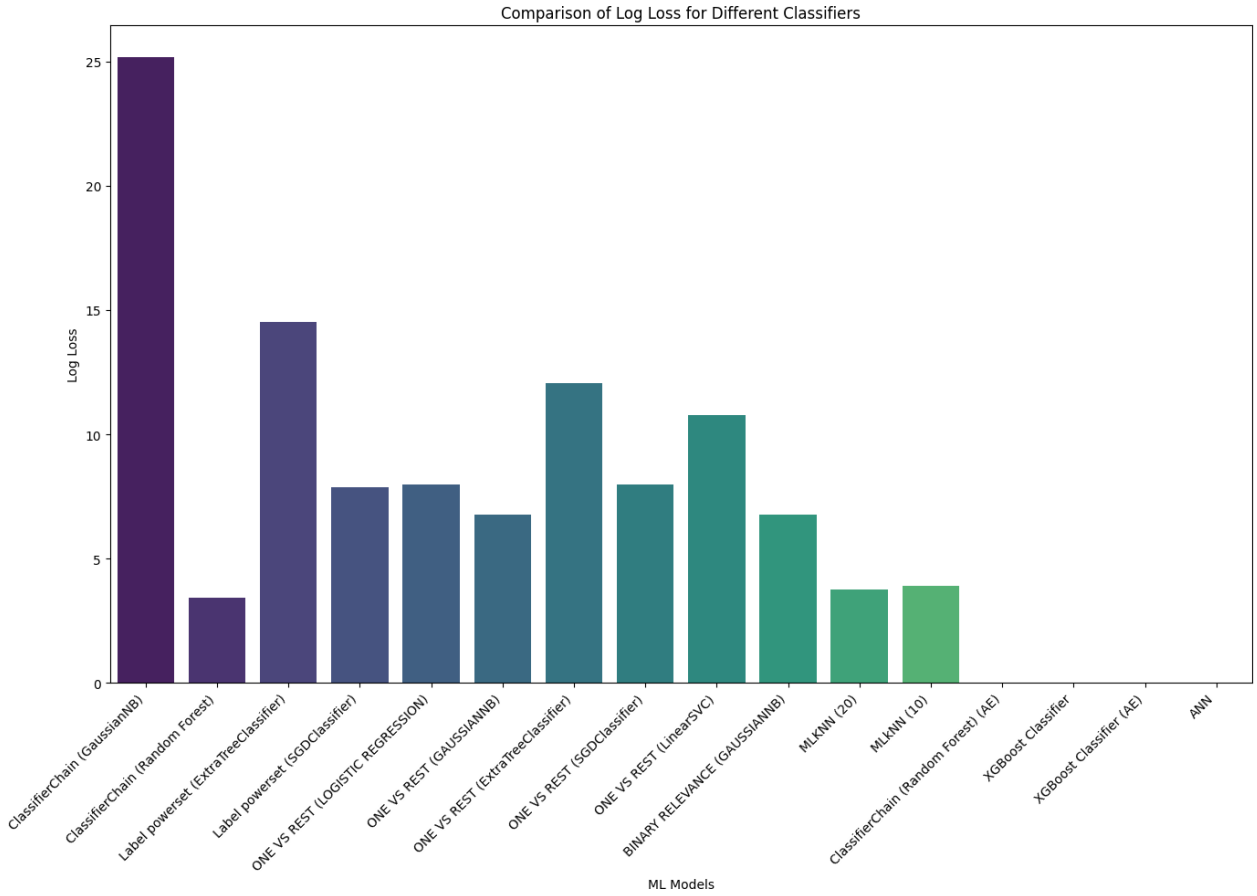
Model	Training Data Log Loss	Validation Data Log Loss	Test Data Log Loss
ANN	0.016	0.0157	0.01707

Score on kaggle

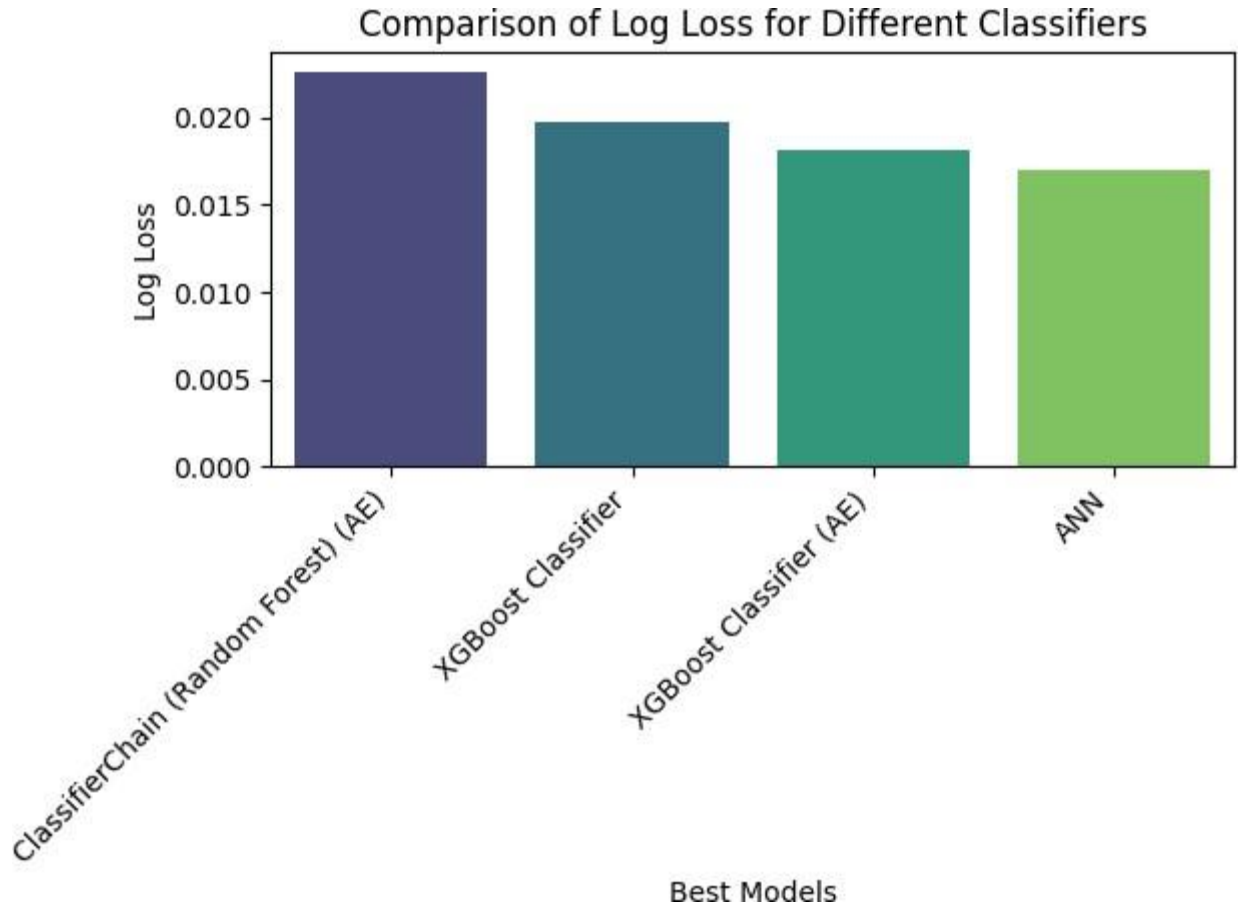




ANN Log Loss Graph



In the testing phase, we evaluated several models using log loss as the primary performance metric. The results indicate that feature extraction using autoencoders significantly improved the performance of the classifier chains, xgboost and ANN models achieving the lowest log loss values.



The final model, an Artificial Neural Network, demonstrated excellent performance across all datasets, suggesting its robustness and effectiveness in predicting the mechanism of action of drugs.

By thoroughly testing and evaluating these models, we ensure that the chosen model generalizes well to new data, providing reliable and accurate predictions in real-world scenarios.

Chapter 10:

Conclusion & Future Work

Chapter 10: Conclusion & Future Work

10.1 Conclusion

10.1.1 Summary of Key Achievements:

- This project set out to develop a predictive model for the Mechanisms of Action (MoA) of compounds, as exemplified by the Kaggle competition "Mechanisms of Action (MoA) Prediction." We successfully created a robust predictive model and a user-friendly web application to facilitate this process.
- We have demonstrated the application of bioinformatics and machine learning techniques in predicting the MoA of drugs, enhancing the understanding of drug interactions at the molecular level.

10.1.2 Impact on Drug Discovery:

- Our project contributes significantly to the field of drug discovery by providing tools that help identify potential drug candidates and predict their mechanisms of action. This can accelerate the development of new therapies and improve the efficiency of the drug discovery process.

- By integrating biological and pharmacological insights with machine learning, we have created a comprehensive resource that can be utilized by researchers and pharmaceutical companies.

10.1.3 Educational Value:

- The project offers substantial educational resources within the application, making it a valuable tool for students and beginners in bioinformatics and pharmacology.
- The detailed classifications of MoAs, along with associated diseases, provide a rich learning experience that bridges computational and biomedical knowledge.

10.1.4 Practical Applications:

- The inclusion of a case study on Type II Diabetes illustrates the practical applications of our predictive models.
- The web application can be extended to support personalized medicine, allowing for tailored therapeutic strategies based on individual genetic and molecular profiles.

10.2 Future Work

10.2.1 Data Expansion:

- We aim to train our model on larger datasets that include more features, such as DNA and RNA sequences. This will enhance

the model's ability to provide more accurate and personalized predictions for each individual.

- Expanding the dataset to include a wider variety of drug compounds and their associated MoAs will improve the model's accuracy and applicability.

10.2.2 Model Enhancements:

- Future work will involve refining the machine learning models by incorporating advanced algorithms and techniques. Continuous improvement of the models will be pursued through user feedback and integration of new data.
- We will explore the use of more sophisticated machine learning methods, such as deep learning, to further enhance the predictive power of our models.

Integration of Biological Features:

- Integrating numerical data for biological features of each drug sample into the model will be a priority. This integration will enhance the model's precision in predicting the MoA and tailoring treatments.
- The inclusion of comprehensive biological data, such as genomic and proteomic information, will enable more accurate and individualized predictions.

10.2.3 Broader Applications:

- We will investigate the application of our predictive models to other diseases and biological pathways, broadening the scope of the project.
- Expanding the web application to support a wider range of therapeutic areas will increase its utility and impact.

Collaboration and User Engagement:

- Encouraging collaboration with other researchers and institutions will be essential for the continued development and refinement of the application.
- Engaging with users to gather feedback will be crucial for improving the user experience and ensuring the application meets the needs of its intended audience.

References

- ✓ Rang and Dale's Pharmacology
- ✓ Medical Pharmacology and Therapeutics
- ✓ Background on Diabetes:

  American Diabetes Association. (2020).

  World Health Organization. (2020).

  International Diabetes Federation. (2019).
- ✓ Type 2 Diabetes Specific Information:

  Pathogenesis of type 2 diabetes mellitus.

  Medical Clinics, Microvascular and macrovascular complications of diabetes.
- ✓ Traditional plant medicines as treatments for diabetes. Cellular and molecular mechanisms of metformin: an overview.
- ✓ Drugs and Medications for Diabetes:

  Diabetes medications as monotherapy or metformin-based combination therapy for type 2 diabetes.

- ✓ Kaggle competition: https://www.kaggle.com/competitions/lish-moa
- ✓ https://medium.com/analytics-vidhya/mechanisms-of-action-moa-prediction-c4fa105e0d34
- ✓ https://suraj1997lodh.medium.com/mechanism-of-action-moa-prediction-6ed93ab8873f
- ✓ https://medium.com/@sameer.pandey617/mechanism-of-action-moa-prediction-a-detailed-case-study-b6ae6a8cad4c
- ✓ https://github.com/shiladityamajumder/mechanism-of-action?source=post_page-----c4fa105e0d34----------
- ✓ https://www.kaggle.com/code/headsortails/explorations-of-action-moa-eda/report
- ✓ https://github.com/oleksandrsirenko/mechanisms-of-action-moa-prediction/tree/master?tab=readme-ov-file
- ✓ https://www.kaggle.com/competitions/lish-moa/discussion/201510

- ✓ https://dongr0510.medium.com/multi-label-classification-example-with-multioutputclassifier-and-xgboost-in-python-98c84c7d379f
- ✓ https://www.kaggle.com/code/fchmiel/xgboost-baseline-multilabel-classification