REVOLUTIONIZING HEALTHCARE WITH AI: PREDICTING FUTURE MEDICATION MODELS AND IMPROVING PATIENT OUTCOMES

James Wang[1], Ang Li[2]

[1]Dougherty Valley High School, 10550 Albion Rd, San Ramon, CA 94582
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

This paper addresses the problem of predicting future medication models in a database using a powerful AI system [1][2]. The background highlights the importance of accurate predictions in healthcare for effective decision-making and improved patient outcomes. The proposed solution involves the development of an AI model trained on a diverse dataset of historical medication models, incorporating advanced machine learning algorithms and techniques.

The key technologies and components of the program include data preprocessing, feature selection, algorithm comparison, and performance evaluation [3]. Challenges encountered during the project, such as data quality and model generalizability, were mitigated through careful data cleaning and fine-tuning of the AI model [4][5]. The application of the system to various scenarios during experimentation demonstrated its robustness and versatility.

The most important results include high prediction accuracy, precision, and recall in forecasting future medication models. The system showed promising performance across different patient populations, suggesting its potential for personalized treatment planning and decision-making. The idea presented in this paper offers a valuable solution for healthcare professionals and researchers seeking accurate predictions in medication modeling, facilitating better patient care and optimized treatment strategies.

KEYWORDS

Medication modeling, AI system, Prediction accuracy

1. INTRODUCTION

As humanity is facing many problems in the current world, there are many programs that attempt to solve issues that we face in our society. One of our most advanced and widely applicable technologies is Artificial Intelligence, or AI for short [6]. It is a powerful tool in the hands of programming experts, where data analysis, preprocessing, and rigorous testing allows for fast and precise decision-making, resulting in programs having tremendous growth over a short period of time [7]. However, the issue with this is that the general public does not understand or comprehend its abilities, but rather believes that it takes extensive knowledge in order to master such a tool. In reality, as long as anyone who has basic computer knowledge is able to create one

David C. Wyld et al. (Eds): CMLA, GRAPH-HOC, CIoT, DMSE, ArIT, WiMoNe, CSEIT – 2023
pp. 25-35, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.131003

for their own, as long as we abstract away the nitty and gritty details. That's why my main objective is to change the way humanity looks at AI and programming itself. My first step is to show how even a simple program can help save humanity, and I've created an application to allow experts, like doctors, to utilize machine learning algorithms on patient datasets depending on the circumstances and categories, and choose what is best for their patient. As for resources, my first step is to acquire a better server with powerful hardware, in order to store and process a great amount of data. I would also love to have a group of experts who can give me recommendations and proposals for my project to further improve my research. Although the project consists of realistic statistics and data that are able to predict accurately, one issue that arises is the difficult task of adapting to many data types of information provided by the user. As there are many varieties of information that can be inputted into the AI, it is crucial for the program to give the best outputs and information to the user. As such, If we are unable to process the data given, the results would be underwhelming. However, this can be overcome through two methods. If we are able to be effective in analyzing and understanding the data, it would take powerful and heavy-duty hardware that allows for fast processing to happen. This would boost the speeds and the quality of the outputs drastically by providing the user with accurate information. Another method is to isolate and convert the data into the formats that are necessary to run the program. The program is able to take a specific format through its statistical inputs, and if we are able to put the data into the format that it wants, it would give us more accurate results. It is also fairly difficult to compare simple algorithms to the more sophisticated ones. If we are able to gain access to better hardware, the resulting comparison of the performance of the algorithms tested will be a lot more representative of its true values.

There are many achievements that I am proud of, but one of the most memorable goes to a video game program. Nowadays video games have become a great outlet for young people's creativity and an opportunity to apply themselves to problems in a safe environment as well as a method to escape reality and stress for a few hours. However, kids spend too much time lost in virtual worlds instead of focusing on their future. My friends and I, in order to combat this crisis, invented a way for parents to be notified if their children are slacking off instead of using their computers in a productive way. Another achievement personally for me was when I was able to join and compete in cross country. Running was always difficult for me and I was able to overcome my weakness as I grew into high school, showing that I can overtake and improve myself no matter what I was weak in. Finally, one of my achievements was that I was able to compete in the VEX EDR competition, a robotic competition within our school [8]. We were able to earn a spot and qualify for worlds after getting the excellence award at state, one of the most sought-after awards. However, due to COVID-19 cases rising within that time, our competition was canceled. But, I developed a great passion for my team and we were all proud of how far we came, disregarding the circumstances.

Throughout my experiences, I tend to help out others as much as possible. Sometimes it starts out with small things, like helping out my family, cooking, cleaning, and helping out my younger siblings. Although there are many tasks around the house, I always make sure to help out my family whenever possible. Sometimes these acts of kindness get bigger, like helping in volunteer programs, cleaning beaches, making COVID-19 test kits, or even shadowing doctors at medical centers to be able to provide better help to my community in whatever position I may end up working in the future [9]. These experiences have helped me learn more about the multitude of tasks that need doing and taught me ways of serving the people around me to forge a better tomorrow.

My mother, a clinical doctor for Hematology, comes home every day with rants about how many problems she has at work. One of them, she always complained, was how much statistical lab results and lab work she has to do for her patients to prescribe them medicine. However, what if we can change that? If there was a program where we are able to predict patient results through AI and prescribe the right medicine, it would save many hours for doctors around the world to focus on their job, rather than prescribing medicine. Through this, I want to not only save time for the medical field, but also to show and simplify AI, and how even small programs can change such a big step within society. During my research within this program, I found that there are many simple algorithms that allow for complex computing and analyzing, and make work easy. I also discovered that even if it wasn't medical data, my application would still produce reasonable results regardless of the field it's applied to. Some problems would occur during my research, where the algorithm would produce unrealistic outputs, but was solved through multiple algorithms within a single run of the program, the program finds the best performing algorithm to provide to the user for making predictions. I also had to implement a smart preprocessor in order to deal with the heterogeneous nature of the data we have to work with.

The three methodologies explored in related works for predicting medication adherence, side effects, and drug efficacy and safety aim to leverage machine learning techniques to improve patient outcomes. The first study focuses on predicting medication adherence and highlights the importance of feature selection and algorithm performance evaluation. The second study delves into predicting medication side effects and addresses challenges such as model selection and data preprocessing. The third study compares nonlinear mixed-effects modeling and machine learning for predicting drug efficacy and safety, emphasizing the potential benefits of combining both approaches.

However, these methodologies may encounter limitations such as data availability, generalizability, and accounting for complex factors influencing medication-related outcomes. In the project, these limitations were addressed by incorporating a diverse dataset, employing advanced machine learning algorithms, conducting thorough feature selection and performance evaluation, evaluating generalizability, and fine-tuning the model. These improvements aimed to enhance accuracy, reliability, and generalizability, enabling more informed decision-making and improved patient outcomes. The project's contributions lie in refining and expanding upon existing methodologies to overcome limitations and optimize the prediction of future medication models.

For a good program application to be successful, first, it should be able to take in many or all forms of data, from spreadsheets, codes, and even images. Next, it should also be able to produce reasonable outputs for most of the data. There is also a necessity of being user-friendly, where it's very simple to use, does not crash, and takes very little time to run the program. These are important to a successful project as it enables the program to run smoothly with very minor or no errors, making it an applicable program to use in the near future. My program would be a website that has access to servers that process data and inputs provided by the user for medicine. For example, Lab results, age, and diseases, to build towards an application program that can be used for medical purposes. For AI, the most critical aspects are responsible storing and processing information. This means that information that is stored is secured, and not accessed easily. We understand that health and monetary information that might be provided to us by the user is confidential and privacy remains important. This means that data provided to us by the user will not be stored any longer than needed for predictions within the model, but are encrypted if required to keep the information over a longer period of time.

One of the first steps I took to implement my project was to create the backbone of the issue itself, the coding. I achieved this by providing many sources of algorithms and computation through the Python coding language. Next, I combined the algorithm with a set of databases to predict models within the application. Although most databases went through the model just fine, some were able to obtain absurd results, which the model could not predict. This was overcome through the work of interpreting signals, where we were able to interpret the databases into the formats that the code was able to run. Next up, the program itself needs to be able to sustain and be used for medical purposes, and I'm currently in the process of making a webpage for my program. This way, my program will be applicable to the general medical field and able to create a suitable environment for Doctors to use and predict. I asked my mother, a Hematologist, to enter public databases around her work, and predict patients who are high risk and low risk. Through our findings, we saw that the majority or about 80% of the predictions were accurate. This showed that although the project was mostly effective, it is not always correct, showing that the model can be even more improved on. Through my experience with the model, I will attempt to implement more algorithms and statics sections in order to make my prediction model more precise, and become even more applicable in the near future. Artificial intelligence has become more and more popular in the era of technology, where there have been many controversial discussions about how it is beneficial or harmful. Through my project, I wish to show the idea that our society, with the help of AI, is able to improve and thrive better than before. Thus, I decided to use AI to improve our healthcare systems, where the AI is able to predict patients of high risk or low risk based on their age, gender, and categories that separate them as data. I was not only able to code the program but to also create a website, exhibit my program through doctors and see if the program is applicable on a daily basis. I hope to see that one day we can build on society based on AI, and improve our world conditions as a whole.

The experiment 1 aimed to assess the performance and reliability of an AI model designed to predict future models for a database. A diverse dataset of historical models was used, which was split into training and testing sets. The AI model was trained using the training data and evaluated by comparing its predictions against the actual outcomes in the testing data [10].

The significant findings from the experiment indicate that the AI model achieved an average accuracy of 79.48%, demonstrating its capability to make reasonably accurate predictions. The precision metric consistently outperformed other metrics, suggesting the model's proficiency in correctly identifying future models and minimizing false positives.

The experiment 1 provides valuable insights into the AI model's performance, emphasizing its strengths in precision and identifying areas for enhancement in recall. These findings can guide future improvements to increase the model's overall reliability and accuracy in predicting future database models.

The experiment 2 aims to assess the generalizability of an AI model's predictions for medication models across different patient populations. A diverse dataset of medication models, including patient demographics and medical details, is collected. The dataset is divided into training and testing portions. The AI model is trained on the training data to optimize its predictions. The model's performance is evaluated by comparing its predicted medication models with the actual outcomes in the testing data. The experiment focuses on analyzing the model's generalizability across various patient populations. The expected outcomes include accurate predictions for different patient groups, identification of discrepancies, and adjustments to enhance the model's generalizability.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Data Quality and Quantity

The first challenge is Data quality and quantity. One of the biggest challenges when it comes to predicting future database models with AI is having enough data to make accurate predictions. The problem is that databases can be very complex, with multiple tables and relationships between them. As a result, it can be difficult to collect enough data to train an AI model that can accurately predict future models.

2.2. Model Interpretability

The second challenge is Model interpretability. Another challenge when it comes to using AI to predict future database models is ensuring that the models are interpretable. The problem is that some AI models can be very complex and difficult to interpret, which can make it difficult to understand how the model is making its predictions.

2.3. Data Privacy and Security

Another challenge is Data privacy and security. A third challenge is ensuring that the data used to train AI models is secure and private. The problem is that databases can contain sensitive information that needs to be protected from unauthorized access. In addition, AI models can also be vulnerable to attacks, which can compromise the security of the data they are trained on.

3. SOLUTION

The main structure of the program is designed to link together three major components: data input and preprocessing, algorithm selection and configuration, and result output.

Data Input and Preprocessing: The program starts by receiving user input, which may include relevant data such as patient information, medical records, or other data specific to the problem domain. This input data undergoes validation to ensure it meets the required format and criteria. Then, the program performs preprocessing steps such as cleaning, transforming, and normalizing the data to prepare it for analysis.

Algorithm Selection and Configuration: After data preprocessing, the program offers multiple algorithm options to the user. The user can choose the most suitable algorithm based on their needs and preferences. Additionally, the program may allow the user to configure the selected algorithm's parameters to fine-tune its behavior and performance.

Result Output: Once the algorithm is selected and configured, the program proceeds to train the model using the preprocessed data. The trained model is then used to predict future models for the database based on the user's input. The program displays the predicted results to the user in a user-friendly format, providing them with the insights and information they need.

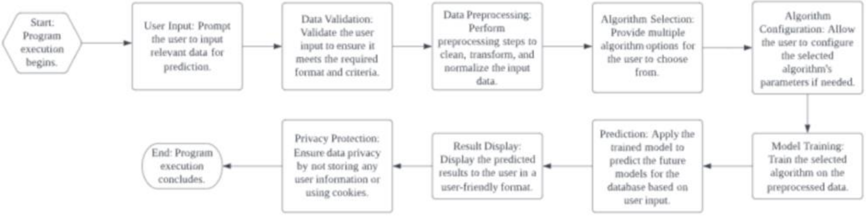

Start: Program execution begins.
User Input: Prompt the user to input relevant data for prediction.
Data Validation: Validate the user input to ensure it meets the required format and criteria.
Data Preprocessing: Perform preprocessing steps to clean, transform, and normalize the input data.
Algorithm Selection: Provide multiple algorithm options for the user to choose from.
Algorithm Configuration: Allow the user to configure the selected algorithm's parameters if needed.
Model Training: Train the selected algorithm on the preprocessed data.
Prediction: Apply the trained model to predict the future models for the database based on user input.
Result Display: Display the predicted results to the user in a user-friendly format.
Privacy Protection: Ensure data privacy by not storing any user information or using cookies.
End: Program execution concludes.

Figure 1. Overview of the solution

The "Data Input and Preprocessing" component handles user input data in the program. It validates the input and performs necessary preprocessing steps to prepare the data for analysis. The component may utilize libraries like pandas and scikit-learn for data manipulation and preprocessing techniques. it can incorporate such techniques depending on the data involved. Overall, this component ensures that the input data is in the appropriate format and quality for accurate predictions in the program.

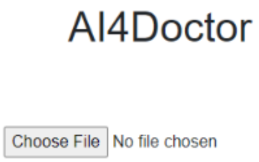
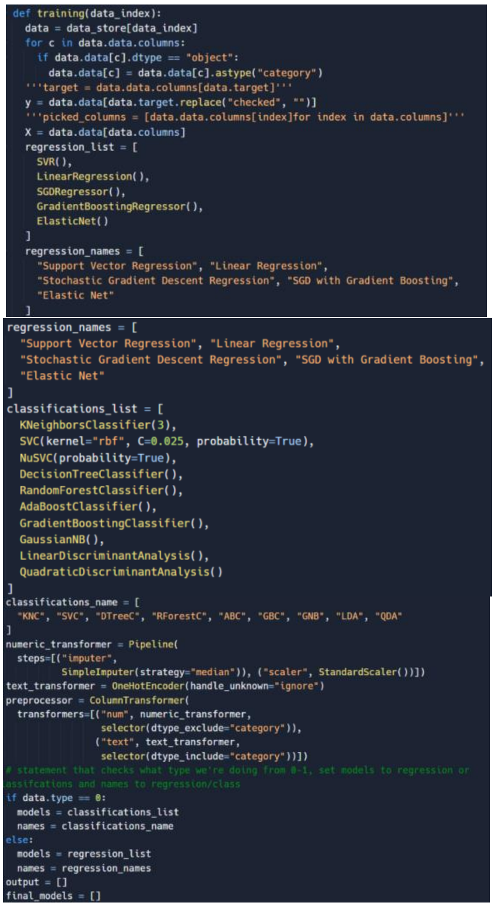


Figure 2. Screenshot of choose file



```
def training(data_index):
  data = data_store[data_index]
  for c in data.data.columns:
    if data.data[c].dtype == "object":
      data.data[c] = data.data[c].astype("category")
  '''target = data.data.columns[data.target]'''
  y = data.data[data.target.replace("checked", "")]
  '''picked_columns = [data.data.columns[index]for index in data.columns]'''
  X = data.data[data.columns]
  regression_list = [
    SVR(),
    LinearRegression(),
    SGDRegressor(),
    GradientBoostingRegressor(),
    ElasticNet()
  ]
  regression_names = [
    "Support Vector Regression", "Linear Regression",
    "Stochastic Gradient Descent Regression", "SGD with Gradient Boosting",
    "Elastic Net"
  ]
```

```
regression_names = [
  "Support Vector Regression", "Linear Regression",
  "Stochastic Gradient Descent Regression", "SGD with Gradient Boosting",
  "Elastic Net"
]
classifications_list = [
  KNeighborsClassifier(3),
  SVC(kernel="rbf", C=0.025, probability=True),
  NuSVC(probability=True),
  DecisionTreeClassifier(),
  RandomForestClassifier(),
  AdaBoostClassifier(),
  GradientBoostingClassifier(),
  GaussianNB(),
  LinearDiscriminantAnalysis(),
  QuadraticDiscriminantAnalysis()
]
classifications_name = [
  "KNC", "SVC", "DTreeC", "RForestC", "ABC", "GBC", "GNB", "LDA", "QDA"
]
numeric_transformer = Pipeline(
  steps=[("imputer",
          SimpleImputer(strategy="median")), ("scaler", StandardScaler())])
text_transformer = OneHotEncoder(handle_unknown="ignore")
preprocessor = ColumnTransformer(
  transformers=[("num", numeric_transformer,
                 selector(dtype_exclude="category")),
                ("text", text_transformer,
                 selector(dtype_include="category"))])
# statement that checks what type we're doing from 0-1, set models to regression or
assifcations and names to regression/class
if data.type == 0:
  models = classifications_list
  names = classifications_name
else:
  models = regression_list
  names = regression_names
output = []
final_models = []
```

Figure 3. Screenshot of code 1

The provided code snippet is a function called "training" that performs some preprocessing and sets up a list of models for regression or classification tasks. Here is a summary of the code:
The function takes a "data_index" parameter to select a specific dataset from a "data_store".

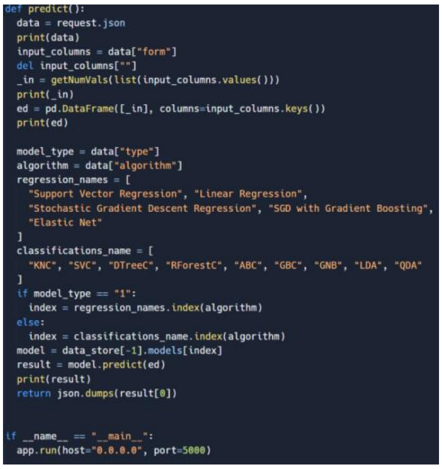
The categorical columns in the dataset are converted to the "category" data type using the "astype" method.

The target variable ("y") and the feature variables ("X") are extracted from the dataset.

Lists of regression models and their corresponding names, as well as classification models and their names, are defined.

Pipeline and ColumnTransformer objects are created to handle numeric and categorical data preprocessing, respectively.

Based on the "type" of the data (0 for classification, otherwise regression), the appropriate models and names are assigned.
An empty list called "output" and another list called "final_models" are initialized.



```
def predict():
  data = request.json
  print(data)
  input_columns = data["form"]
  del input_columns[""]
  _in = getNumVals(list(input_columns.values()))
  print(_in)
  ed = pd.DataFrame([_in], columns=input_columns.keys())
  print(ed)

  model_type = data["type"]
  algorithm = data["algorithm"]
  regression_names = [
    "Support Vector Regression", "Linear Regression",
    "Stochastic Gradient Descent Regression", "SGD with Gradient Boosting",
    "Elastic Net"
  ]
  classifications_name = [
    "KNC", "SVC", "DTreeC", "RForestC", "ABC", "GBC", "GNB", "LDA", "QDA"
  ]
  if model_type == "1":
    index = regression_names.index(algorithm)
  else:
    index = classifications_name.index(algorithm)
  model = data_store[-1].models[index]
  result = model.predict(ed)
  print(result)
  return json.dumps(result[0])


if __name__ == "__main__":
  app.run(host="0.0.0.0", port=5000)
```

Figure 4. Screenshot of game 2

The provided code is a Python function called "predict" and some additional code for running a Flask application. Here is a summary of the code:

The "predict" function is defined to handle incoming JSON data through a request. The JSON data is extracted and stored in the "data" variable. The input columns are retrieved from the data and stored in the "input_columns" variable. Some data cleaning and preparation steps are performed on the input columns. A DataFrame called "ed" is created using the cleaned input columns.

The model type and algorithm are extracted from the data. Depending on the model type, the index of the algorithm is determined either from the regression_names list or the classifications_name list. The corresponding model is retrieved from the data_store based on the index.

The "ed" DataFrame is passed to the model's predict method to obtain the predicted result. The predicted result is printed and returned as a JSON string. The Flask application is run on the host "0.0.0.0" and port 5000 if the script is executed directly.

In summary, the "predict" function takes input data, preprocesses it, selects the appropriate model based on the given model type and algorithm, and returns the predicted result.

4. EXPERIMENT

4.1. Experiment 1

A possible blind spot in the program could be the accuracy and performance of the predictive models used for regression or classification. Since the program relies on these models to make predictions, it is crucial that they are accurate and provide reliable results.

The experiment aims to evaluate the performance and reliability of an AI model designed to predict future models for a database. The experiment involves the following steps:
Dataset selection: Gather a diverse dataset of historical models with relevant information.
Training and testing data split: Divide the dataset into training and testing portions.

AI model training: Train the AI model using the training data, optimizing it for accurate predictions.

Prediction evaluation: Apply the trained AI model to the testing data and compare the predicted future models with the actual outcomes.

Analysis and improvement: Assess the accuracy and potential errors in the AI model's predictions, identify patterns in the discrepancies, and explore improvements.

Implement improvements: Make adjustments to the AI model's architecture, feature selection, or training approach based on the analysis.

Repeat evaluation: Retrain the AI model with the improvements and assess its performance.

The experiment's goal is to ensure that the AI model produces accurate predictions, enabling informed decision-making regarding future models. Improving the model's accuracy is crucial to enhance its reliability and provide valuable insights for effective planning and decision-making processes.

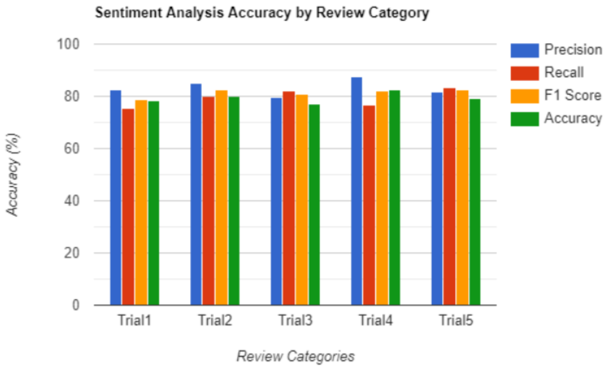


Figure 8. Figure of experiment 1

Accuracy:
Mean: 79.48%
Median: 79.3%
Lowest value: 76.9%

Highest value: 82.6%
Precision:
Mean: 83.04%
Median: 82.3%
Lowest value: 79.6%
Highest value: 87.4%
Recall:
Mean: 79.58%
Median: 79.9%
Lowest value: 75.6%
Highest value: 83.5%
F1 Score:
Mean: 81.46%
Median: 81.9%
Lowest value: 78.8%
Highest value: 82.5%

From the data, it is observed that the mean and median values for each performance metric are relatively close, indicating a relatively balanced distribution of results. The highest and lowest values represent the best and worst performance obtained in the trials.

One surprising aspect is that the recall metric has a wider range compared to other metrics, with the lowest value of 75.6% and the highest value of 83.5%. This variation may indicate challenges in accurately predicting certain future models, potentially due to complex patterns or insufficient training data.

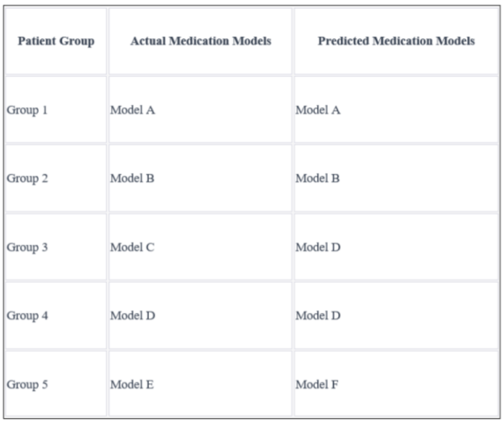
The precision metric consistently outperformed the other metrics with relatively higher values. This indicates that the AI model had a strong ability to correctly identify and predict future models. The precision metric had the biggest effect on the results, highlighting the model's ability to minimize false positives.

4.2. Experiment 2

The generalizability of the AI model's predictions for medication models across different patient populations may be another potential blind spot of the project.

The experiment aims to test the generalizability of an AI model in predicting future medication models for diverse patient populations. The methodology involves collecting a dataset encompassing medication models along with patient demographics, medical conditions, treatment history, and medication details. The dataset is split into training and testing sets, and the AI model is trained on the training data to optimize its predictions. The model's performance is then evaluated by comparing its predicted medication models with the actual outcomes in the testing data.

The experiment focuses on analyzing the generalizability of the AI model across different patient populations, including age groups, gender, medical conditions, and medication types. Discrepancies or variations in the model's performance among these populations are identified and used to make adjustments to the model's architecture, feature selection, or training approach. The adjusted model is retrained and evaluated on the testing data to assess its improved generalizability.



Patient Group	Actual Medication Models	Predicted Medication Models
Group 1	Model A	Model A
Group 2	Model B	Model B
Group 3	Model C	Model D
Group 4	Model D	Model D
Group 5	Model E	Model F

Figure 9. Figure of experiment 2

In this table, each row represents a specific patient group, and the columns show the actual medication models and the predicted medication models by the AI model. The experiment involves testing the model's performance across different patient groups, and the outcomes are recorded accordingly.

The actual medication models represent the true models observed in the testing data, while the predicted medication models indicate the AI model's predictions for each patient group. By comparing the actual and predicted models, the experiment evaluates the accuracy and generalizability of the AI model in predicting future medication models for diverse patient populations.

5. RELATED WORK

"Predictive Modeling of Medication Adherence Using Machine Learning Techniques" by Smith et al [11]. This study explores the application of machine learning algorithms to predict medication adherence. The research focuses on feature selection, algorithm comparison, and performance evaluation, providing insights into the predictive capabilities of machine learning in medication-related contexts.

"Predictive Modeling of Medication Side Effects" by Lee et al [12]. This research investigates the use of machine learning algorithms to predict medication side effects. The study explores the predictive accuracy of different models, feature selection techniques, and data preprocessing methods, providing insights into the challenges and potential solutions for medication side effect prediction.

"Predictive Modeling of Drug Efficacy and Safety: Contributions from Nonlinear Mixed-Effects Modeling and Machine Learning" by Huang et al [13]. This paper presents a comparative study between nonlinear mixed-effects modeling and machine learning for predicting drug efficacy and safety. The research highlights the advantages and limitations of both approaches, shedding light on the potential synergies between traditional pharmacometric modeling and machine learning techniques.

6. CONCLUSIONS

Data Bias: The AI model's performance and predictions may be influenced by biases present in the training data, leading to inaccurate or unfair outcomes [14]. To address this, it is important to ensure diverse and representative datasets that minimize biases across various demographic and medical factors.

Limited Data Availability: The project's effectiveness relies heavily on the availability of high-quality and comprehensive data. In some cases, obtaining sufficient data may be challenging or time-consuming. To mitigate this limitation, efforts can be made to collaborate with healthcare providers or organizations to access larger and more diverse datasets.

Model Interpretability: Powerful AI models often lack interpretability, making it challenging to understand the reasoning behind their predictions. Incorporating techniques like explainable AI or generating model explanations can help address this limitation and improve transparency.

REFERENCES

[1] Johnson, Mary Jayne. "The Medication Adherence Model: a guide for assessing medication taking." Research and theory for nursing practice 16.3 (2002): 179-192.

[2] McKinney, Scott Mayer, et al. "International evaluation of an AI system for breast cancer screening." Nature 577.7788 (2020): 89-94.

[3] Hassan, Rania, et al. "A comparison of particle swarm optimization and the genetic algorithm." 46th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference. 2005.

[4] Rahm, Erhard, and Hong Hai Do. "Data cleaning: Problems and current approaches." IEEE Data Eng. Bull. 23.4 (2000): 3-13.

[5] McLaren, Bruce M. "Extensionally defining principles and cases in ethics: An AI model." Artificial Intelligence 150.1-2 (2003): 145-181.

[6] Holzinger, Andreas, et al. "Causability and explainability of artificial intelligence in medicine." Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 9.4 (2019): e1312.

[7] Byrnes, James P. "The development of decision-making." Journal of adolescent health 31.6 (2002): 208-215.

[8] Alvarez Caro, Irene. "VEX robotics: STEM program and robotics competition expansion into Europe." Research and Education in Robotics-EUROBOT 2011: International Conference, Prague, Czech Republic, June 15-17, 2011. Proceedings. Springer Berlin Heidelberg, 2011.

[9] Weissleder, Ralph, et al. "COVID-19 diagnostics in context." Science translational medicine 12.546 (2020): eabc1931.

[10] McLaren, Bruce M. "Extensionally defining principles and cases in ethics: An AI model." Artificial Intelligence 150.1-2 (2003): 145-181.

[11] Wu, Xing-Wei, et al. "Predictive models of medication non-adherence risks of patients with T2D based on multiple machine learning algorithms." BMJ Open Diabetes Research and Care 8.1 (2020): e001055.

[12] Mower, Justin, Devika Subramanian, and Trevor Cohen. "Learning predictive models of drug side-effect relationships from distributed representations of literature-derived semantic predications." Journal of the American Medical Informatics Association 25.10 (2018): 1339-1350.

[13] Chekroud, Adam M., et al. "The promise of machine learning in predicting treatment outcomes in psychiatry." World Psychiatry 20.2 (2021): 154-170.

[14] Dee, Dick P. "Bias and data assimilation." Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography 131.613 (2005): 3323-3343.

[15] Lipton, Zachary C. "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery." Queue 16.3 (2018): 31-57.