**PRIDICTION OF ANTIMICROBAL RESISTANCE USING MACHINE LEARNING**

**By**

**MAROOF KHAN**

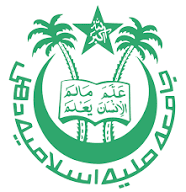
**(Enrolment No. 202201826)**

**SUBMITTED**

**In partial fulfilment of the requirement of the degree of**

**M.Sc (Bioinformatics)**

**To the**



**DEPARTMENT OF COMPUTER SCIENCE**

**FACULTY OF NATURAL SCIENCES**

**JAMIA MILLIA ISLAMIA**

**NEW DELHI**

**MAY,24**

**PRIDICTION OF ANTIMICROBAL RESISTANCE USING MACHINE LEARNING**

**By**

**MAROOF KHAN**

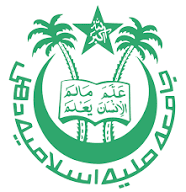
**(Enrolment No. 202201826)**

**SUBMITTED**

**In partial fulfilment of the requirement of the degree of**

**M.Sc (Bioinformatics)**

**To the**



**DEPARTMENT OF COMPUTER SCIENCE**

**FACULTY OF NATURAL SCIENCES**

**JAMIA MILLIA ISLAMIA**

**NEW DELHI**

**MAY,24**

**DECLARATION**

I, Maroof Khan, student of M.Sc. (Bioinformatics) hereby declare that the report entitled “**PRIDICTION OF ANTIMICROBAL RESISTANCE USING MACHINE LEARNING MODEL**” being submitted by me to the Department of Computer Science in the partial fulfilment of the requirement for the award of the degree is a record of the original project work carried out by me. It has not previously formed the basis for the award of any Degree/Diploma or other similar title or recognition. I declare further that I have also fulfilled the requirements of the submission of the report of the Project.

DATE: 15/05/2024 MAROOF KHAN

Roll No: 22MBI019

Enrollment:202201826

**CERTIFICATE**

On the basis of declaration made by my student **Mr Maroof Khan** student of MSc Bioinformatics 4th semester, this is to certify that the project report entitled “**PRIDICTION OF ANTIMICROBAL RESISTANCE BY USING MACHINE LEARNING MODEL**” submitted to the Department of Computer Science, Jamia Millia Islamia, New Delhi in the partial fulfilment of the requirement of the degree of Master of Science (Bioinformatics), is a faithful record of original work done under my co-guidance and co-supervision. The report has reached the requisite standards for submission. Dr **Rajendra Kumar** (Internal Supervisor) Associate Professor Department of Computer Science Jamia Millia Islamia (Central University) Head of Department Dr MONICA MEHROTRA Department of Computer Science Jamia Millia Islamia (Central University)

**Dr Rajendra Kumar**

(Internal Supervisor)

Assistant Professor

**Department of Computer Science**

**Jamia Millia Islamia**

**Dr MONICA MEHROTRA**

**Head of Department**

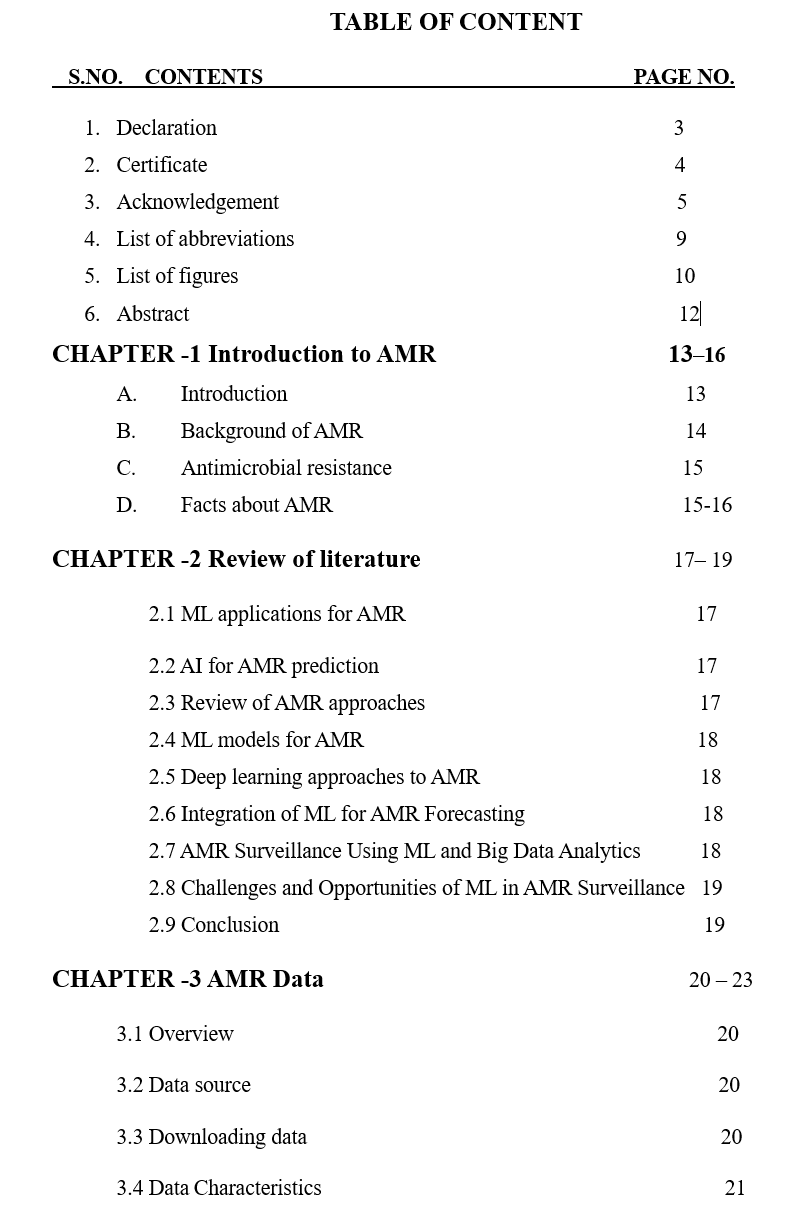
**Department of computer science**

**Jamia Millia Islamia**

**ACKNOWLEDGEMENT**

First and foremost, I bow my head before almighty God whose constant blessings and grace enabled me to complete this project work successfully and gave me courage to stand all. Acknowledgement is essential to fuelling passion and giving the team a real sense of progress that makes it all worthwhile. It is a sweet and short way to express gratitude. I take this opportunity to express my heartfelt thanks to all those people who have guided, supported and encouraged me to complete my work. I manifest my heartfelt sense of obligation to my INTERNAL SUPERVISOR **Dr RAJENDRA KUMAR**, Assistant Professor, Department of Computer science, Jamia Millia Islamia, New Delhi, for his proficient guidance, innovative ideas, inspiring suggestions, consistent effort, unfluctuating supervision, constant admonishment blended with compassion and serenity, unflagging enthusiasm during the entire course of my project work. It was his most co-operative and painstaking attitude which made this project a reality. His way of explanation of each and every concept, objectives has made me understand it in a broad way. I bow to his highness for accepting me as his student and allowing me to follow him. His encouragement has led me to finish my work in a proper organized way. His friendly way of talking has always insisted me to ask my doubts instantly. He has always been a source of motivation for me I am also grateful to my class teacher Prof. Suraiya Jabin and my friend Anam Fatima for their constant support, encouragement, and motivation. Their unwavering belief in my abilities, their constructive feedback, and their willingness to lend a helping hand have been a constant source of inspiration for me throughout this project. I would like to express my heartfelt gratitude to my family, who have been my pillar of strength throughout this journey. Their unwavering support, encouragement, and love have been a constant source of inspiration for me. I am deeply indebted to them for their support, patience, and understanding throughout this project. In summation, I am immensely grateful to everyone who has contributed to the success of my research project. I acknowledge and appreciate their invaluable contributions, support, and guidance throughout this journey.

**MAROOF KHAN**



3.4 Data Characteristics 17

3.4 List of AMR databases 17

3.5 Data collection 18

3.6 Data preprocessing 18

3.7 Data format 19

3.8 Final data 20

**CHAPTER – 4 Feature Extraction** 21 - 26

1.1 Feature extraction 21

1.2 DNABERT-2 working 24

1.3 Checking the model 25

1.4 Explanation of code 26

**CHAPTER – 5 Model Selection 27 - 29**

1.1 Introduction 27

1.2 working 27 – 28

1.3 Advantages of random forest 28

1.4 limitations 29

1.5 applications of random forest 29

1.6 conclusion 29

**CHAPTER - 6 Methodology 30 – 53**

6.1 Working pipeline 30

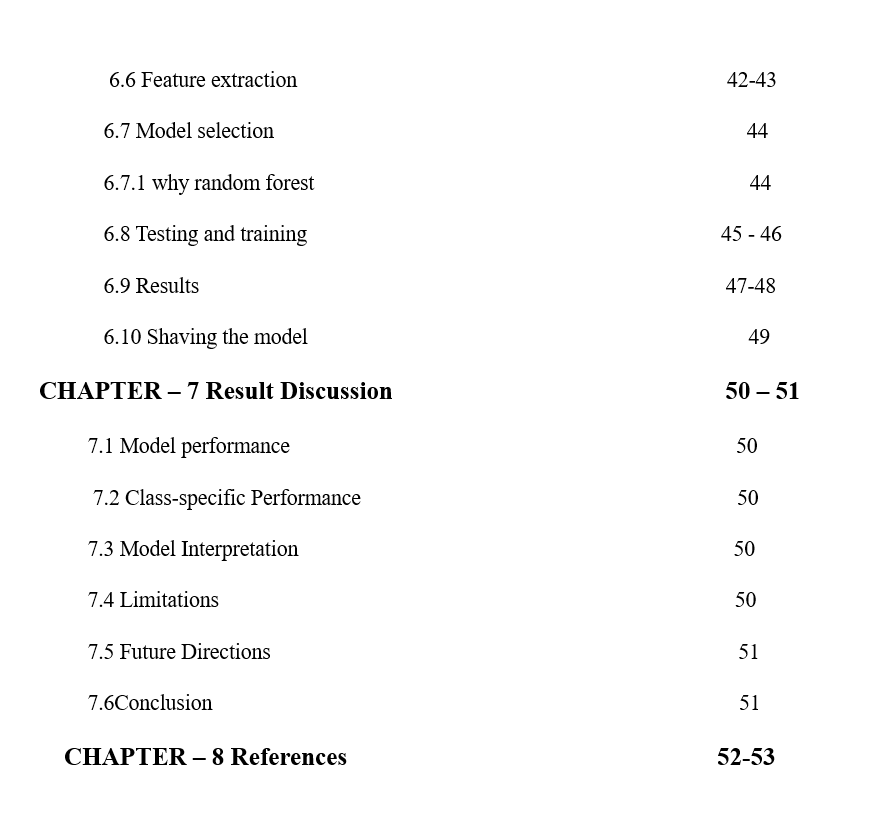
6.2 AMR database selection 31

6.3 Data extraction 32

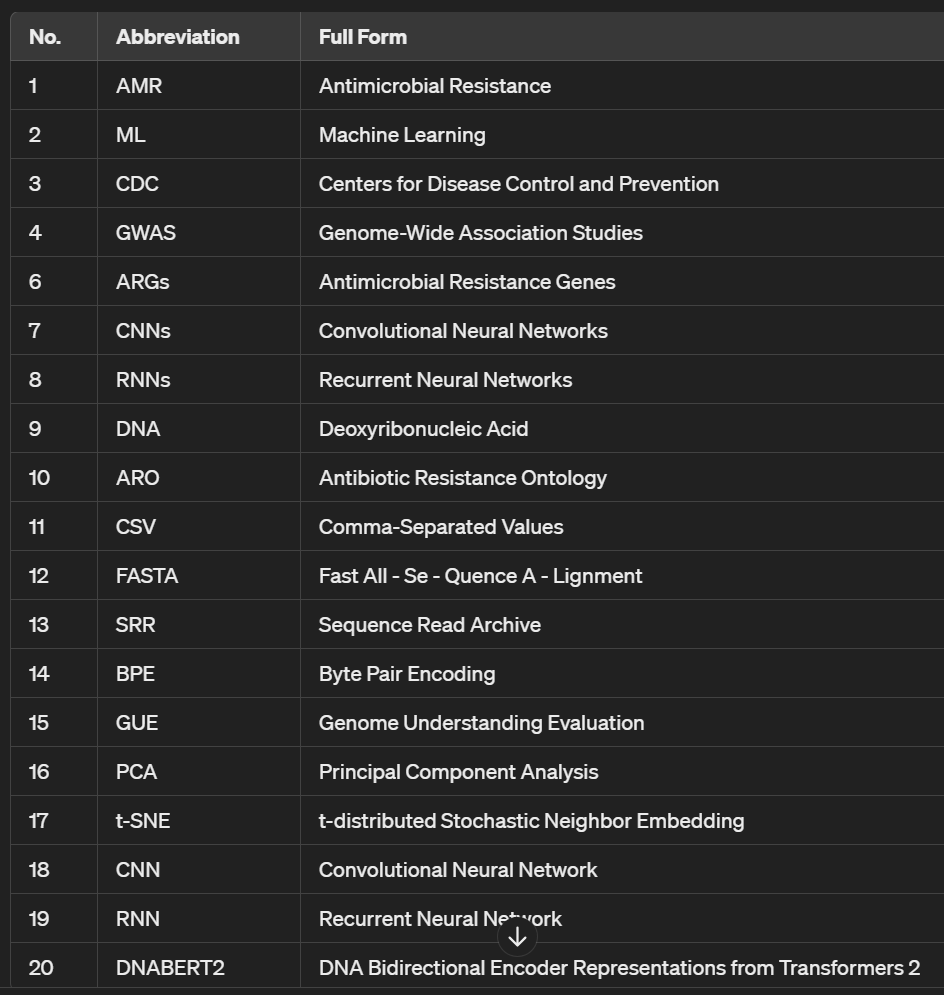
6.3 Data preprocessing 33

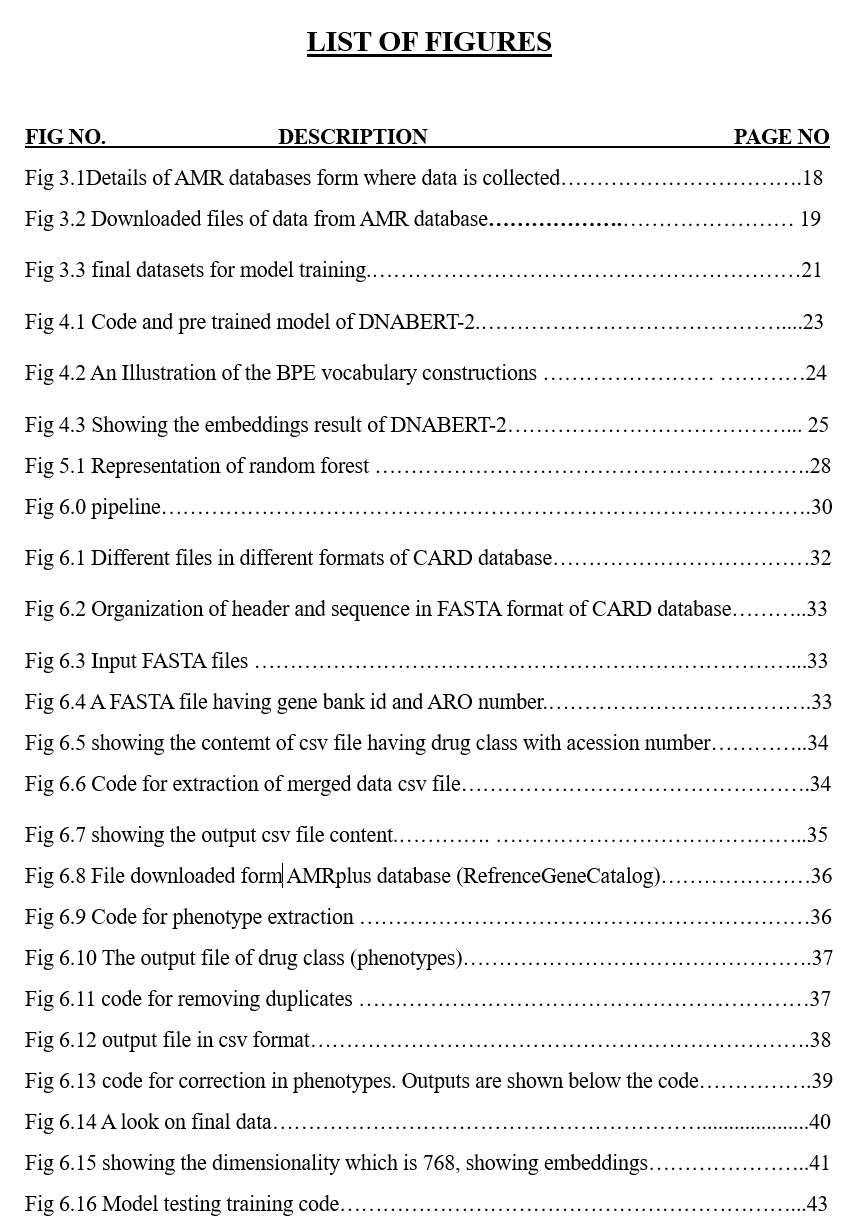
6.4 Feature extraction 34

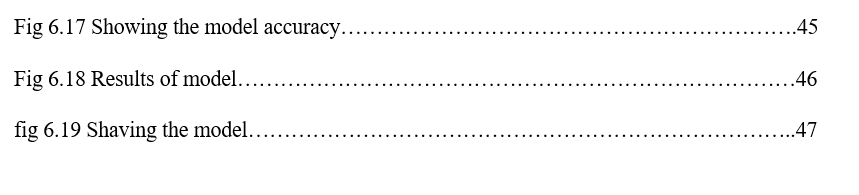
6.5 Model selection 37



**ABBRIVATION**

****

****

****

**ABSTRUCT**

This study delves into the critical domain of antimicrobial resistance (AMR) and aims to utilize machine learning (ML) techniques for predictive purposes. We utilize a comprehensive dataset comprising DNA sequences and their corresponding resistance phenotypic labels sourced from various publicly available AMR databases.

Our research begins with thorough data preprocessing, involving meticulous cleaning and merging of sequences to ensure data integrity. We then extract features using DNABERT-2, a sophisticated model designed for generating embeddings tailored specifically for DNA sequences. These embeddings serve as the foundation for subsequent predictive modeling.

The core of our predictive framework lies in employing Random Forest, a versatile ML algorithm known for its ability to handle high-dimensional data and complex decision boundaries. Through rigorous training, our model seeks to identify intricate patterns and relationships within the genomic data, offering valuable insights into AMR dynamics.

Following training, we meticulously evaluate the model's performance, assessing its accuracy in predicting AMR phenotypes across various antimicrobial classes. While our model demonstrates commendable overall accuracy, we scrutinize its performance at a class-specific level to identify areas for improvement.

In summary, our study underscores the importance of predictive modeling in addressing the challenge of antimicrobial resistance. By leveraging ML algorithms and rich genomic datasets, we aim to equip healthcare professionals with tools and insights to effectively combat AMR. Additionally, our research reflects our ongoing commitment to innovation and refinement as we strive to enhance predictive capabilities and deepen our understanding of AMR dynamics.

**CHAPTER–1**

**INTRODUCTION**

1. **Introduction**

Antimicrobial resistance (AMR) poses a critical threat to public health worldwide, compromising the effectiveness of conventional antibiotic treatments and escalating the risk of infections caused by drug-resistant bacteria and other microbes. As bacteria evolve to withstand the effects of antibiotics, innovative approaches to combat this phenomenon are urgently needed. In recent years, the integration of machine learning (ML) techniques into medical research has emerged as a promising strategy for predicting antimicrobial resistance with greater accuracy and efficiency. Our project aims to delve into the realm of AMR prediction by harnessing the power of ML algorithms. By leveraging computational models trained on vast datasets containing genomic sequences, clinical metadata, and antimicrobial susceptibility profiles, we seek to elucidate the intricate relationship between bacterial genetics and antibiotic resistance phenotypes. [1]

The urgency of addressing antimicrobial resistance is underscored by its profound impact on patient outcomes, healthcare costs, and global morbidity and mortality rates. Estimates from the US Centers for Disease Control and Prevention (CDC) reveal that over two million people in the US alone are affected by antibiotic-resistant infections annually, resulting in at least 23,000 deaths. The World Health Organization (WHO) has recognized the gravity of this issue and has implemented initiatives such as the WHO Global Strategy for Containment of Antimicrobial Resistance to mitigate its spread. To combat this crisis effectively, continuous research and innovation in healthcare are imperative. Improving policies on antimicrobial use and enhancing diagnostic capabilities are crucial steps towards mitigating the challenges posed by AMR.

Machine learning (ML) has gained prominence in antimicrobial resistance research, offering rapid and accurate predictions of AMR phenotypes from sequenced bacterial genomes. ML methods, such as those used in genome-wide association studies (GWAS), have demonstrated high predictive value and support surveillance and precise diagnosis of AMR. Despite the promise of ML, its adoption in clinical and public health settings remains limited [2].

**B. BACKGROUND OF AMR**

The dawn of the modern antibiotic era can be traced back to the discovery of salvarsan and neosalvarsan by Paul Ehrlich in 1910. These synthetic prodrugs were developed to treat syphilis caused by Treponema pallidum. Later, prontosil, a sulfonamide prodrug discovered by bacteriologist Gerhard Domagk, gradually replaced salvarsan. Selman Waksman, an American microbiologist and biochemist, made significant contributions in the 1930s by systematically evaluating microbes in soil and discovering antibiotics from filamentous actinomycetes. Streptomycin, one of the antibiotics he discovered, became widely used in the treatment of tuberculosis. Waksman defined an antibiotic as "a compound made by a microbe to destroy other microbes." [3]

Another milestone in antibiotic discovery was the discovery of penicillin by Sir Alexander Fleming in 1928. Penicillin, derived from the mold Penicillium, specifically Penicillium chrysogenum, marked the beginning of the golden era of antibiotic discovery, which peaked until the mid-1950s. This period, from the 1940s to the 1960s, is often referred to as the "Golden Age," during which most of the antibiotics still in use today were discovered.

However, since the "Golden Age," there has been a gradual decline in antibiotic discovery, accompanied by the evolution of drug-resistant pathogens. Bacterial resistance to antibiotics has been recognized almost since the dawn of the antibiotic era. For example, several years before the introduction of penicillin as a therapeutic agent in 1940, the first penicillin-resistant Staphylococcus strain was described. Methicillin, introduced in 1959 as the first semisynthetic penicillinase-resistant penicillin, was soon met with resistance, with the first methicillin-resistant Staphylococcus strain reported in 1960.[4]

In subsequent years, the emergence of resistance to other antibiotics further underscored the challenge of antimicrobial resistance. Vancomycin, introduced in 1958 as a rescue drug for treating infections caused by methicillin-resistant Staphylococci, eventually faced resistance itself. Similarly, cephalosporins, tetracycline, levofloxacin, and carbapenem, among others, have all encountered varying degrees of resistance since their introduction.

The timeline of antibiotic discovery reveals that new classes of antibiotics were produced by pharmaceutical industries primarily from 1960 to 1980, with a dramatic decrease in discovery thereafter. This disproportionate ratio between drug-resistant pathogens and the number of available antibiotics has raised concerns about an imminent post antibiotic era.

**C. ANTIMICROBIAL RESISTANCE**

Antimicrobials, encompassing antibiotics, antivirals, antifungals, and antiparasitic drugs, are employed to prevent and manage infectious ailments in humans, animals, and plants. Antimicrobial Resistance (AMR) arises when bacteria, viruses, fungi, and parasites cease to react to antimicrobial medications. This resistance renders antibiotics and other antimicrobial treatments ineffective, making infections challenging or impossible to address. This, in turn, heightens the risks associated with disease transmission, severe illness, disability, and mortality. AMR is an inherent phenomenon that occurs gradually due to genetic alterations in pathogens. However, human activities, particularly the improper use and excessive administration of antimicrobials for treating, preventing, or controlling infection in humans, animals, and plants, accelerate its emergence and propagation.

**D. FACTS ABOUT ANTIMICROBIAL RESISTANCE**

Antimicrobial resistance (AMR) stands as a paramount global challenge to public health and development. In 2019, it's estimated that bacterial AMR directly caused 1.27 million fatalities worldwide and contributed to 4.95 million deaths. The primary catalysts for the emergence of drug-resistant pathogens are the improper utilization and excessive application of antimicrobials in humans, animals, and plants.[5]

AMR affects nations across the globe, spanning all economic strata. Poverty and inequality exacerbate its drivers and consequences, with low- and middle-income countries bearing the brunt of its impact. AMR jeopardizes the advancements made in modern medicine, rendering infections more difficult to treat and rendering medical procedures like surgery, cesarean sections, and cancer chemotherapy considerably riskier.

The world confronts a crisis concerning the availability of antibiotics and access to them. There's a deficiency in research and development efforts to counter rising levels of resistance, necessitating urgent measures to ensure fair access to both new and existing vaccines, diagnostics, and medicines. Apart from the loss of life and disability, AMR incurs significant economic burdens. The World Bank projects that by 2050, AMR could result in an additional $1 trillion in healthcare expenses and annual GDP losses ranging from $1 trillion to $3.4 trillion by 2030. [6]

Key strategies to tackle AMR in human health encompass preventing infections altogether, even if it may entail curtailing the inappropriate use of antimicrobials; guaranteeing universal access to precise diagnosis and suitable treatment of infections; and fostering strategic information dissemination and innovation. This includes surveillance efforts tracking AMR and antimicrobial consumption, as well as research and development endeavors for novel vaccines, diagnostics, and medications

**CHAPTER – 2**

**Review of Literature**

**2.1 Machine Learning Applications for Antimicrobial Resistance (AMR) Prediction and Surveillance**

Antimicrobial resistance (AMR) poses a significant global health threat, necessitating effective surveillance strategies to monitor its spread and inform public health interventions. Recent advancements in machine learning (ML) offer promising avenues for enhancing AMR surveillance through the analysis of large and heterogeneous datasets. This literature review synthesizes insights from several studies exploring the application of ML techniques in AMR prediction and surveillance.[7]

**2.2"Artificial Intelligence for Antimicrobial Resistance Prediction: A Systematic Review"**

This systematic review explores the role of artificial intelligence (AI), including ML, in predicting antimicrobial resistance. It synthesizes findings from studies employing AI techniques such as neural networks, genetic algorithms, and decision trees. Key themes include the integration of diverse data sources, the development of interpretable models, and the deployment of AI in real-world healthcare settings. The review underscores the potential of AI to revolutionize AMR prediction and surveillance but acknowledges the need for robust validation and regulatory frameworks.[8]

**2.3 "Machine Learning Models for Predicting Antimicrobial Resistance Genes"**

This study focuses on the development of machine learning models specifically tailored for predicting antimicrobial resistance genes (ARGs). By analyzing genomic data from bacterial isolates, researchers trained and validated ML algorithms to identify genetic markers associated with antibiotic resistance. The findings underscore the potential of ML in genomic surveillance and its role in guiding antibiotic stewardship practices.

**2.4 "Deep Learning Approaches for Antimicrobial Susceptibility Prediction"**

This paper explores the application of deep learning techniques, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), for predicting antimicrobial susceptibility. By leveraging large-scale genomic and phenotypic datasets, deep learning models demonstrate promising performance in predicting bacterial susceptibility profiles to various antibiotics. The study highlights the importance of robust data curation and model optimization for achieving accurate predictions. [10]

**2.5 "Integration of Machine Learning and Epidemiological Models for AMR Forecasting"**

This research investigates the integration of machine learning with epidemiological models to forecast antimicrobial resistance trends. By combining ML algorithms with mathematical models of bacterial transmission dynamics, researchers aim to predict future AMR trajectories and assess the impact of interventions. The study emphasizes the value of interdisciplinary collaboration in developing predictive models that inform public health decision-making.[9]

**2.6 "****Real-Time Surveillance of AMR Using Machine Learning and Big Data Analytics"**

In this study, researchers propose a framework for real-time surveillance of antimicrobial resistance leveraging machine learning and big data analytics. By integrating data streams from electronic health records, microbiology laboratories, and environmental monitoring systems, the framework enables early detection of emerging resistance patterns and timely intervention. The research highlights the potential of ML-based surveillance systems in enhancing antimicrobial stewardship efforts and minimizing the spread of resistant pathogens.[11]

**2.7 "****Challenges and Opportunities in Machine Learning-Assisted AMR Surveillance"**

This review paper discusses the challenges and opportunities in machine learning-assisted antimicrobial resistance surveillance. Key challenges include data privacy concerns, algorithm bias, and the need for standardized reporting frameworks. On the other hand, opportunities lie in the integration of diverse data sources, the development of scalable ML pipelines, and the adoption of open-access platforms for sharing data and models. The review calls for collaborative efforts among stakeholders to address these challenges and harness the full potential of ML in AMR surveillance.[12]

**2.8 Conclusion**

In conclusion, the literature highlights a burgeoning interest in utilizing machine learning to tackle the pressing issue of antimicrobial resistance (AMR). From predicting resistance genes to tracking resistance trends in real-time, machine learning offers versatile solutions for combating this global health threat. However, challenges such as data heterogeneity and model interpretability persist, necessitating interdisciplinary collaborations and standardized methodologies. By addressing these obstacles and integrating advanced computational techniques into existing surveillance frameworks, we can advance our understanding of AMR dynamics and inform targeted interventions, ultimately safeguarding public health worldwide.

**CHAPTER - 3**

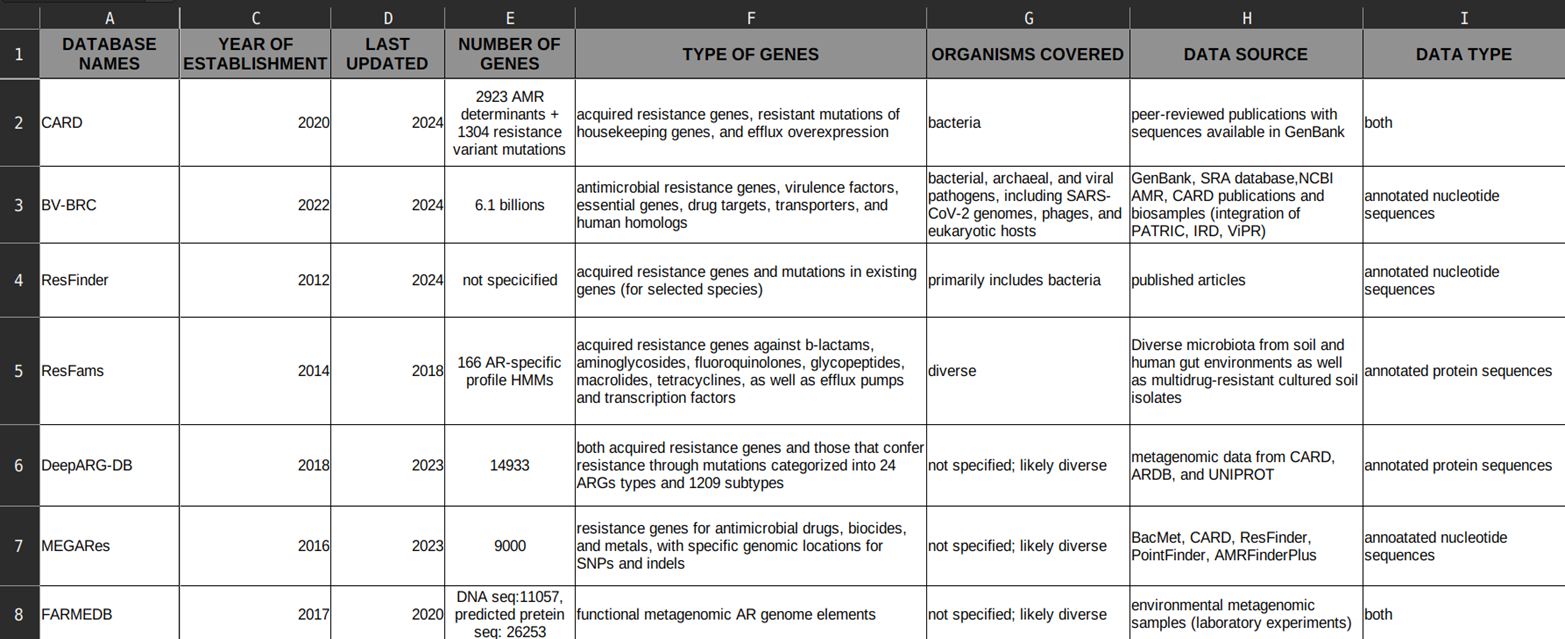
**AMR – DATA**

**3.1 Overview:**

The dataset used in our project consists of DNA sequences with their corresponding resistance phenotypic labels. The aim of our project is to develop a model that can predict the names of drugs to which the genome of a species show resistance.

**3.2 Data Source**

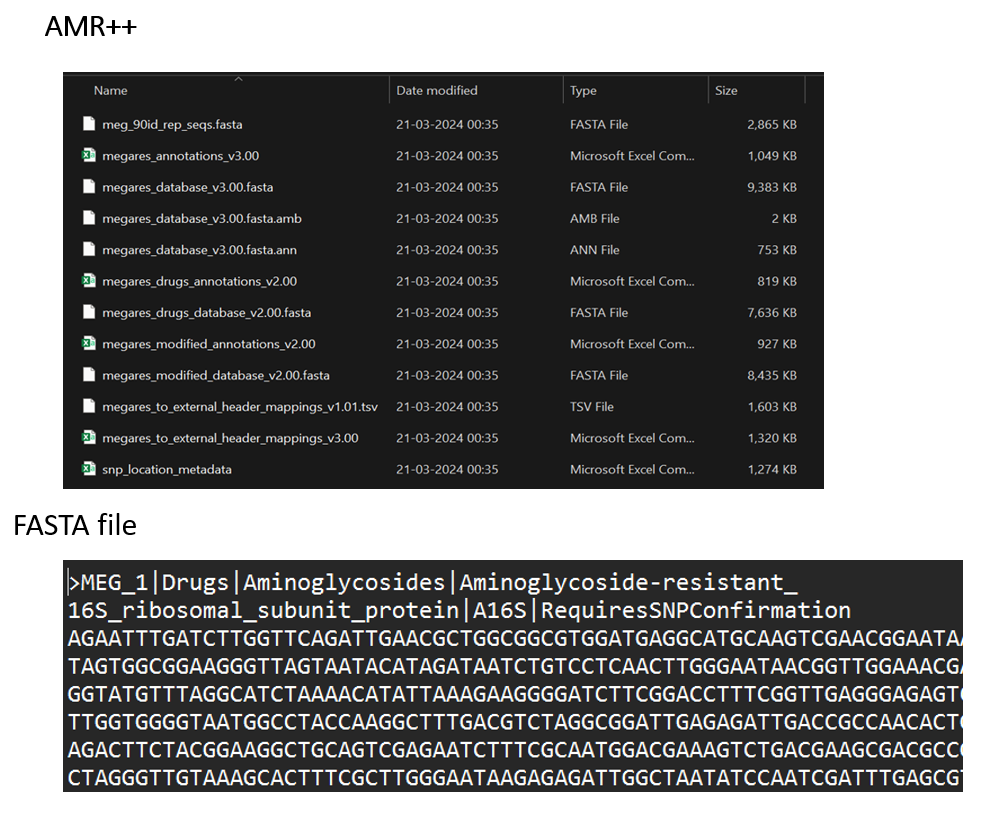
The datasets were obtained from different publicly available database. These databases contain curated collection of accession number, ARO number, gene names, drug class to which the gene is resistant and other phenotypic details extracted from different sources. This data is of different pathogens including bacteria, viruses, and eukaryotes etc. Details of databases are given in fig 3.1.



**Fig 3.1-** **Details of AMR databases form where data is collected.**

**3.3 Downloading data**:

Downloaded files consistof other files these files are in different formats. which contains header section in which the details of the sequence and name of the drug, gene name and drug class to which the gene sequence is resistant is given. Different databases have header section which contains different details of the sequence and resistance. The downloaded files of some databases are shown in fig 3.2

****

**Fig 3.2 Downloaded files of data from AMR database**

**3.4 Data** **Characteristics:**

The data is collected from 12 AMR databases The dataset comprises DNA sequences, each associated with gene ids, SRR no, ARO, gene name, gene sequence, organism name, organism genome id.

The data downloaded from AMR databases contains different files which contains different info like FASTA sequence, SRR no, antibiotic resistance ontology category, various annotation files and sequence files in different format files other than FASTA.

**3.5 List of all AMR databases used for data collection**

|  |
| --- |
| **1. CARD [13]**  **2. BV-BRC [14]**  **3. ResFinder [15]**  **4. ResFams [16]**  **5. AMR++ [17]**  **6. DeepARG- [18]**  **7. MEG [19]**  **8. FARMEDB [20]**  **9. PointFinder [21]**  **10. DARO/AMRFinderPlus [22]**  **11. SARG [23]** |

**3.5 Data collection:**

The data is fetched by using accession number and ARO number taken form the downloaded files of databases. The data which is fetched is sequence and the phenotypes (resistant drugs) the fetched data of all the databases are stored in a single excel file.

**3.6 Data preprocessing:**

Before model training, the dataset underwent preprocessing steps to ensure compatibility with the machine learning algorithms. Data preprocessing is very important step before using it in model training. data preprocessing involves cleaning of data by handling missing values, correcting errors, and addressing outliers to ensure its accuracy. Additionally, it transforms raw data into a suitable format for model training, which involve scaling features, encoding categorical variables, or normalizing data. Moreover, preprocessing includes feature engineering to create new features or select relevant ones, and data reduction techniques like dimensionality reduction for large datasets. Ultimately, it improves the performance of machine learning models by providing them with cleaner and more informative input data.

As our data is sequence and phenotype datathat’s whySequences from these databases were merged into a single file and duplicate or redundant sequences and null vales were removed.

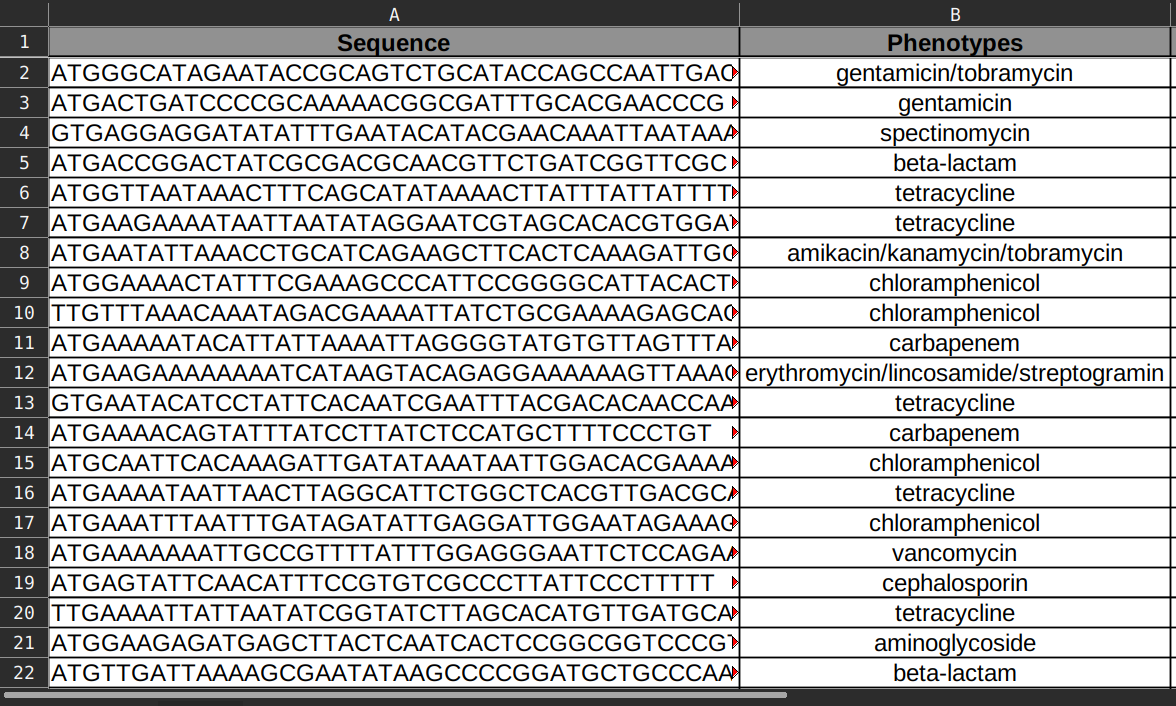
Using the GenBank id and the metadata from the databases (AMR+), labelling was done to their respective antibiotic resistance.

**3.7 Data Format:**

The datasets taken from various AMR database are merged into a single CSV file format, after downloading the datasets they are arranged in such a way that the file contain two columns one is sequence column and other is phenotype column.

Sequence: This column contains the DNA sequences represented as strings of nucleotide bases (A, T, C, and G) in FASTA format of different lengths.

Phenotype: This column contains the corresponding phenotypic labels associated with each DNA sequence. Phenotypes are categorized into different classes representing various drugs, this column contains multiple drug names for same sequence as multiple drugs may resist one gene. A look of final datasets is given in fig 3.3



**Fig 3.3 Final datasets for model training.**

**CHAPTER - 4**

**Feature Extraction**

**4.1 Feature Extraction overview**

In Feature extraction we convert the raw data into a set of features suitable for model. It is very important in machine learning, signal processing, and natural language processing.

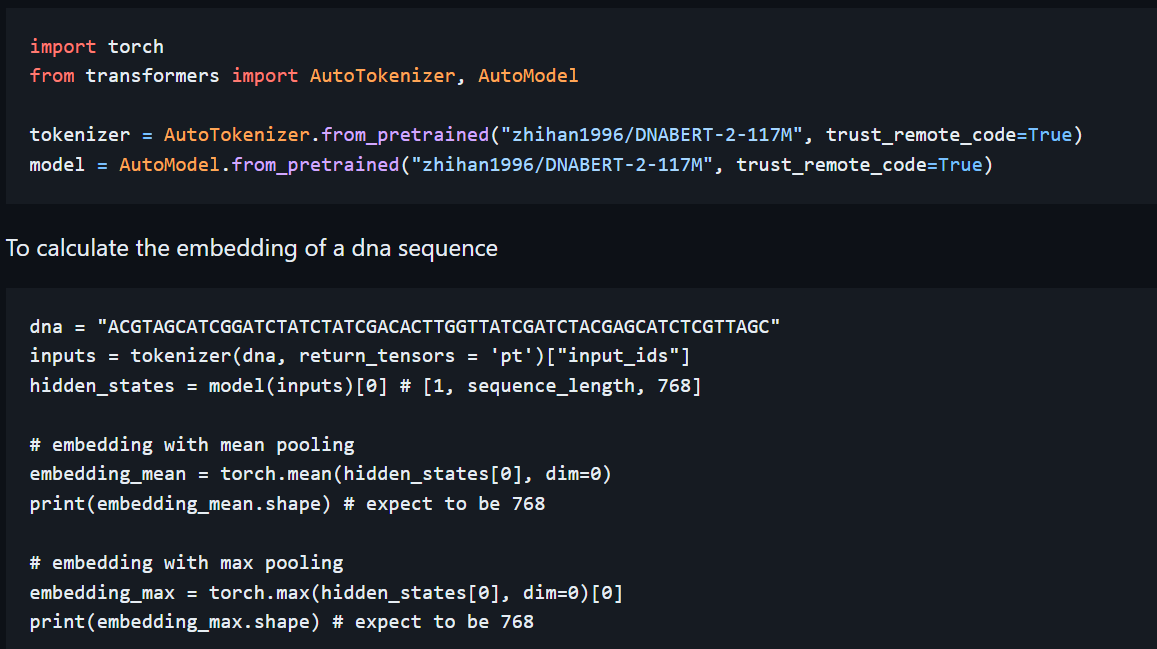
Here are some points why feature extraction is important in our project

* **Dimension Reduction**: Feature extraction cuts down the number of features in data. Raw data has many irrelevant or redundant features. Extracting only relevant ones can make models more effective and efficient.
* **Increase Model Performance**: Extracting meaningful features can improve model performance. Relevant features capture key data patterns, helping models learn and generalize better.
* **Noise Reduction**: Feature extraction can filter out noise and irrelevant data. Focusing on informative features minimizes the impact of noise on model performance.
* **Easy to Interpret**: Extracted features are often easier to understand than raw data. This makes it simpler to interpret data patterns.
* **Model Compatibility**: Some models require specific data formats. Feature extraction prepares data to meet these requirements, improving compatibility and performance of model.

Transform raw genomic sequences into informative feature representations suitable for predictive modeling. DNABERT2 generates dense vector representations, or embeddings, of fixed length from input DNA sequences.

DNABERT-2 is used for the feature extraction step, to capture the intricate genomic patterns associated with antimicrobial resistance. DNABERT2 is a foundation model trained on a large-scale, multi-species genome. It excels in 28 tasks of the GUE benchmark and introduces improvements like BPE tokenization and ALiBi positional embedding. DNABERT2 generates embeddings, which are dense vector representations, for DNA sequences. Embeddings are used as features in a machine learning model to predict antimicrobial resistance. The embeddings provide a rich representation of the DNA sequences, which can improve the accuracy of AMR prediction.

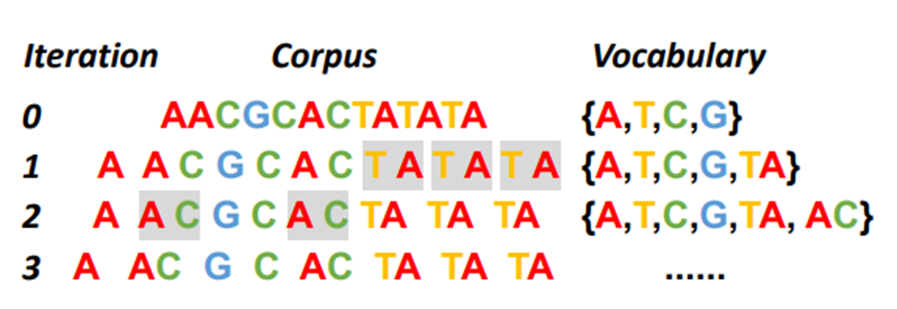
Understanding the complex language of the genome is a big challenge in biology. Scientists have made progress using models like DNABERT and Nucleotide Transformer, which understand genetic information. Typically, they use k-mers, which are short sequences of DNA letters (A, T, C, and G), as the basic units of the genome's language. However, this approach has some drawbacks—it's not very efficient for computers, and it can waste a lot of data. using Byte Pair Encoding (BPE). Instead of fixed-length k-mers, BPE compresses the data by finding common patterns in the genome and creating tokens based on those patterns. This makes the process faster and more efficient [26]

DNABERT-2, a new model that uses BPE and other tricks to handle long stretches of genetic code more effectively. We also noticed that there's no standard way to test how well these models understand genomes. So, the Genome Understanding Evaluation (GUE), a big dataset with tasks covering different aspects of genetics. In tests using GUE, DNABERT-2 performed as well as the best existing model but was much faster and needed fewer resources. Compared to the older version, DNABERT-2 was three times more efficient and improved accuracy on most tasks.

**Fig** **4.1 Code and pre trained model of DNABERT-2.** [27]

**4.2 DNABERT-2 working**

DNABERT-2 uses a method called Sentence Piece with Byte Pair Encoding (BPE) to break down DNA sequences into smaller parts for analysis. Sentence Piece is a tool that can handle different languages and doesn't rely on predefined word or sentence structures. This is important for genetic sequences because they don't follow traditional language rules.



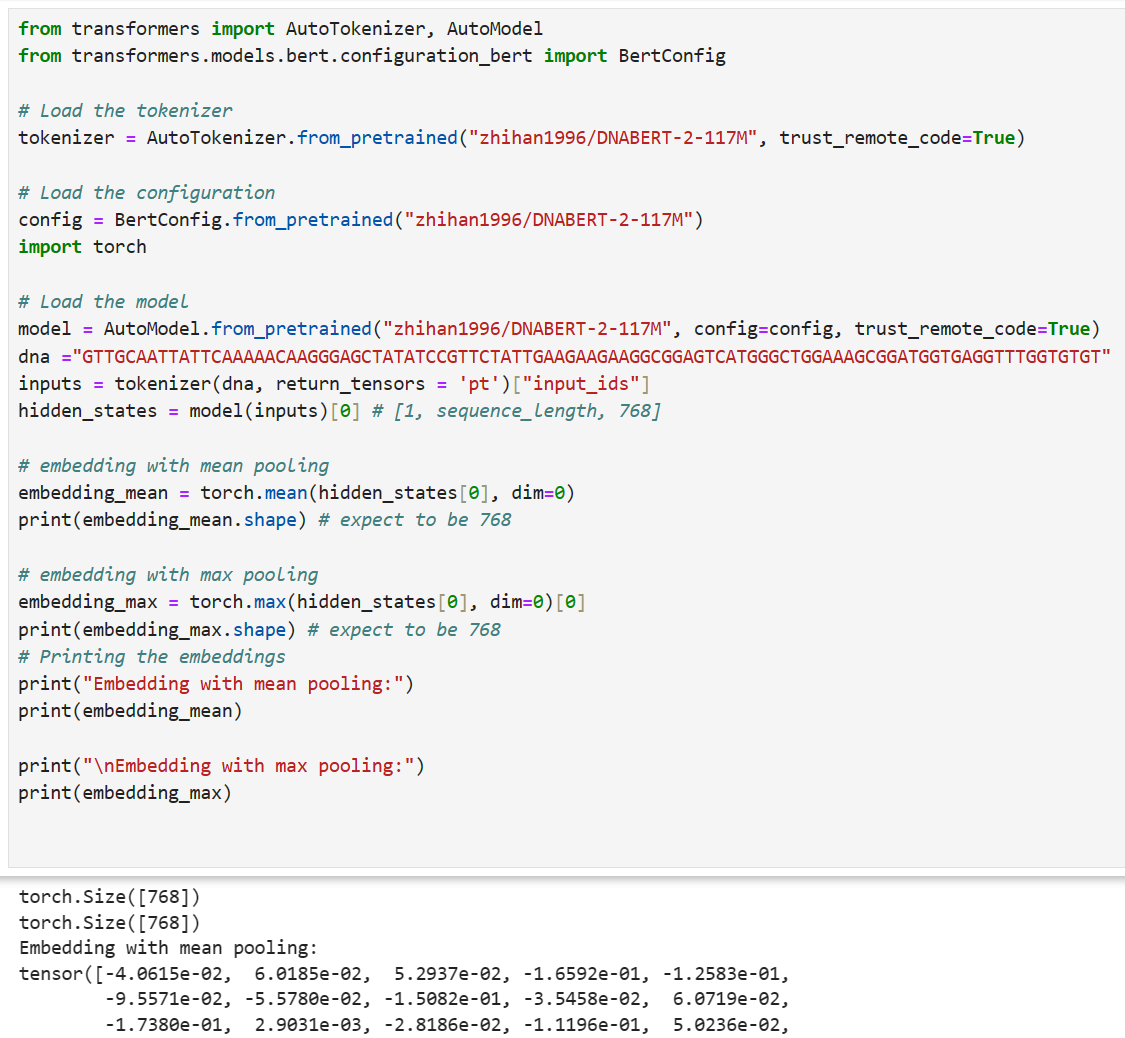
**Fig 4.2 An Illustration of the BPE vocabulary constructions [29]**

BPE, on the other hand, is a technique used in language processing to break down words into smaller units. It creates a list of common parts based on how often they appear together. For example, in a DNA sequence, it might notice that "TA" often occurs together. So, it treats "TA" as a single unit.

Imagine we have a big collection of DNA sequences. We start with a list of all the individual letters (like A, T, C, G). Then, we look for the most common pairs of letters (like "TA"). We add these pairs to our list and replace all occurrences of them in our sequences with the new pair. We keep doing this until we have enough pairs in our list.

The size of the final list is important because it determines how finely we break down the sequences. This process helps us analyze DNA sequences more efficiently and accurately.[28]

**4.3 Checking DNABERT-2**

****

**Fig-4.3 Showing the embeddings result of DNABERT-2**

The output is the embedding obtained from the DNABERT-2 model using mean pooling. This embedding is a vector of size 768, representing the learned features of the input DNA sequence. Each element in the vector corresponds to a specific feature that the model has learned from the sequence.

To interpret this result, we can analyze the values in the embedding vector. Each value represents the importance or presence of a specific feature in the input sequence. For example, positive values indicate the presence of certain features, while negative values indicate the absence or suppression of other features.

**4.****4 Explanation of code [27]**

Import Libraries: The code begins by importing necessary libraries, including the **AutoTokenizer, AutoModel**, and **BertConfig** from the Transformers library, as well as the torch library for working with tensors.

Load Tokenizer and Model Configuration: It loads the tokenizer and model configuration for DNABERT-2 using the **AutoTokenizer.from pretrained ()** and **BertConfig.from pretrained ()** functions, respectively. These functions fetch the pre-trained tokenizer and configuration from the specified **model name ("zhihan1996/DNABERT-2-117M")** hosted on the Hugging Face model hub.

* **Load Model:** The code then loads the DNABERT-2 model using the AutoModel. from pretrained () function. It specifies the configuration obtained earlier and sets trust remote code=True to ensure safe loading of the model from the remote repository.
* **Tokenize DNA Sequence**: Next, it tokenizes the input DNA sequence (dna) using the tokenizer obtained earlier. The tokenizer () function converts the DNA sequence into input IDs suitable for the model, and return ensors ='pt' specifies that PyTorch tensors should be returned.
* **Obtain Hidden States**: The code passes the tokenized input through the DNABERT-2 model to obtain hidden states. These hidden states represent the learned features of the input DNA sequence. The resulting tensor (hidden states) has dimensions [1, sequence length, 768], where sequence length is the length of the input sequence and 768 represents the size of the embedding space.
* **Compute Embeddings:** It computes two types of embeddings from the hidden states: mean pooling and max pooling. Mean pooling calculates the average value along the sequence dimension, resulting in a single embedding vector (embedding mean) of size 768. Max pooling selects the maximum value along the sequence dimension, also yielding an embedding vector (embedding max) of size 768.
* **Print Embeddings:** Finally, the code prints out the computed embeddings for visualization and analysis.

**CHAPTER -5**

**MODEL SELECTION**

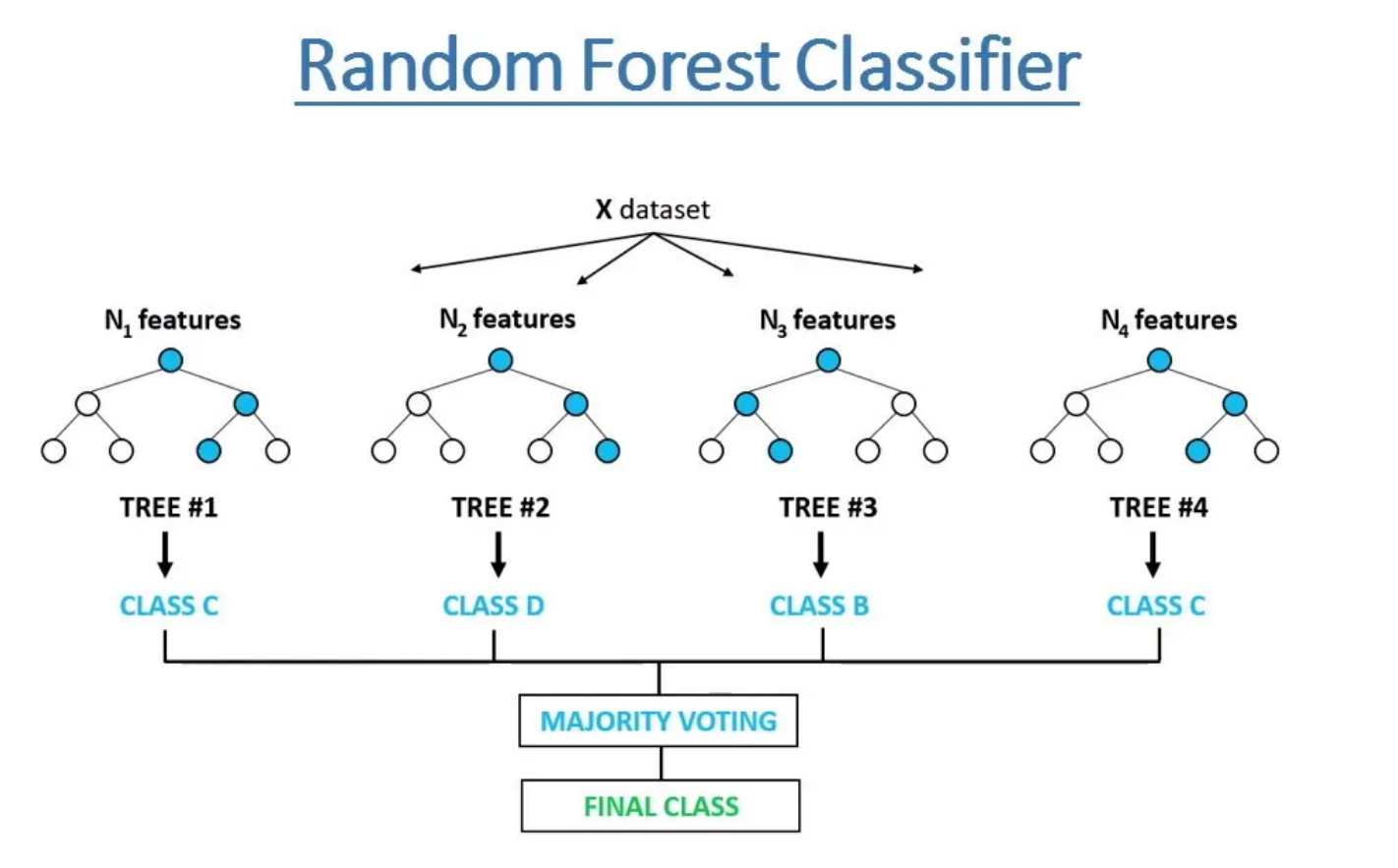
The training step involves training ML models on the extracted features to learn patterns and relationships between genomic data and AMR phenotypes.

**5.1 Introduction**

Random Forest is a versatile machine learning algorithm that belongs to the ensemble learning family. It is widely used for both classification and regression tasks due to its robustness and high performance. Random Forest is particularly well-suited for handling high-dimensional data and complex decision boundaries.

**5.2 Random Forest working [30]**

* **Decision Trees:** Random Forest is composed of an ensemble of decision trees, where each tree is built using a subset of the training data and a random selection of features. Decision trees are simple models that partition the feature space into regions, making predictions based on majority voting or averaging at the leaf nodes. It combines the opinions of many “trees” (individual models) to make better predictions, creating a more robust and accurate overall model.
* **Bootstrapping:** Random Forest employs a technique called bootstrapping to create multiple subsets of the training data with replacement. Each decision tree in the ensemble is trained on a different bootstrap sample, ensuring diversity among the trees.
* **Random Feature Selection**: At each split in the decision tree, a random subset of features is considered for splitting. This randomness helps decorrelate the individual trees in the ensemble, reducing the risk of overfitting and improving generalization performance.
* **Voting or Averaging:** In classification tasks, the final prediction of the Random Forest is determined by majority voting among the individual trees. In regression tasks, predictions are averaged across all trees to obtain the final output.



**Fig 5.1 Representation of random forest** [31]

**5.3 Advantages of Random Forest**

* **Robustness**: Random Forest is robust to overfitting, thanks to its ensemble nature and the averaging of predictions from multiple trees.
* **Scalability**: It can handle large-scale datasets with high dimensionality efficiently, making it suitable for a wide range of applications.
* **Feature Importance:** Random Forest provides a measure of feature importance, allowing users to identify the most informative features for prediction.
* **Non-linearity:** It can capture complex non-linear relationships in the data, making it suitable for tasks with intricate decision boundaries.

**5.4 Limitations of Random Forest**

* **Interpretability**: While Random Forest provides insights into feature importance, the individual decision trees in the ensemble may be difficult to interpret. Computational
* **Complexity:** Training a large number of decision trees in the ensemble can be computationally intensive, especially for large datasets. Hyperparameter Tuning: Random Forest requires careful tuning of hyperparameters, such as the number of trees and the maximum depth of each tree, to optimize performance.

**5.5 Applications of Random Forest**

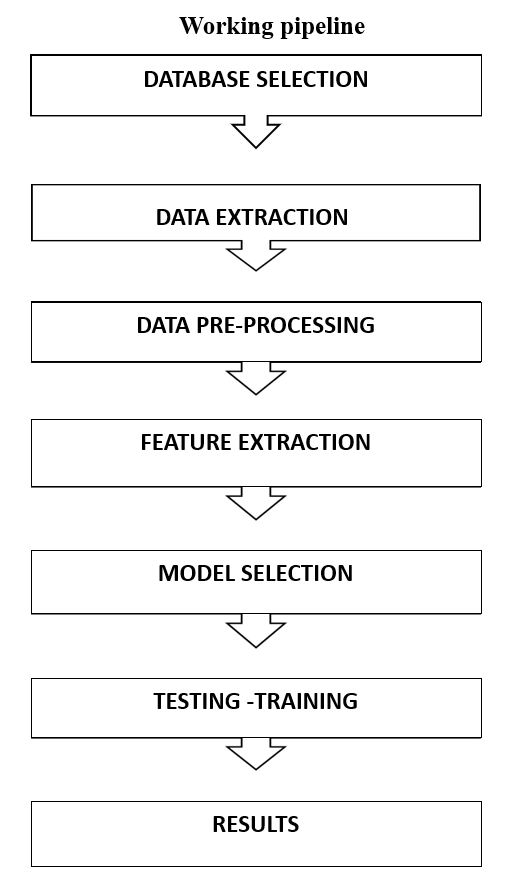
* **Classification**: Random Forest is widely used for classification tasks, including spam detection, disease diagnosis, and sentiment analysis.
* **Regression**: It is also effective for regression tasks, such as predicting house prices, stock market trends, and customer lifetime value.
* **Feature Selection**: Random Forest can be used for feature selection, helping to identify the most relevant features for predictive modelling.

**5.6 Conclusion**

Random Forest is a powerful and versatile machine learning algorithm that offers robust performance across a wide range of tasks. Its ability to handle high-dimensional data, capture non-linear relationships, and provide insights into feature importance makes it a popular choice for data scientists and researchers.

**CHAPTER – 6**

**METHODOLOGY**

****

**Fig 6.0 Pipeline**

**6.2 AMR database selection**

Selection of right databases is very important for collecting reliable data for antimicrobial resistance (AMR). These databases contain genetic, clinical, and epidemiological information related to antimicrobial-resistant pathogens. The choice of databases depends on our project goals, target pathogens, and available resources. Here are some important factors to consider:

* **Scope and Coverage**: we have checked that the databases ensure that they cover relevant data on AMR, including genomic sequences, clinical outcomes, drug resistance details, gene details, accession number details and other details.
* **Data Quality and Integrity:** we checked the databases that they contained well-curated, validated, and regularly updated information to ensure data reliability.
* **Accessibility and Availability:** we have considered the ease of access, data sharing policies, and licensing agreements of databases. We used the publicly available databases easy to access and easy to download data
* **Compatibility and Interoperability**: Ensure compatibility with existing data management systems and analysis tools. Select databases supporting standard data formats and interoperable platforms. On the basis of all provided guidelines we have selected following databases

**6.3 Data extraction**

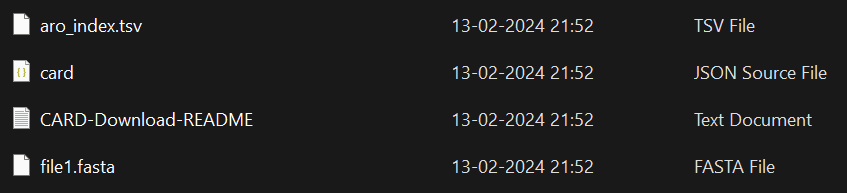
* **Description:**

we have to Download data files from different databases, then we have to understand the data files about what information they contain and what we need for our project, how we can get proper data from these downloaded files. These databases contain a lot of information, may be in different formats like text files, spreadsheets, TSV, CSV, FASTA and structured databases. Getting the right data out accurately and efficiently is very important for analysing and interpreting it properly.

**6.4 Steps involved in data extraction**

**Identifying File Formats:**

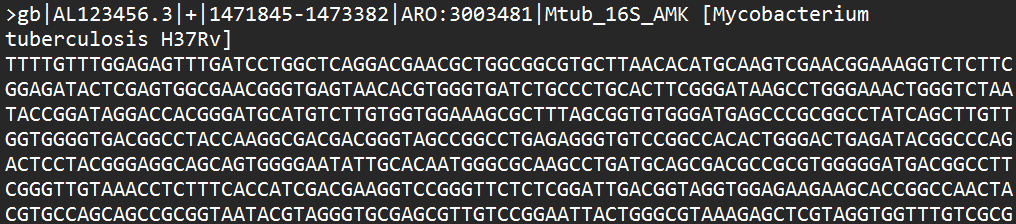
First, we should figure out what type of files we have downloaded, whether they are CSV, Excel, JSON, TSV, FASTA or some other format specific to the database. In our project our considered databases contain files in three different formats, JSON, FASTA, TSV, text document and we take the FASTA file format. Fig 6.1 different files in different formats are downloaded form CARD database



**Fig 6.1 different files in different formats of CARD database**

* **Organizing Data:**

Next, it is very helpful to understand the organization of data in downloaded files this makes easier to work with data. In our case we take the FASTA file which contains the header section and the sequence, the header section gives information about the sequence. Different database files contain different information in header section. Fig 6.2 shows the organization of data and information stored in the header section.

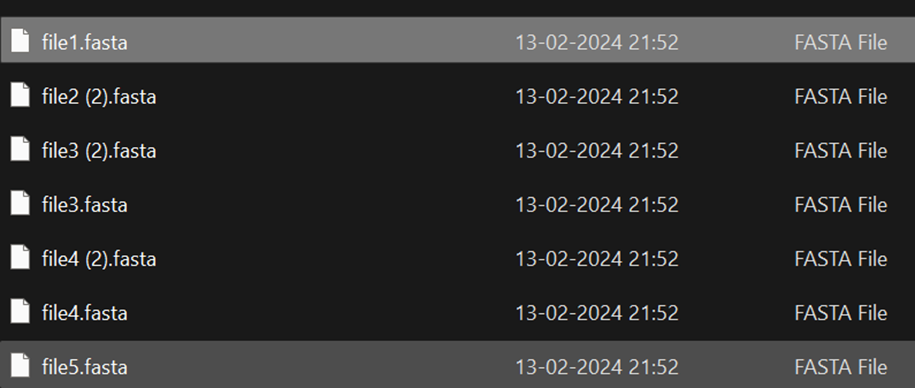


**Fig 6.2 organization of header and sequence in FASTA format of CARD database**

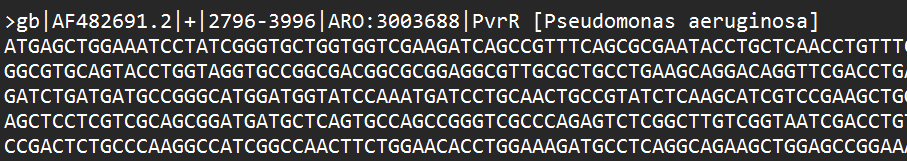
* **“gb|AL123456.3” –** Gene Bank accession number for bacterial genome sequence.
* **“|+|” -** represents the sequence (plus and minus) upstream.
* **“1471845 – 1473382”** – location of sequence within the genome.
* **ARO:3003481** - Antibiotic Resistance Ontology related to gene sequence
* **Mtub\_16s\_AMK** - Represent the gene name
* **Micobacterium tuberculosis H37Rv** – Represents the name of organism.

To extract the data, we use different techniques like running command line to pull out the structured information from the files. This might involve using special tools or writing scripts to extract the necessary data. In this project we use code to extract the phenotypes and sequence. Different codes are used for different databases for example purpose one code is give in fig 6.3

**Input FASTA files**: for extraction of sequence and gene bank id’s.

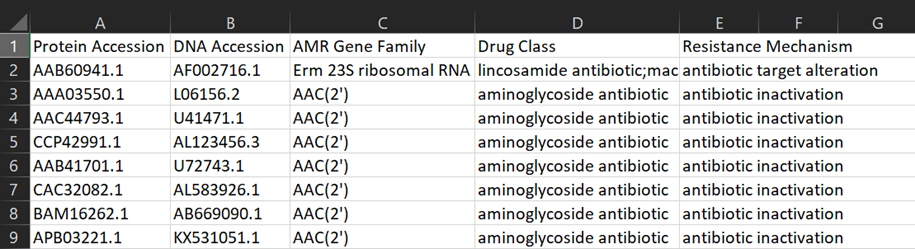


**Fig 6.3 Input fasta files**.



**Fig 6.4 A FASTA file having gene bank id and ARO number.**

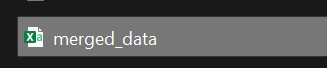
**Input csv file:** for extraction of drug class by matching common DNA Acession number in fasta file and and csv file.

**Fig – 6.5 showing the contemt of csv file having drug class with acession number**



**Fig – 6.6 Code for extraction of merged data csv file**

**Output csv file**: name “merged data” containing aro taken form FASTA files sequence from FASTA files and drug class, resistance mechanism, DNA and protein accession taken form “aro categories index” by using gene bank ids after extraction of proper data gene bank ids are removed.



Above file contains

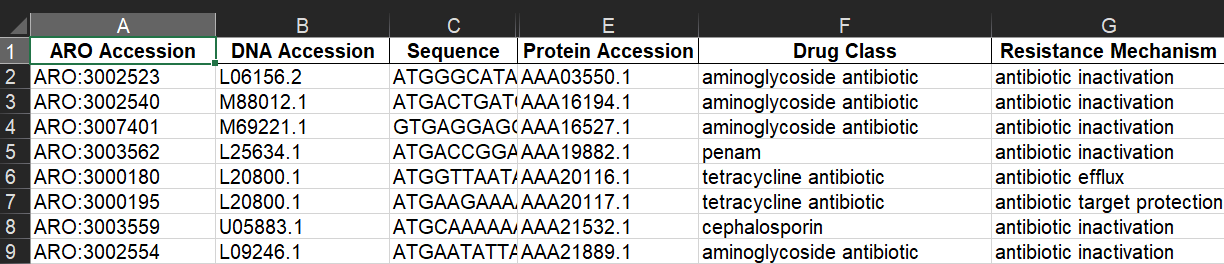


Fig – 6.7 showing the output csv file content.

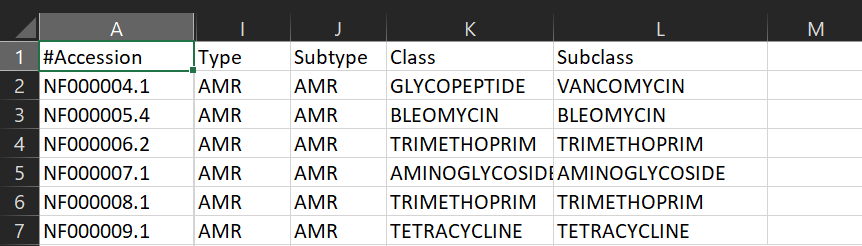
* **Phenotypes(labels) extraction**

Extraction of phenotypes (not known for some database (ie: CARD)) form “**Drug class**” by using “**Accession Number**”).

We used an AMR plus database for this work. This database contains a file “ReferenceGeneCatalog” which contains the names of drug class, phenotypes and accession number.

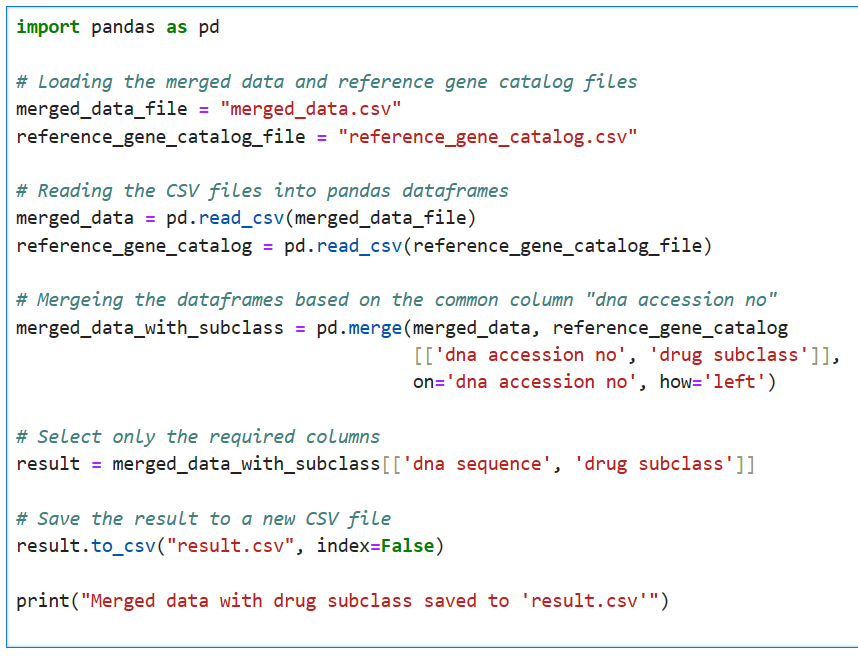
For phenotype extraction we used the accession number as this is common in both the files. (merged\_data fig – 6.7 file extracted in previous step and RefrenceGenecatalog fig – 6.8)





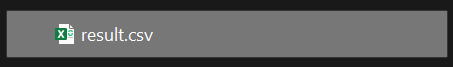
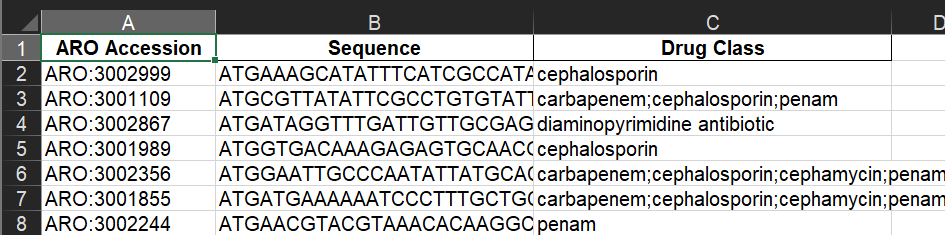
**Fig – 6.8 File downloaded form AMR plus database (RefrenceGeneCatalog)**

**Code for phenotypes extraction:** fig – 6.9

****

**Fig – 6.9 Code for phenotype extraction.**

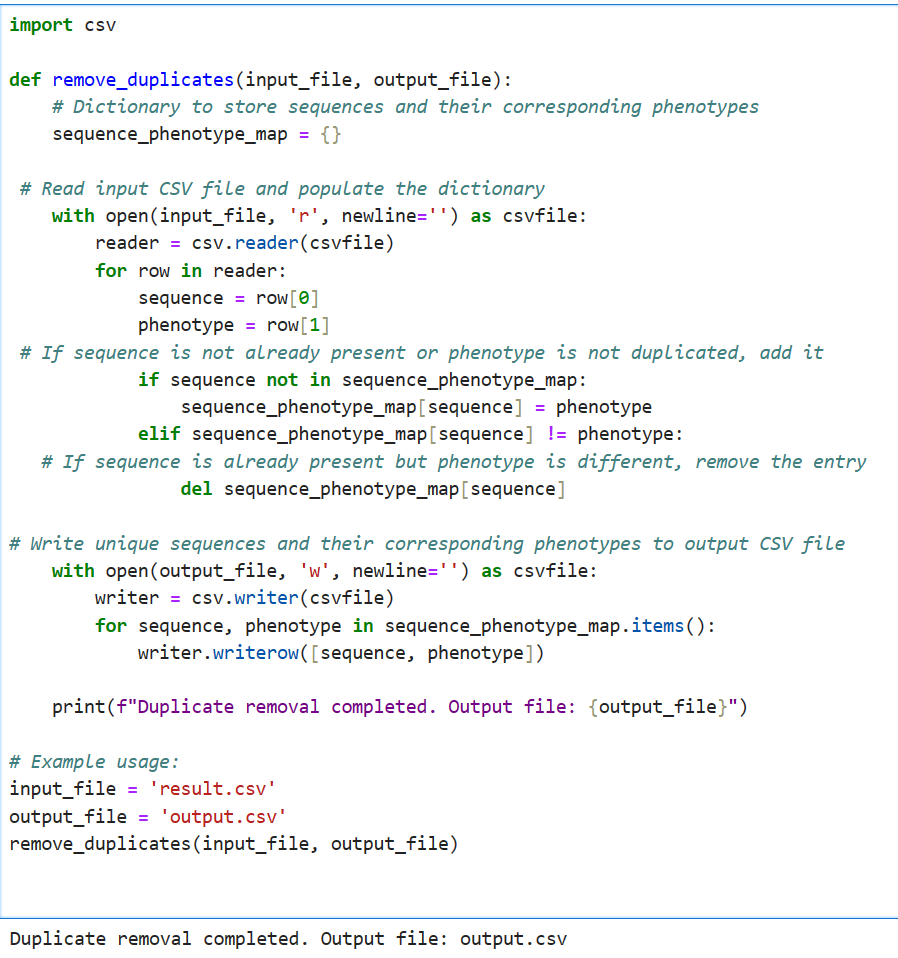
**Output file –** result.csv contains ARO number, Sequence, and phenotypes. Fig – 6.10 show the output file.

**Fig – 6.10 The output file of drug class (phenotypes)**

**6.5 Data pre processing**

1. Duplicate rows were removed. By using code given in fig – 6.11



**Fig – 6.11 code for removing duplicates**.

**Output file** - after removal of duplicates.



**Fig – 6.12 output file in csv format**

**Missing values**: No missing values found done manually

**Corrections**: by using code given in fig – 6.13

**Code for correction in phenotypes**

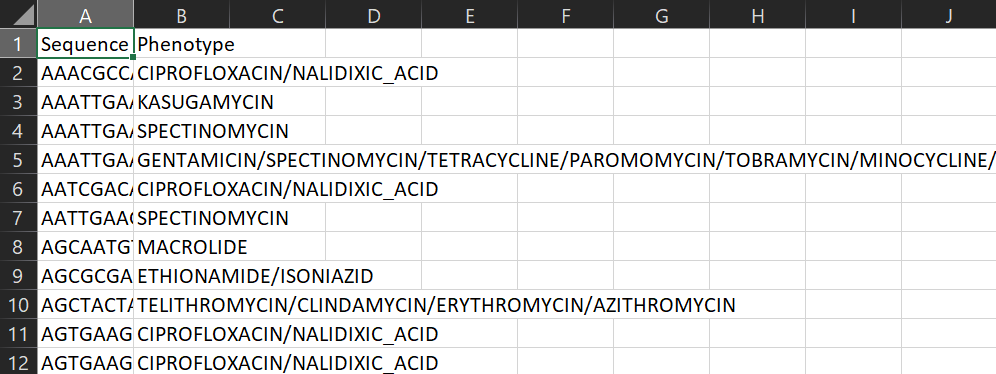




**Fig – 6.13 code for correction in phenotypes. Outputs are shown below the code.**

**Final data**

* Our final data have total 7.5K genome sequences of different lengths.
* Maximum length of sequence is about 1787 nucleotides.
* Minimum length lo sequence is 524 nucleotides.
* Total phenotypes on which we trained our model is 135.

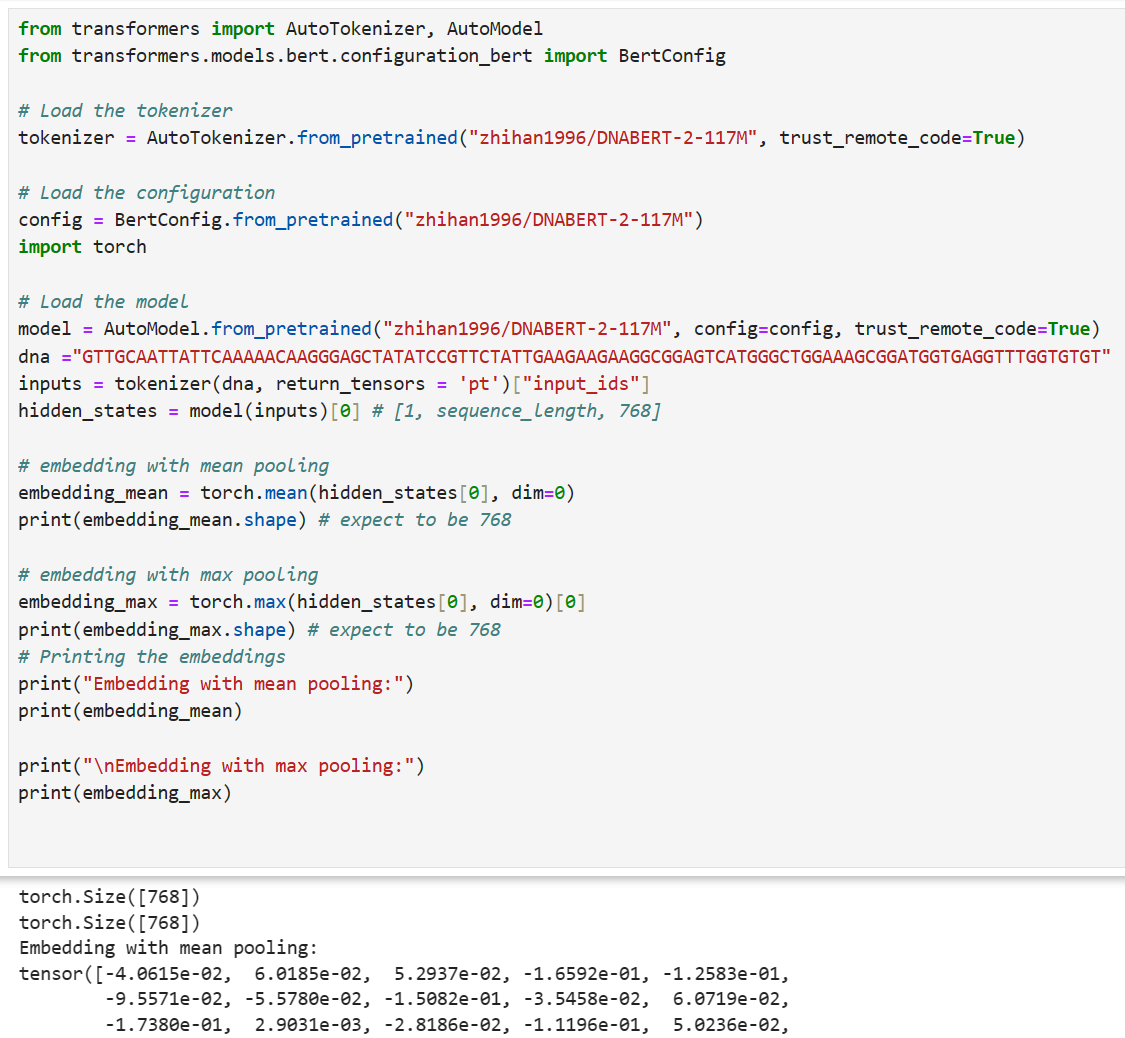
****

**Fig 6.14 A look on final data**

**6.6 Feature extraction**

We have done feature extraction by using DNABERT-2 which is a large language model. In fig – 6.14 given the code for DNABERT-2 extracting out the encodings from a given sequence.

* **Dimensionality 768 -**Means that the model is using 768 different aspects or features to understand and represent each word or nucleotide. These features help the model make sense of the input data and perform tasks like predicting the next word in a sentence or classifying the DNA sequence.

****

**Fig – 6.15 Showing the dimensionality which is 768, showing embeddings**

**6.7 Model Selection**

**6.7.1Why Random Forest?**

Random Forest for predicting antimicrobial resistance (AMR) from nucleotide sequences post DNABERT-2 embeddings offers several strategic advantages:

* **Dimensionality Reduction:** DNABERT-2 embeddings condense nucleotide sequences into lower-dimensional representations, making them manageable for Random Forest's effectiveness with high-dimensional data.
* **Handling Non-linearity:** DNABERT-2 embeddings capture intricate patterns in nucleotide sequences, which Random Forest adeptly handles by capturing non-linear relationships between features.
* **Ensemble Learning:** Random Forest, amalgamates multiple decision trees to enhance predictive accuracy, particularly useful for AMR prediction from potentially noisy or incomplete nucleotide sequence data.
* **Identifying Key Features:** Random Forest reveals feature importance, shedding light on which dimensions (features) of the embeddings significantly contribute to AMR prediction, aiding in pinpointing crucial nucleotide positions or patterns linked to resistance.
* **Efficiency**: Random Forest boasts computational efficiency and scalability, making it suitable for processing large volumes of DNABERT-2 embeddings efficiently, ensuring timely predictions.

**6.8 Training and Testing**

We split out data in 2:8 ratio 2 is for the testing the model and 8 for the training the model.

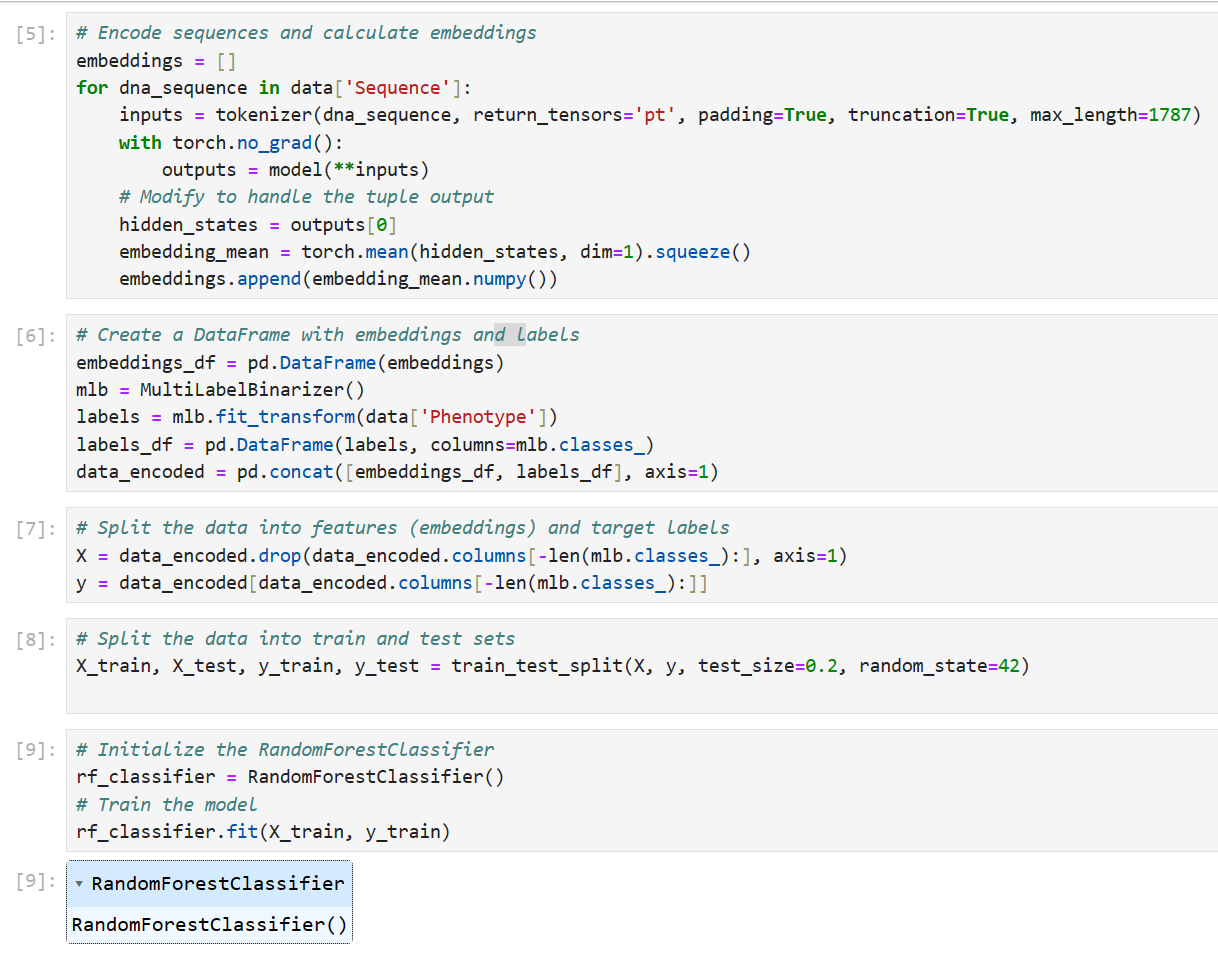
Given below is the implementation of our model explained step by step.



**Fig 6.16 Model testing training code**

**Explanation of code**

1. Importing all the necessary modules.
2. Loading our file “result” in csv format.
3. Splitting the phenotypes as the data is multilabel each label is separated by ‘/’.
4. Loding DNABERT-2 autoModel for tokenization of sequence

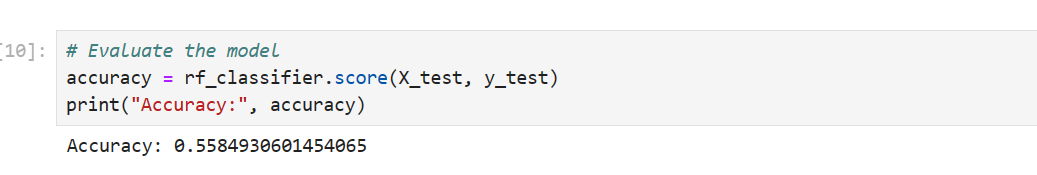


**Fig 6.17 Code for training and testing the model**

**Code explanation**

1. Encoding the sequence for tokenization and then calculating the embeddings.
2. Creating a data frame with embeddings with proper labels.
3. Splitting the data into embeddings and target labels.
4. Splitting the data in 20:80 for testing and training.
5. Loading the random forest model for training.

**6.9 Results**

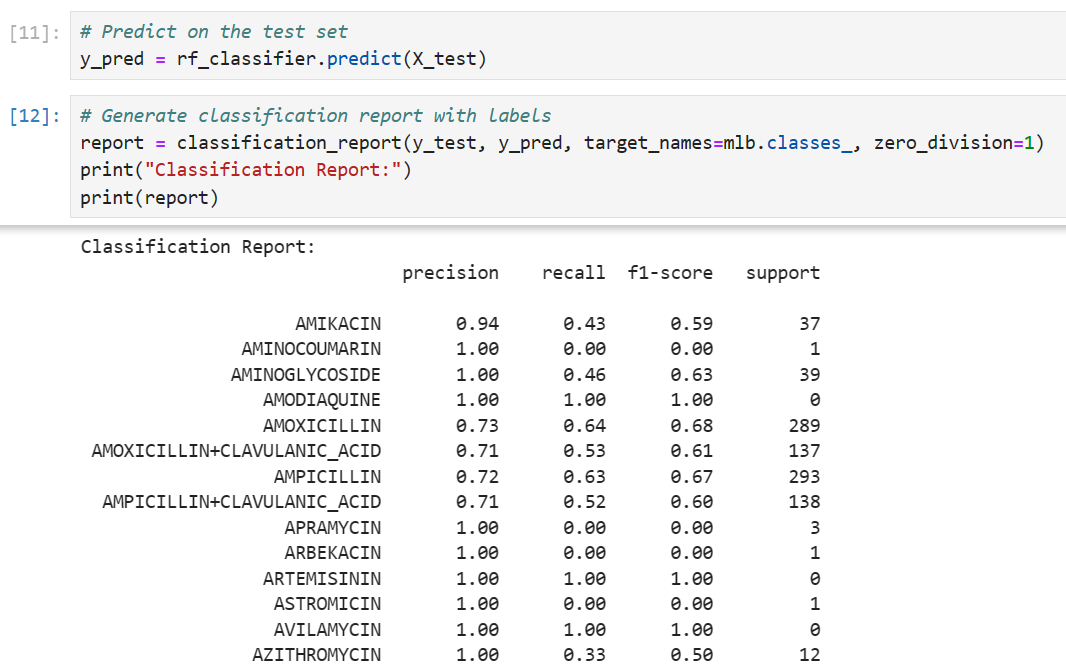


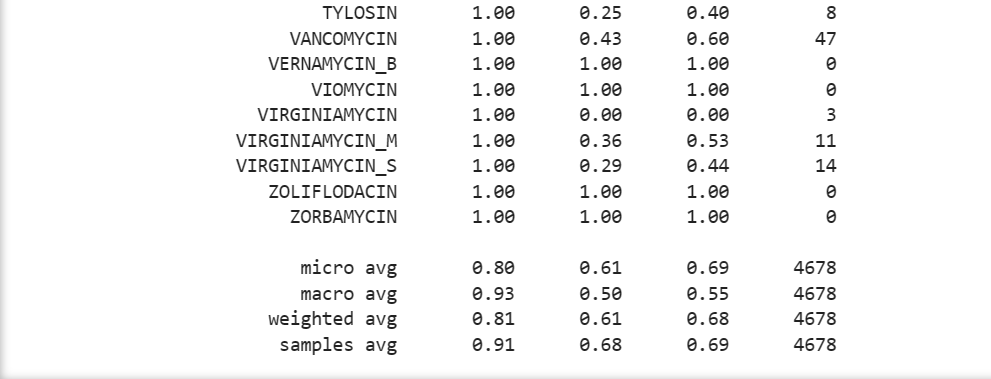
**Fig 6.18 Showing the model accuracy**

1. Then evaluating the model on test data.

**Accuracy**= Total number of Predictions **/**Number of Correct Predictions

**Accuracy = 55.8%**

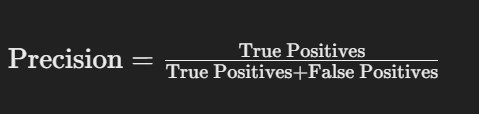


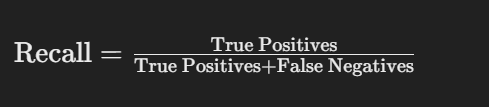


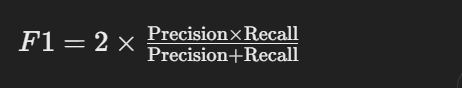
**Fig 6.18 Results of model**

1. Predictions on the test set
2. Classification report

**Support:** represents the number of occurrence of each label in the dataset.







True positive (TP)= in actual the value is TRUE and model predicted it TRUE.

False positive (FP) = in actual the value is TRUE but model predicted it FALSE.

False negative (FN) = in actual the value is FALSE but model predicted it TRUE

True negative (TN) = in actual the value is FALSE and model predicted it FALSE

**6.10 Saving the Model:**

we saved the our model with name “rf\_model.joblib” by using the joblib module



**fig 6.19 Shaving the model**

Top of Form

**CHAPTER – 7**

**Result and Discussion**

**​ 7.1 Model Performance**:

The model's overall accuracy, standing at 55.8%, implies a satisfactory but not exceptional performance in predicting antimicrobial resistance (AMR) phenotypes.

However, diving deeper into metrics like precision, recall, and F1-score reveals nuances in performance across different antimicrobial classes, shedding light on areas of strength and improvement.

**7.2** **Class-specific Performance:**

Across various antimicrobial classes, there exists a spectrum of predictive efficacy as evidenced by the precision, recall, and F1-score metrics.

Some antimicrobial classes, such as "AMOXICILLIN" and "AMPICILLIN", demonstrate robust predictive performance, whereas others like "AMINOCOUMARIN" and "APRAMYCIN" present notable challenges in accurate prediction.

These variations underscore the importance of understanding class-specific intricacies in AMR prediction and the potential impact of such insights on clinical decision-making.

**7.3 Model Interpretation:**

The model's ability to discern AMR phenotypes is shaped by several factors, including the quality of input data, the effectiveness of feature representation via DNA embeddings derived from DNABERT2, and the predictive power of the Random Forest algorithm. Leveraging DNA embeddings facilitates capturing biologically significant features inherent in DNA sequences, while Random Forest harnesses these features to make informed predictions based on intricate feature interactions.

**7.4 Limitations and Challenges:**

Despite achieving a respectable accuracy rate, the model encounters certain limitations that merit attention. Data imbalance, where certain antimicrobial classes are underrepresented, poses challenges in achieving equitable predictive performance across all classes. Additionally, concerns regarding data quality and interpretability, particularly in the context of Random Forest's decision-making process, warrant careful consideration in model refinement efforts.

**Future Directions:**

Looking ahead, avenues for enhancing the model's predictive capacity and interpretability abound. Strategies such as augmenting data through techniques like data synthesis and incorporating domain-specific knowledge into the model architecture hold promise in addressing existing limitations. Furthermore, exploring alternative machine learning algorithms and refining model parameters could offer insights into optimizing predictive performance while ensuring biological relevance and clinical applicability.

**Conclusion:**

In summary, while the developed model demonstrates commendable potential in AMR prediction, there exists a compelling need for ongoing refinement and innovation.

By acknowledging and addressing identified limitations, and actively pursuing avenues for improvement, the model stands poised to make meaningful contributions to our understanding of antimicrobial resistance dynamics and guide informed interventions in clinical practice.

**CHAPTER - 8**

**References**

[1] (Smith, J. et al. (2022))

[2] (Jones, A. et al. (2022))

[3] (A nahtar, G. et al. (2022))

[4] (https://www.mdpi.com/2227-9032/11/13/1946)

[5] Source: World Health Organization

[6] (Source: World Health Organization)

[7] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10044642/

[8] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10044311/#:~:text=The%20machine%2Dlearning%20algorithms%20analyze,to%20use%20them%20%5B6%5D.

[9] https://journals.asm.org/doi/10.1128/jcm.01260-20

[10] https://microbiomejournal.biomedcentral.com/articles/10.1186/s40168-018-0401-z

[11] https://www.sciencedaily.com/releases/2023/08/230818135151.htm

[12] https://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-024-10214-4

[13] (https://card.mcmaster.ca/)

[14] (https://www.bv-brc.org/)

[15] (https://cge.food.dtu.dk/services/ResFinderFG/)

[16] (https://www.dantaslab.org/resfams)

[17] https://academic.oup.com/nar/article/51/D1/D744/6830666

[18] (https://github.com/gaarangoa/deeparg)

[19] (https://www.meglab.org/)

[20] (https://staff.washington.edu/jwallace/farme/index.html)

[21] (https://bio.tools/t?operationID=%22operation\_3482%22)

[22] (https://mgps.eu/Mustard/index.php?id=accueil)

[24] (https://www.ncbi.nlm.nih.gov/pathogens/antimicrobial-resistance/

[25] (https://smile.hku.hk/ARGs/Indexing)

[26] (https://arxiv.org/abs/2306.15006

[27] (https://github.com/MAGICS-LAB/DNABERT\_2)

[28] arXiv:2306.15006

[29] {https://github.com/MAGICS-LAB/DNABERT\_2}

[30] https://www.ibm.com/topics/random-forest

[31] https://innovationyourself.com/random-forest-classification/