Predictive Modeling of Immune Responses to Pertussis Vaccination

Advisors:

Barry Grant (Professor)

Jason Hsiao (Biological Sciences PhD Student)

Team Members:

Peng Cheng

Javier Garcia

Weikang Guan

Brian Qian

Table of Contents

1 Abstract

Our capstone project focuses on Pertussis, commonly known as Whooping cough, a highly contagious respiratory infection. We explore the challenges and nuances of the two main vaccines: whole-cellular (wP) and acellular (aP). Our research highlights the balance between the safety and effectiveness of vaccines, emphasizing the necessity of ongoing monitoring and research to evaluate how vaccine-induced immunity fluctuates over time in individuals, ensuring sustained effectiveness and safety in public health. The primary goal of the project was to create predictive models for forecasting immune response outcomes following pertussis vaccination, specifically targeting IgG antibody titer levels 14 days post-vaccination, monocyte frequencies one day post-vaccination, and gene expression levels of genes like CCL3 three days post-vaccination. The team utilized a comprehensive dataset of over 500 blood samples from 118 participants, including detailed demographic and immunological profiles. Through rigorous data preprocessing, including handling missing values, detecting outliers, and feature selection, the data was prepared for model building. A variety of models, from simple linear regressors to advanced ensemble learners like Random Forest and Gradient Boosting, were trained and evaluated using cross-validation. The models' performance was assessed using metrics such as R-squared, Mean Absolute Error (MAE), and Mean Squared Error (MSE), with ensemble methods demonstrating superior predictive accuracy. The findings revealed that targeted feature selection and advanced modeling techniques significantly enhanced the predictive power and reliability of the models in understanding and forecasting immune responses to vaccinations.

2 Introduction and Question Formulation

2.1 Challenges

The challenge that led to the problem in our project was rooted in the complexities of understanding and predicting immune responses to pertussis vaccination. Pertussis, or Whooping cough, is a significant public health concern due to its highly contagious nature and potential severity, particularly in infants and young children. Despite the availability of two main types of vaccines—whole-cellular (wP) and acellular (aP)—each presents its own set of challenges that complicate efforts to ensure long-term immunity and safety. Individuals exhibit varied immune responses to the pertussis vaccine, influenced by factors such as age, sex, ethnicity, and pre-existing health conditions, making it difficult to predict who will mount a robust immune response. The whole-cellular vaccine (wP) is known for its strong and durable immune response but has a higher incidence of side effects, whereas the acellular vaccine (aP) is safer but associated with waning immunity over time. This trade-off between safety and long-term effectiveness necessitates a nuanced approach to vaccine administration and monitoring.

Moreover, the immune response to vaccination involves a complex interplay of various immunological markers, including antibody levels, cell frequencies, and gene expression profiles. Capturing and modeling these dynamics requires sophisticated analytical tools and comprehensive datasets. The push towards personalized medicine, where treatments and interventions are tailored to individual characteristics, further underscores the importance of developing predictive models that can accurately forecast immune responses based on a person's unique demographic and immunological profile. This approach aims to enhance the efficacy and safety of vaccination programs. Additionally, continuous monitoring and research are essential to evaluate how vaccine-induced immunity changes over time, identifying potential gaps in immunity and the need for booster doses to ensure sustained public health protection.

These multifaceted challenges highlight the need for robust predictive models. Our project sought to address these issues by leveraging a comprehensive dataset of over 500 blood samples from 118 participants, incorporating detailed demographic and immunological data. Through advanced data preprocessing techniques and sophisticated modeling approaches, we aimed to unravel the complexities of the immune response to pertussis vaccination and provide actionable insights to improve vaccine strategies.

2.2 Data Science Problem

We combined several critical ingredients, which collectively enabled us to construct robust predictive models for understanding immune responses to pertussis vaccination. We started with a comprehensive dataset that included detailed measurements of immunological markers such as antibody titers, cell frequencies, and gene expression levels. This dataset also contained demographic information like age, sex, ethnicity, race, and birthdate, providing context and helping us understand how different groups respond to the vaccine. Temporal data on the timing of vaccination and subsequent immune responses (e.g., day 0, day 1, day 3, and day 14 post-vaccination) was crucial for capturing the dynamics of the immune response over time.

Data preprocessing techniques were essential, including methods for handling missing values, detecting and treating outliers, feature scaling to standardize the range of data features, and feature selection to identify the most relevant predictors among a vast array of variables. These steps ensured the completeness, reliability, and uniformity of our data, enhancing model performance and interpretability.

Advanced modeling approaches were employed, ranging from simple linear regression to complex ensemble methods like Random Forest and Gradient Boosting. Cross-validation techniques were used to assess model performance and ensure generalizability, while tools like GridSearchCV optimized model parameters, balancing complexity and predictive power. Evaluation metrics such as R-squared (R^2), Mean Absolute Error (MAE), and Mean Squared Error (MSE) were used to measure the accuracy and explanatory power of the models.

Biological and domain knowledge was integral to our approach. Understanding the differences between whole-cellular (wP) and acellular (aP) vaccines, including their efficacy and safety profiles, was essential. Collaborating with immunologists ensured that our features and models were biologically meaningful and relevant.

Iterative development and validation processes involved continuously refining models based on performance metrics and feedback, ensuring robustness and accuracy as new data became available. Collaboration with domain experts was crucial for validating findings and improving the practical applicability of the models.

Visualization and interpretation were key components. We created visual tools to illustrate key findings, model performance, and relationships between variables, aiding in the communication of results to stakeholders. Ensuring that models were interpretable provided insights into the underlying biological processes, making the findings actionable and relevant for improving vaccine strategies and public health outcomes. By integrating these ingredients, we formulated a comprehensive data science problem that addressed the complexities of predicting immune responses to pertussis vaccination.

2.3 Questions

Some of the questions that have been formulated to address our challenges are:

- How can we accurately predict IgG antibody titer levels 14 days post-pertussis vaccination using immunological data?
- What are the most significant immunological predictors of monocyte frequencies one day after vaccination?
- How do gene expression levels of specific genes, such as CCL3, change three days post-vaccination, and what factors influence these changes?
- What is the comparative effectiveness of whole-cellular (wP) versus acellular (aP) vaccines in inducing a sustained immune response over time?
- What are the key indicators of waning immunity, and how can ongoing monitoring be used to predict the need for booster doses?
- How can advanced machine learning models be employed to improve the predictive accuracy and interpretability of immune response data?

- What are the trade-offs between model complexity and performance, and how can we optimize model parameters to balance these?
- How can visualization tools be developed to effectively communicate complex immunological data and model predictions to stakeholders?
- What are the best practices for integrating biological domain knowledge with data science techniques to enhance model relevance and applicability?

2.4 Related Work

- Longitudinal analysis shows durable and broad immune memory after SARS-CoV-2 infection with persisting antibody responses and memory B and T cells
  https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8095229/pdf/nihpp-2021.04.19.21255739v2.pdf
- Modeling of malaria vaccine effectiveness on disease burden and drug resistance in 42 African countries
  https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10576074/pdf/43856_2023_Article_373.pdf
- mSphere of Influence: Predicting Immune Responses and Susceptibility to Influenza Virus
  https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7082138/pdf/mSphere.00085-20.pdf
- Exploring the optimal vaccination strategy against hepatitis B virus in childhood
  https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10293880/pdf/br-19-01-01631.pdf
- Gene expression profiling in vaccine therapy and immunotherapy for cancer
  https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3411321/pdf/nihms394797.pdf
- A machine learning model identifies patients in need of autoimmune disease testing using electronic health records
  https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10130143/pdf/41467_2023_Article_37996.pdf
- Comparative Effectiveness of COVID-19 Vaccines in Preventing Infections and Disease Progression from SARS-CoV-2 Omicron BA.5 and BA.2, Portugal
  https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9973705/pdf/22-1367.pdf

3 Team Roles and Responsibilities

Barry Grant - Advisor (Professor)

- Scientific Guidance: Provides high-level guidance on the project's direction and scientific validity. Offers expertise in the biological aspects of the immune response and ensures that the project aligns with broader research goals.
- Research Oversight: Oversees the research methodology and ensures that the scientific approaches used are robust and appropriate for the study.

Jason Hsiao - Advisor (Biological Sciences PhD Student)

- Immunology Expertise: Offers specialized knowledge in biological sciences, particularly in immunology. Assists in the interpretation of data and results from a biological perspective and ensures that the modeling approaches are scientifically sound.
- Technical Support: Provides technical support and advice on the experimental aspects of the study, including the handling and analysis of biological data.

Peng Cheng - Solution Architect & Budget Controller

- Solution Architect: Oversees the project's architecture, ensuring that all components are integrated efficiently. Designs the data pipeline and modeling framework.
- Budget Controller: Manages the project budget, ensuring that resources are allocated appropriately and that the project stays within financial constraints.

Javier Garcia - Business Manager & Data Engineer (MLops)

- Business Manager: Handles the operational management of the project, including timelines, deliverables, and coordination with stakeholders.
- Data Engineer (MLops): Develops and maintains the data pipeline, ensuring data is processed efficiently and available for modeling. Manages the deployment of models in cloud infrastructure and oversees continuous integration and deployment (CI/CD) processes.

Weikang Guan - Data Analyst & Visualization Developer

- Data Analyst: Conducts detailed data analysis to uncover patterns and insights from the dataset. Performs preprocessing tasks such as cleaning, transforming, and feature engineering.
- Visualization Developer: Creates visual representations of the data and modeling results. Develops dashboards and visual tools to communicate findings effectively to stakeholders and team members.

Brian Qian - Data & Dashboard Developer

- Data Developer: Develops the data infrastructure needed for the project, including database management and data retrieval systems. Ensures that data is accessible and well-organized for analysis.

- Dashboard Developer: Designs and implements interactive dashboards that present key metrics and model outcomes. Ensures that the dashboards are user-friendly and provide meaningful insights to researchers and stakeholders.

4 Data Acquisition

4.1 Data Sources

The data sources for our project include a comprehensive collection of immunological, demographic, and temporal data. Immunological data encompasses thousands of data points for each participant, capturing various markers such as antibody titers, cell frequencies, and gene expression levels. This data is collected at specific time points post-vaccination (day 0, day 1, day 3, and day 14), ensuring a detailed temporal analysis of the immune response. Demographic data, collected once for each of the 118 participants, includes birthdate, sex, ethnicity, and race, providing context for understanding how different groups respond to the vaccine. Vaccine Responds Timeline is shown as figure 4-1.

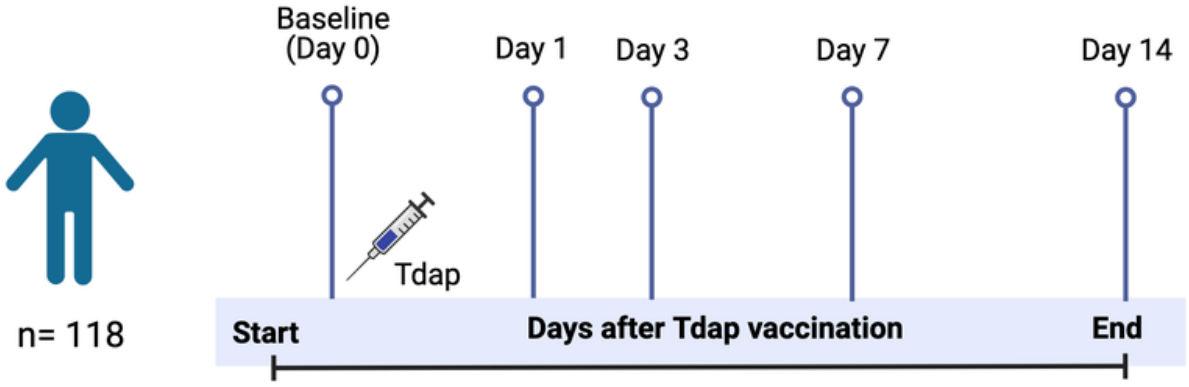

Figure 4-1 Vaccine Responds Timeline

We also utilize temporal data related to the dates and times of vaccination and subsequent immune response measurements, ensuring a structured timeline for our analysis. The PBMC cell frequency data provides detailed information on various cell populations per sample, while plasma gene expression data includes transcript counts and TPMs for numerous genes, offering a deep dive into gene activity post-vaccination. Plasma antibody titer data details antibody levels, including antigen specificity and Mean Fluorescence Intensity (MFI) units, while plasma cytokine concentration data quantifies protein expressions, particularly cytokine levels, which are crucial for understanding immune signaling mechanisms.

Additionally, metadata/ontology tables provide a comprehensive cross-reference for gene, transcript, and protein IDs, facilitating data interoperability and validation. Overall, our dataset is large and varied, with data collected at specific intervals to capture the dynamic immune response, allowing for robust analysis and the development of predictive models that yield meaningful insights into vaccine efficacy and safety. Dataset Overview is shown as figure 4-2.

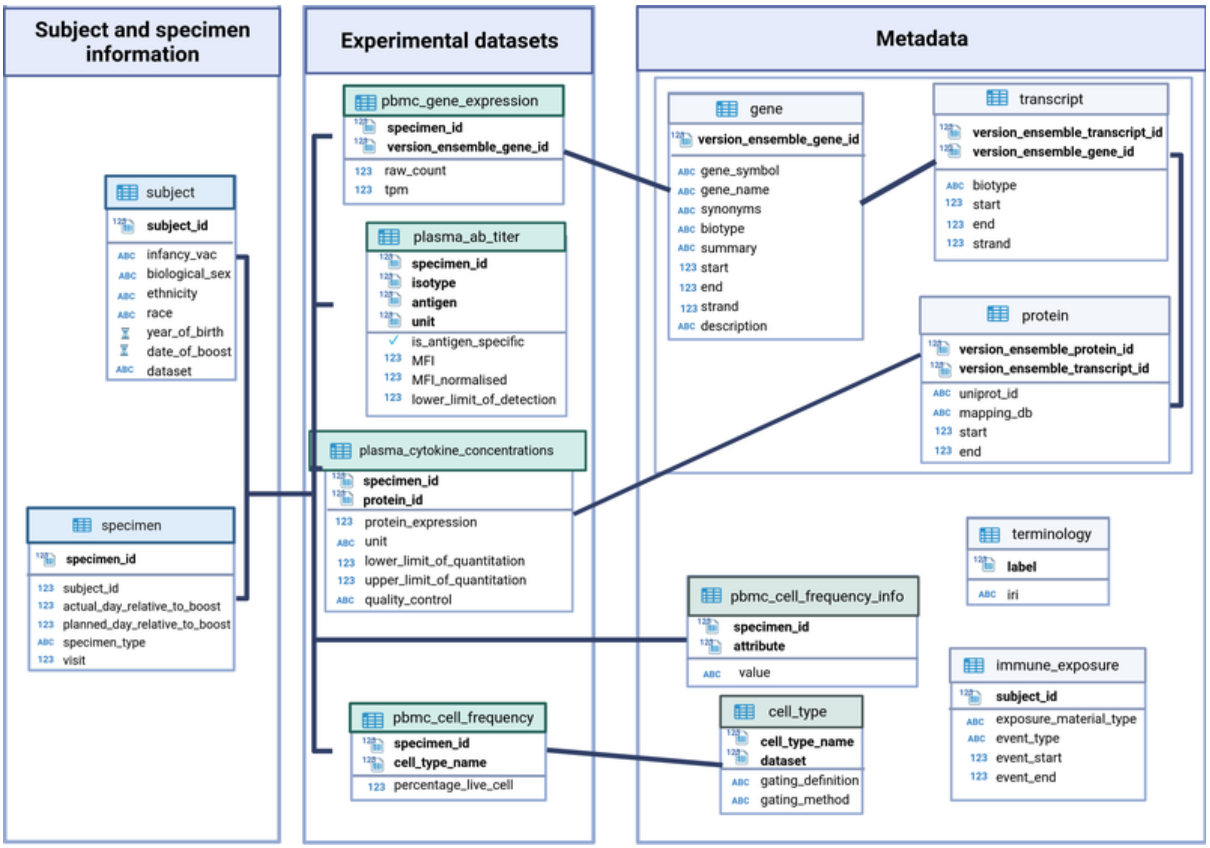

Figure 4-2 Dataset Overview

4.2 Data Collection

4.2.1 Data Sizes

To access and retrieve the data required for our project, we use two primary methods provided by the CMI-PB laboratory: APIs and SFTP. The first method involves using the Swagger UI and PostgREST API. The Swagger UI provides a user-friendly interface for exploring and interacting with the CMI-PB API, which is accessible at https://www.cmi-pb.org/api/v4_1. The PostgREST API dynamically generates endpoints for various data tables and supports operations such as GET, POST, DELETE, and PATCH. This allows us to retrieve information from tables like cell_type, gene, pbmc_cell_frequency, pbmc_gene_expression, plasma_ab_titer, plasma_cytokine_concentration, specimen, and subject. Each of these tables provides specific data such as cell population names, gene mapping information, cell frequencies, gene expression levels, antibody titers, cytokine concentrations, clinical sample details, and demographic information.

The second method involves using SFTP provided by the CMI-PB consortium for large-scale data downloads and bulk access. This method is particularly useful for downloading text files containing the data for the second CMI-PB challenge. The SFTP access allows us to download comprehensive datasets including subject information for 118 subjects, specimen information for 939 samples, antibody titer data for 112 subjects and 829 samples, gene expression data for 93 subjects and 507 samples, plasma cytokine data for 75 subjects and 406 samples, and cell frequency data

for 74 subjects and 414 samples. This method ensures we can efficiently handle and analyze large volumes of data. Data Sizes is shown as figure 4-3.

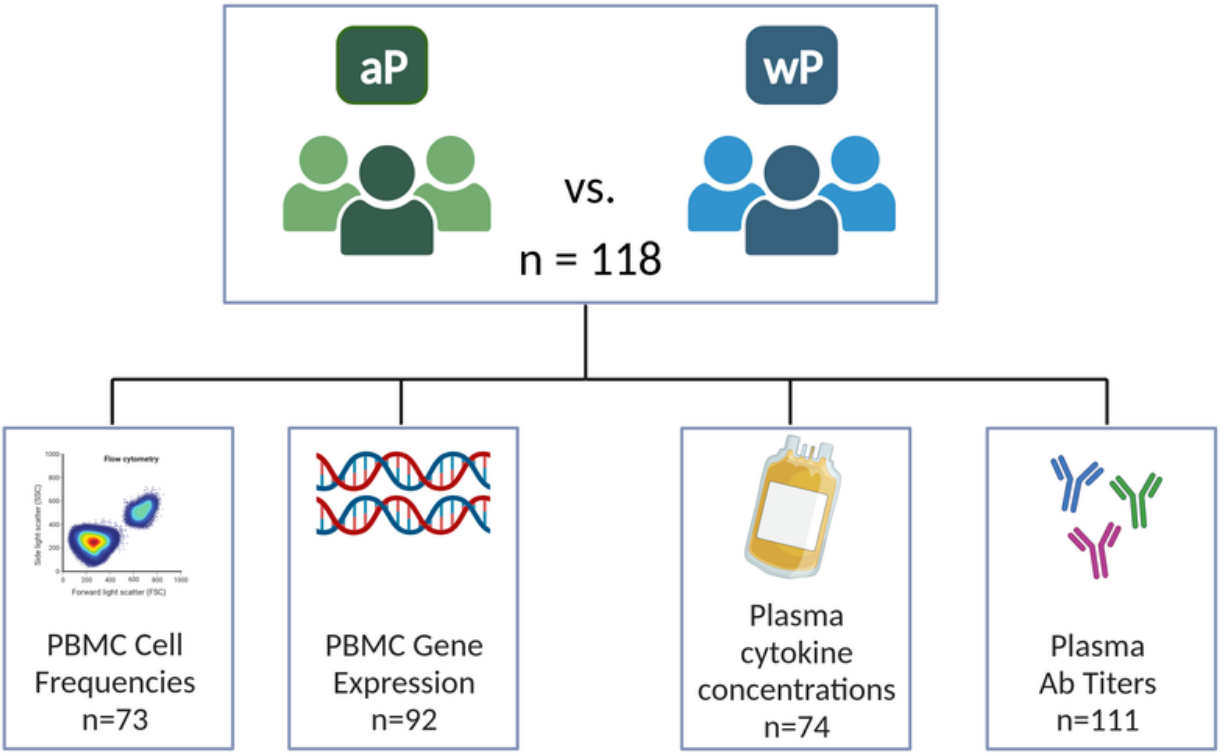

Figure 4-3 Data Sizes

By leveraging these technologies and methods, we can access a comprehensive and high-quality dataset necessary for our predictive modeling and analysis. This approach ensures robust and reliable insights into the immune response to pertussis vaccination, enabling us to address the key questions identified in our project.

4.2.2 Data Pipelines

Data pipeline from ingestion to analysis to sharing your results including the process and infrastructure. To address the need for real-time access to dynamic immune response data and ensure periodic updates for analysis, our approach incorporates the use of APIs for seamless data access. PosgREST API Description is shown as table 4-1.

Table 4-1 PosgREST API Description

Table Description	Subjects	Samples	API endpoint
Subject information	118		subject
Specimen information		939	specimen
Antibody titer data	112	829	plasma_ab_titer
Gene expression data	93	507	pbmc_gene_ expression
Plasma cytokine data	75	406	plasma cytokine concentration
Cell frequency data	74	414	pbmc_cell_frequency
cell type gating information	52		cell_type

This method is complemented by regular refresh cycles, enabling continuous analysis and ensuring that our datasets remain current and reflective of the latest immune response patterns. Data Pipeline Workflow is shown as figure 4-4.

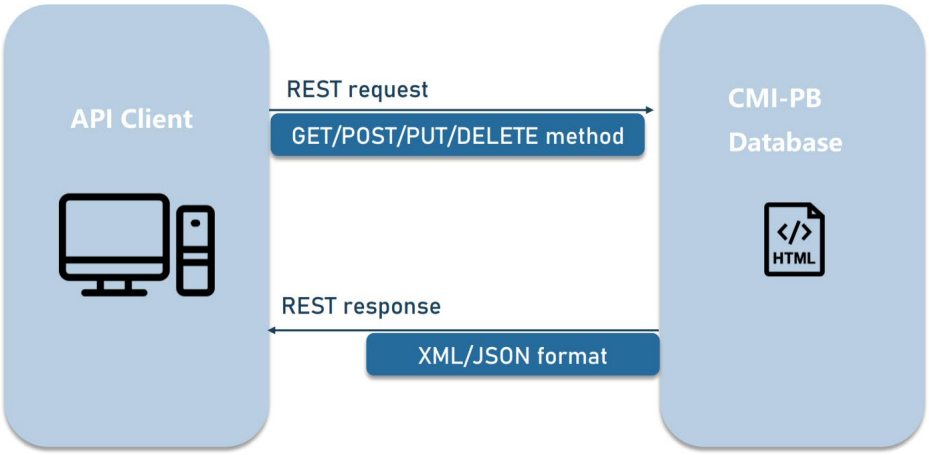

Figure 4-4 Data Pipeline Workflow

4.2.3 Data Environment Set Up

Our data environment is designed to efficiently handle the diverse and extensive datasets required for our predictive modeling project on immune responses to pertussis vaccination. We employ a hybrid setup that leverages both cloud and local resources, combining the strengths of databases and flat files for optimal performance and flexibility.

In the cloud environment, we use Amazon Web Services

Our data environment is designed to efficiently handle the diverse and extensive datasets required for our predictive modeling project on immune responses to pertussis vaccination. We employ a hybrid setup that leverages both cloud and local resources, combining the strengths of databases and flat files for optimal performance and flexibility.

In the cloud environment, we use Amazon Web Services (AWS) for scalable storage and compute resources. AWS provides the flexibility to handle large datasets and the computational power required for complex modeling tasks. For data storage, we utilize Amazon S3, which offers scalable and durable storage for raw data, intermediate files, and

processed datasets. For data processing, AWS EC2 instances are used to run data processing and machine learning workloads, allowing us to adjust computational power based on the demands of our tasks. Additionally, AWS SageMaker is employed for building, training, and deploying machine learning models, providing a managed environment that simplifies development and scaling.

The cloud environment offers several advantages, including scalability to easily adjust resources based on project needs, flexibility to access a wide range of services and tools, and enhanced collaboration by allowing team members to access the environment from any location.

Locally, we use servers for storing smaller, processed datasets and for initial data exploration. This setup helps reduce latency and speeds up the development process. Local machines are used for preliminary data analysis, visualization, and model development, providing a convenient and immediate environment for testing and debugging. The local environment offers advantages such as low latency, immediate access to data without internet connectivity, and cost efficiency by utilizing existing infrastructure for initial stages of data processing and model development.

We manage structured data using PostgreSQL, which provides robust querying capabilities and ensures data integrity. PostgreSQL is ideal for handling structured data such as demographic information, cell frequency data, and antibody titer data, facilitating easy integration and complex querying essential for linking different datasets and extracting insights.

For unstructured data, we use flat files (CSV/TSV) to store large-scale raw gene expression data and intermediate processing files. Flat files are also used for data transfer between different systems and for initial data exploration using tools like Python and R. This approach combines the flexibility and simplicity of flat files with the robust data management capabilities of databases.

By leveraging both cloud and local resources, integrating databases and flat files, our comprehensive setup ensures efficient processing, analysis, and modeling of the data necessary to understand and predict immune responses to pertussis vaccination. This hybrid environment provides the scalability, flexibility, and efficiency needed to handle the diverse requirements of our project.

5 Data Preparation

5.1 Datasets Quality Issues

The datasets we utilized for our predictive modeling project on immune responses to pertussis vaccination had several quality issues that needed to be addressed to ensure the accuracy and reliability of our analysis.

Firstly, there were missing values across various data points, including demographic information, immunological markers, and gene expression levels. Missing data can lead to biased results and reduce the overall effectiveness of our predictive models. Addressing these gaps required implementing strategies such as imputation, where we estimated the missing values based on available data, or excluding incomplete records where appropriate.

Secondly, outliers were present in the dataset. These anomalous data points could skew the results and affect the performance of our models. Identifying and handling outliers involved statistical methods to detect values that deviated significantly from the norm and deciding whether to correct or exclude them to maintain data integrity.

Thirdly, there was a need for feature scaling. The dataset included variables with different scales and units, which could disproportionately influence the model outcomes. Normalizing or standardizing the data was necessary to ensure that all variables contributed equally to the predictive models.

Additionally, the data contained inconsistencies in labeling and formatting. For instance, demographic categories such as ethnicity and race were not consistently coded, requiring standardization to ensure uniformity across the dataset. Similarly, different measurement units for immunological markers needed to be harmonized to avoid discrepancies.

Furthermore, there were challenges related to the temporal aspect of the data. Some time points for sample collection were not consistently recorded, making it difficult to analyze the immune response dynamics accurately. Ensuring accurate and consistent timestamping was essential for the temporal analysis.

Lastly, the dataset's size and complexity presented challenges in handling and processing the data efficiently. Large-scale gene expression data, in particular, required robust computational resources and efficient data processing techniques to manage and analyze effectively.

Addressing these quality issues involved a comprehensive data preprocessing workflow, including data cleaning, normalization, imputation, outlier detection, and standardization. This ensured that the datasets were accurate, consistent, and ready for reliable predictive modeling and analysis.

5.2 transform and integration

To transform and integrate raw data into a format suitable for analysis in our project on predicting immune responses to pertussis vaccination, we implemented a comprehensive data preprocessing workflow. This involved several key steps to ensure the data was clean, consistent, and suitable for robust analysis and modeling.

By implementing these data cleaning, transformation, and integration steps, we successfully prepared the raw data for robust predictive modeling and analysis. This comprehensive preprocessing workflow ensured that our datasets were accurate, consistent, and suitable for generating reliable insights into immune responses to pertussis vaccination.

5.2.1 Data Cleaning

5.2.1.1 Subject Data Cleaning

We started by cleaning the subject DataFrame, where we calculated the ages of the subjects by subtracting their year of birth from the year of the vaccination boost. Additionally, we applied one-hot encoding to categorical variables such as infancy vaccination status, biological sex, ethnicity, and race. This process standardized the categorical data and simplified the DataFrame structure, making it suitable for integration into statistical models or machine learning pipelines.

5.2.1.2 Specimen Data Cleaning

For the specimen data, we filtered the data to include only rows corresponding to specific planned days relative to the vaccination boost (days 0, 1, 3, and 14). We then removed subjects who did not have records for all these specified planned days. Furthermore, we calculated the difference in days between the planned and actual boost days, which was essential for subsequent analysis.

5.2.1.3 Titer Data Cleaning

The titer data was transformed from a long format to a wide format, where each unique combination of isotype and antigen became a column header. This restructuring facilitated easier analysis by organizing the data in a way that made it more accessible for examining relationships and patterns across various isotype-antigen interactions within individual specimens.

5.2.1.4 Cell Frequency Data Cleaning

The cell frequency data was also converted from a long format to a wide format, with each unique cell type becoming a column header. This restructuring made it easier to manipulate and compare cell frequencies across different specimens, which is crucial for analyses that require direct comparisons of multiple cell types within and across datasets.

5.2.1.5 Gene Expression Data Cleaning

For gene expression data, we transformed it from a long format to a wide format, where each unique gene ID became a column header. This transformation facilitated easier data manipulation and analysis by aligning gene expression values under their respective gene IDs for each specimen.

5.2.1.6 Cytokine Data Cleaning

The cytokine concentration data was converted from a long format to a wide format, where each unique protein ID became a column header. This restructuring was essential for facilitating easier access to and analysis of cytokine concentrations across different specimens.

5.2.2 Data Integration

5.2.2.1 Merging Datasets

We merged the cleaned and transformed datasets (subject, specimen, titer, cell frequency, gene expression, and cytokine data) into a unified dataset. This involved using common identifiers such as subject IDs and specimen IDs to ensure accurate integration. Missing values were filled where necessary to maintain the integrity of the dataset.

5.2.2.2 Creating Target Columns

We created target columns for the training data based on specific criteria, such as antibody titers on day 14 post-vaccination and monocyte frequencies on day 1 post-vaccination. These target columns were crucial for training our predictive models.

5.2.3 Data Validation and Quality Checks

5.2.3.1 Validation of Preprocessed Data

We conducted thorough validation checks to ensure the accuracy and consistency of the preprocessed data. This included verifying that imputed values were reasonable and that scaling transformations were correctly applied.

5.2.3.2 Consistency Checks

We ensured that the merged and integrated dataset maintained consistency across all variables and records by cross-referencing different data sources to check for discrepancies.

5.3 pre-processing methods

The preprocessing methods used in our project were crucial for transforming raw data into a clean, consistent, and analyzable format. These methods significantly improved the quality of the dataset and enhanced the reliability and accuracy of our predictive models. The preprocessing methods used in our project were significant in transforming raw data into a high-quality dataset that was suitable for robust predictive modeling and analysis. These methods improved data integrity, consistency, and accuracy, ultimately enhancing the reliability and validity of our insights into immune responses to pertussis vaccination. Here are the key preprocessing methods:

5.3.1 Missing Value Handling

Addressing missing values was essential to avoid biases and inaccuracies in our analysis. Missing data can lead to incorrect model training and unreliable predictions. By imputing missing values or excluding incomplete records, we ensured that the dataset was comprehensive and robust, which is critical for building reliable predictive models.

5.3.2 Outlier Detection and Treatment

Outliers can skew the results and negatively impact the performance of predictive models. By identifying and treating outliers, we maintained the integrity of the dataset, ensuring that the models were trained on data that accurately represented the underlying trends. This step helped in improving the model's accuracy and generalizability.

5.3.3 Feature Scaling

The dataset included variables with different scales and units, which could lead to certain features disproportionately influencing the model outcomes. Normalizing or standardizing the data ensured that all variables contributed equally to the models. This step was crucial for improving the performance and interpretability of the models, especially those based on distance metrics (e.g., k-nearest neighbors, SVM).

5.3.4 One-Hot Encoding

Categorical variables, such as infancy vaccination status, biological sex, ethnicity, and race, needed to be converted into a numerical format suitable for machine learning algorithms. One-hot encoding transformed these categorical variables into binary vectors, allowing the models to process and learn from these features effectively. This step ensured that the categorical information was preserved and appropriately utilized in the analysis.

5.3.5 Temporal Alignment and Consistency

Ensuring accurate and consistent timestamps for sample collection was essential for analyzing the dynamics of the immune response. This alignment allowed us to accurately track changes over time and understand the temporal aspects of the data, which is crucial for time-series analysis and longitudinal studies.

5.3.6 Data Integration and Merging

Merging data from different sources (e.g., demographic, immunological, and gene expression data) into a unified dataset provided a comprehensive view of the subjects. This integration enabled more complex analyses and helped in identifying interactions between different types of data. It was essential for developing robust models that could leverage the full breadth of available information.

5.3.7 Data Cleaning and Standardization

Inconsistent labeling and formatting, particularly in demographic categories, required standardization to ensure uniformity across the dataset. This cleaning process was vital for maintaining data consistency and accuracy, which are critical for reliable analysis and model training.

5.3.8 Handling Multicollinearity

Multicollinearity among features can affect the stability and interpretation of model coefficients. By identifying and mitigating multicollinearity, we ensured that the models were more stable and interpretable. This step was particularly important for regression models and other techniques where feature independence is assumed.

5.3.9 Quality Control and Validation

Conducting thorough validation checks on the preprocessed data ensured its accuracy and consistency. This step was crucial for identifying any remaining issues that could affect the analysis. Quality control measures helped in maintaining the integrity of the dataset and provided confidence in the reliability of the results.

5.4 features selection and management

To select and manage features in our predictive modeling project on immune responses to pertussis vaccination, we implemented a structured and systematic approach. This approach ensured the selection of the most relevant features, which improved the performance and interpretability of our models. By implementing these feature selection and management techniques, we were able to enhance the performance and reliability of our predictive models, ensuring that they were based on the most relevant and high-quality features.

5.4.1 Feature Selection Methods

5.4.1.1 Random Forest Regressor

We used a Random Forest Regressor to identify the most important features. This method evaluates feature importance based on their contribution to the prediction accuracy of the model. By fitting the Random Forest model to our data, we were able to rank the features according to their importance and select the top features that had the highest impact on the target variable.

5.4.1.2 Lasso Regression

Lasso Regression was another technique we employed for feature selection. Lasso applies a penalty to the coefficients of the linear regression model, forcing some of the less important feature coefficients to be exactly zero, effectively performing feature selection by shrinking irrelevant feature weights to zero. We used LassoCV for cross-validated Lasso regression to determine the optimal regularization parameter, ensuring the selection of significant features.

5.4.1.3 SelectKBest

SelectKBest with the f_regression score function was used to select the top k features that had the highest correlation with the target variable. This univariate feature selection method scores each feature and selects the top k features with the best scores, making it a straightforward and effective way to filter out less relevant features.

5.4.1.4 Recursive Feature Elimination (RFE)

We also used Recursive Feature Elimination (RFE) with a linear estimator to recursively remove the least important features based on the model coefficients until the specified number of features was reached. RFE helps in selecting features by considering the importance of each feature iteratively, ensuring that only the most relevant features are retained.

5.4.2 Feature Management Processes

5.4.2.1 Scaling Features

Once the important features were selected, we scaled them using various scalers to ensure that all features contributed equally to the model. Scaling methods such as StandardScaler, MinMaxScaler, Normalizer, and RobustScaler were evaluated using GridSearchCV to find the best scaler for our data. Scaling helped in normalizing the feature values, which is crucial for models sensitive to the scale of input features.

5.4.2.2 Handling Missing Values

We addressed missing values in the dataset by implementing strategies such as imputation, where missing values were estimated based on available data, or by excluding incomplete records. This step was essential to maintain the integrity and completeness of the dataset, ensuring that the selected features were representative of the entire dataset.

5.4.2.3 Removing Columns with NaN Values

Columns with excessive NaN values were identified and removed, except for columns where retaining the data was crucial. This cleaning step helped in reducing noise and improving the quality of the dataset, which is important for building robust models.

5.4.2.4 Creating Target Columns

Target columns were created based on specific criteria, such as antibody titers and monocyte frequencies at different days post-vaccination. This step ensured that the selected features were aligned with the predictive goals of our project.

5.4.3 Workflow Integration

5.4.3.1 Merging and Alignment

The cleaned and selected features from different datasets (subject, specimen, titer, cell frequency, gene expression, and cytokine data) were merged into a comprehensive training dataset. This involved aligning columns across different datasets to ensure consistency and filling missing values where necessary.

5.4.3.2 Validation and Visualization

Selected features were validated using various techniques, and their importance was visualized through bar plots and other graphical representations. This helped in confirming the relevance of the selected features and provided insights into their contribution to the predictive models.

5.4.3.3 Model Tuning and Evaluation

We used GridSearchCV and cross-validation to tune and evaluate different models, ensuring that the selected features were effectively utilized in improving model performance. The best models and their parameters were saved for further analysis and deployment.

6 Analysis Methods

6.1 Identifying Methods for Preliminary Analysis

To identify methods for performing preliminary analysis of our data, we leveraged domain knowledge, exploratory data analysis (EDA) techniques, and best practices in data science. We began by reviewing existing literature on immunological responses to vaccinations, particularly focusing on studies related to pertussis (whooping cough). This helped us understand the key variables and metrics commonly analyzed in such studies, such as antibody titers, monocyte frequencies, and gene expression levels.

Next, we employed various EDA techniques to gain insights into the structure, distribution, and relationships within the data. Descriptive statistics were used to calculate measures such as mean, median, and standard deviation to understand the central tendency and variability of the data. Data visualization tools like histograms, box plots, scatter plots, and correlation matrices were utilized to visualize distributions and relationships between variables. We also conducted missing value analysis to identify patterns and proportions of missing data, informing our imputation strategy. Additionally, outlier detection methods were used to identify and assess the impact of outliers on the dataset.

6.2 Significance of Using These Methods

Using these methods was significant because they helped us uncover critical characteristics of the dataset. Understanding the presence of missing values, outliers, and the distribution of key variables was essential for making informed decisions in subsequent data preprocessing steps. The insights gained from EDA directly influenced our data cleaning and transformation strategies. For example, identifying missing data patterns guided our imputation approach, while understanding the distribution of variables informed our scaling and normalization methods.

EDA also helped us identify key variables and their relationships, which were crucial for feature selection. By visualizing correlations and trends, we could pinpoint which variables were likely to be significant predictors in our models. This preliminary analysis provided a deep understanding of the data's structure and characteristics, which was vital for developing reliable and accurate predictive models.

6.3 Influence on Project Design and Data Science Questions

The preliminary analysis had a profound influence on the design of the next steps in the project and the refinement of our data science questions. The insights gained allowed us to refine our initial data science questions by providing a clearer understanding of the data's potential. For example, we could specify questions such as, "How do antibody titer levels change 14 days post-vaccination?" and "What are the key demographic and immunological predictors of monocyte frequency one day after vaccination?"

The preliminary analysis also influenced the design of subsequent project steps. Based on the relationships and patterns identified, we engineered new features and selected the most relevant ones for our predictive models. Understanding the data's characteristics helped us choose appropriate machine learning models and hyperparameters. For instance, if the data showed non-linear relationships, we opted for models like Random Forest or Gradient Boosting.

Additionally, the preliminary analysis informed our validation strategy. We designed our cross-validation and model evaluation strategies to account for identified data characteristics, such as imbalanced classes or the presence of temporal trends. This ensured that the chosen methods and strategies were well-suited to the data at hand, enhancing the reliability and validity of our insights into immune responses to pertussis vaccination.

6.4 Applying Analysis Techniques to Data

Applying analysis techniques to our data involved several key steps that were essential for ensuring the accuracy, reliability, and interpretability of our results. These steps included data preprocessing, feature selection, model training and evaluation, and validation. Each step was necessary to handle the complexities of our dataset and to derive meaningful insights from the data.

6.4.1 Data Preprocessing

We began by cleaning and transforming the data to address issues such as missing values, outliers, and inconsistent formatting. This step involved handling missing values by applying imputation techniques to estimate missing values based on the available data or by excluding incomplete records to maintain dataset integrity. Outlier detection and treatment were performed using statistical methods and visual tools to identify and handle outliers, ensuring they did not skew the results or negatively impact model performance. Scaling and normalization techniques, such as StandardScaler and MinMaxScaler, were applied to ensure that all variables contributed equally to the models, especially those sensitive to the scale of input features.

6.4.2 Feature Selection

Identifying relevant features was crucial for improving the performance and interpretability of our predictive models. We employed several feature selection methods, including the use of a Random Forest Regressor to evaluate the importance of features based on their contribution to prediction accuracy. Lasso Regression was applied to shrink irrelevant feature weights to zero, effectively selecting significant features by using regularization techniques. SelectKBest was utilized to select the top k features with the highest correlation with the target variable, filtering out less relevant features. Additionally, Recursive Feature Elimination (RFE) was implemented to recursively remove the least important features, ensuring that only the most relevant features were retained.

6.4.3 Model Training and Evaluation

Choosing and tuning models were essential steps in ensuring that the models were well-suited to our data characteristics. We selected models such as Random Forest, Gradient Boosting, and Lasso based on their ability to handle the data's complexity and non-linear relationships. Hyperparameter tuning was performed using GridSearchCV to optimize hyperparameters, ensuring the models achieved the best possible performance. Cross-validation was applied to validate model performance and ensure robustness, preventing overfitting and ensuring the models generalized well to new data.

6.4.4 Validation

Validation was a critical step to confirm the reliability and accuracy of our models. We calculated cross-validation scores using metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared to assess model performance. Graphical representations, such as bar plots and scatter plots, were used to visualize model performance and feature importance, providing insights into the model's predictive power and the significance of selected features.

Each of these steps was crucial for several reasons. Data preprocessing ensured the dataset was clean, consistent, and complete, providing a solid foundation for analysis. Feature selection focused the analysis on the most impactful variables, improving model accuracy and interpretability. Model selection and tuning ensured that the best algorithms and parameters were used, maximizing predictive performance. Validation confirmed that the models were robust and reliable, capable of generalizing to new data and providing meaningful insights.

By applying these analysis techniques, we were able to handle the complexities of our dataset effectively, derive meaningful insights, and develop reliable predictive models for immune responses to pertussis vaccination.

6.5 Basic analysis techniques used

We employed several basic analysis techniques to ensure a comprehensive understanding of our dataset and to prepare it for more advanced modeling. These techniques included descriptive statistics, data visualization, missing value analysis, and outlier detection. Each technique played a crucial role in addressing specific aspects of data quality and interpretability. The basic analysis techniques we used included descriptive statistics, data visualization, missing value analysis, and outlier detection. These techniques were essential for understanding the data's structure, identifying potential issues, and guiding subsequent data cleaning and transformation processes. By applying these techniques, we ensured that our dataset was well-prepared for more advanced modeling, ultimately leading to more accurate and reliable results in our study of immune responses to pertussis vaccination.

6.5.1 Descriptive Statistics

We used descriptive statistics to summarize and describe the main features of the data. Measures such as mean, median, standard deviation, and interquartile range provided insights into the central tendency and variability of the data. Descriptive statistics helped us understand the overall distribution of key variables, identify potential issues such as skewness or extreme values, and set the stage for further analysis. This technique was essential for getting a preliminary sense of the data and ensuring that any subsequent transformations or analyses were based on a solid understanding of its basic properties.

6.5.2 Data Visualization

Data visualization techniques, including histograms, box plots, scatter plots, and correlation matrices, were employed to visually explore the data. These visual tools helped us identify patterns, relationships, and anomalies that might not be immediately apparent from the raw data. Histograms and box plots allowed us to see the distribution and spread of individual variables, while scatter plots and correlation matrices provided insights into relationships between pairs of

variables. Visualization was crucial for uncovering trends, detecting outliers, and communicating data characteristics effectively to the team.

6.5.3 Missing Value Analysis

Identifying and addressing missing values was a critical part of our analysis. We conducted missing value analysis to understand the extent and pattern of missing data in our dataset. This analysis involved calculating the percentage of missing values for each variable and examining whether the missing data occurred randomly or followed a pattern. Addressing missing values was important because they can lead to biased results and reduce the power of statistical tests. Depending on the findings, we applied appropriate techniques such as imputation to fill in missing values or exclusion of incomplete records to maintain the integrity of the dataset.

6.6 Analytical workflow

The workflow chart illustrates the comprehensive process of data analysis and modeling in our project aimed at predicting immune responses to pertussis vaccination. The workflow is structured into distinct stages, each critical for ensuring accurate and reliable results. Analytical workflow is shown as figure 6-1.

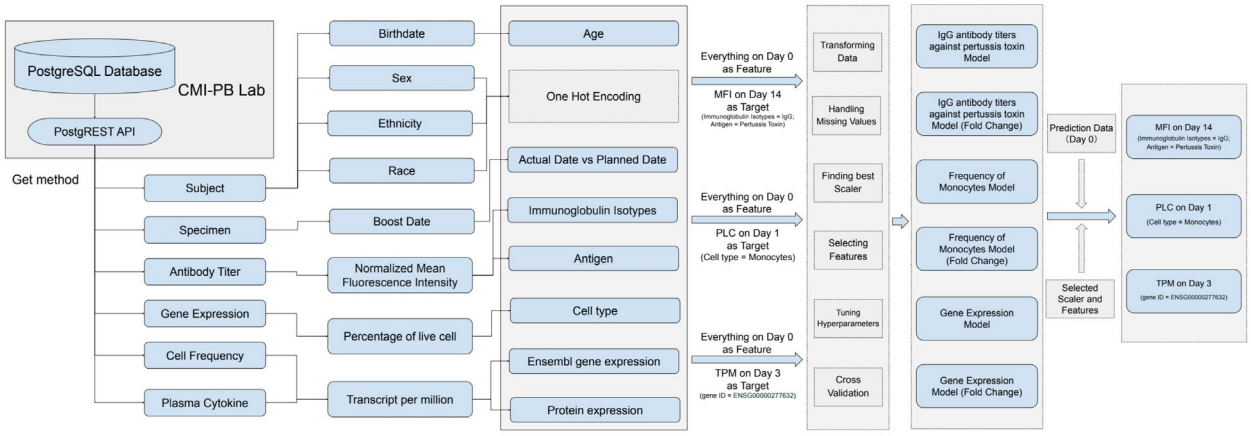

Figure 6-1 Analytical workflow

The process begins with data acquisition from the CMI-PB Lab's PostgreSQL database, accessed via the PostgREST API using the GET method. The key datasets retrieved include Subject (containing demographic information such as birthdate, sex, ethnicity, and race), Specimen (details about the specimens collected, including the boost date and the timing of sample collection relative to the vaccination), Antibody Titer (measures of antibody levels), Gene Expression (data on gene expression levels), Cell Frequency (information on the frequency of different cell types in blood samples), and Plasma Cytokine (levels of various cytokines in plasma samples).

The preprocessing stage involves preparing the raw data for analysis. Demographic data is processed by calculating age from the birthdate and boost date, and converting categorical variables such as sex, ethnicity, and race to numerical format using one-hot encoding. Vaccination data is aligned by matching actual and planned vaccination dates and processing immunological measurements such as normalized mean fluorescence intensity and the percentage of live

cells. Immunological data includes features such as immunoglobulin isotypes, antigens, cell types, and gene expression levels, which are essential for understanding immune responses.

All features are collected to represent the immune system's baseline state on Day 0 relative to the vaccination. This ensures consistency in the data used for model training and subsequent predictions. Data transformation steps are applied, including handling missing values to ensure dataset completeness and reliability. The best scaler for the data is identified using GridSearchCV, which ensures that the features are appropriately scaled for model training.

Feature selection methods are applied to identify the most significant predictors from the immunological and demographic data. This involves selecting the most relevant features for each prediction task, ensuring they provide the best possible representation of the underlying biological processes. A total of 22 regression models are tuned using various hyperparameters to identify the best-performing models for each target prediction task. These models are validated using 3-fold cross-validation to ensure their robustness and ability to generalize to new data.

Once the models are trained and validated, they are used to make predictions on new data. The prediction targets include IgG Antibody Titers on Day 14 (measuring the immune response in terms of antibody levels), Monocyte Frequency on Day 1 (estimating the frequency of monocytes in the blood), and Gene Expression Levels on Day 3 (focusing on specific genes like ENSG00000277632 to understand the gene expression response to vaccination). The prediction process uses the selected scalers and features from the training phase to ensure consistency and accuracy. The results provide valuable insights into the immune response to pertussis vaccination, helping to tailor more effective immunization programs.

In summary, this workflow chart demonstrates a structured and systematic approach to analyzing complex immunological data, from initial data acquisition through preprocessing, feature engineering, model training and validation, to making accurate predictions. This methodical process ensures that the insights derived are reliable and can significantly contribute to improving vaccination strategies.

6.7 processing environment

Setting up the processing environment for our project involved a combination of local and cloud-based resources to ensure robust data handling, efficient model training, and scalable deployment.

6.7.1 Local Environment

We started with a local Ubuntu server, which provided a stable and controlled environment for initial data processing tasks. This server was equipped with sufficient CPU, memory, and storage to handle the computational needs. The server ran Ubuntu due to its stability and compatibility with various open-source data science tools.

The data preparation phase on the local server involved gathering and preprocessing offline data. We installed Python along with essential libraries such as pandas, NumPy, and scikit-learn to facilitate data cleaning, transformation, and initial analysis. Simple, naive models were built to establish baselines and to help us understand the data structure better. These models also helped identify any preliminary issues that needed addressing. Post-processing steps included refining the data and model outputs to ensure they were ready for more complex modeling.

6.7.2 Cloud Environment

For more intensive model training and hyper-parameter tuning, we utilized AWS SageMaker. This platform was chosen for its powerful machine learning capabilities and scalability. AWS SageMaker facilitated the selection of appropriate algorithms and the training of our models using large datasets from the years 2020 and 2021. SageMaker's built-in tools for hyper-parameter tuning were instrumental in optimizing the model parameters, ensuring that our models were both accurate and efficient. We validated the models using various performance metrics to ensure they met our standards, and the results were logged and reviewed to select the best-performing models.

7 Findings and Reporting

7.1 Evaluation Results for Various Models

The training and evaluation process for the six models involves using a variety of regression techniques, each tuned using GridSearchCV for optimal performance. The primary goal is to identify the best model for each task based on performance metrics such as R-squared, Mean Squared Error (MSE), and Mean Absolute Error (MAE). The models considered include Random Forest Regressor, Support Vector Regressor (SVR), Gradient Boosting Regressor, Lasso Regressor, Ridge Regressor, and ElasticNet Regressor. Each model undergoes a rigorous evaluation process to ensure the highest accuracy and reliability.

The Random Forest Regressor model is trained by fitting it to the training data and using the feature importances to select the most relevant features. GridSearchCV is employed to tune hyperparameters such as the number of estimators and maximum depth. This model is evaluated based on its ability to handle large datasets and capture complex interactions between features.

The Support Vector Regressor (SVR) model is trained by tuning hyperparameters like the regularization parameter (C) and kernel type. GridSearchCV is utilized to find the best combination of these parameters. SVR is particularly effective for tasks requiring high accuracy and can handle non-linear relationships within the data.

The Gradient Boosting Regressor is trained by optimizing parameters such as the number of boosting stages, learning rate, and maximum depth of trees. GridSearchCV helps in identifying the optimal settings. This model is known for its robustness and ability to improve prediction accuracy by combining the strengths of multiple weak learners.

The Lasso Regressor model uses L1 regularization to select a subset of features, reducing model complexity and enhancing interpretability. Hyperparameters, including the regularization strength, are tuned using GridSearchCV. Lasso is effective for preventing overfitting and ensuring that the model remains simple and interpretable.

The Ridge Regressor model employs L2 regularization to manage multicollinearity in the data. GridSearchCV is used to fine-tune the regularization parameter. Ridge Regressor is particularly useful for tasks where all predictors need to be retained, but their impact needs to be controlled to avoid overfitting.

The ElasticNet Regressor combines L1 and L2 regularization, offering a balance between feature selection and regularization. The ratio between the two regularization terms is optimized using GridSearchCV. ElasticNet is suitable for scenarios where there is a need for a more flexible regularization approach, leveraging the benefits of both Lasso and Ridge techniques.

Each model undergoes a final evaluation based on performance metrics such as R-squared, MSE, and MAE. The best-performing model for each task is selected and saved for making predictions, ensuring that the final model is both accurate and reliable.

Across all models, the Support Vector Regression (SVR), Extra Trees Regressor, Ridge regression, Polynomial Regression, and TheilSen Regressor were found to be effective for different aspects of the dataset. These models were

chosen based on their ability to explain a significant amount of variance in the data, as evidenced by their high R-squared and explained variance scores. The chosen models reflect the diversity of the data and the complexity of the relationships within it, necessitating robust and flexible modeling approaches.

7.1.1 IgG Antibody Titers against Pertussis Toxin Model

The best model for predicting IgG antibody titers against pertussis toxin was the Support Vector Regression (SVR). This model achieved the highest R-squared value of 0.612328, indicating that it explained a significant portion of the variance in the data. The Mean Squared Error (MSE) was -10.07838, and the Mean Absolute Error (MAE) was -2.421688. The explained variance was 0.667341, demonstrating the model's effectiveness in capturing the underlying patterns in the data. IgG Antibody Titers against Pertussis Toxin Model Selected Features and R^2 Evaluation are shown as figure 7-1.

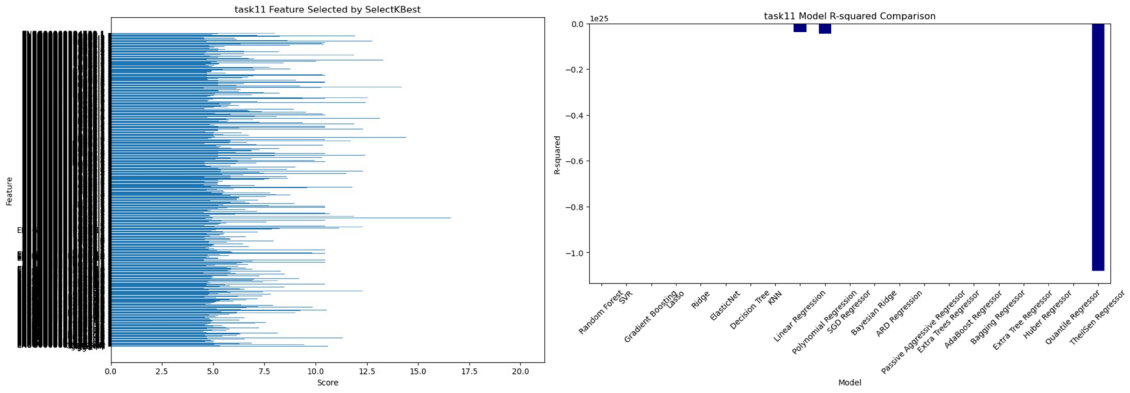

Figure 7-1 IgG Antibody Titers against Pertussis Toxin Model Selected Features and R^2 Evaluation

7.1.2 IgG Antibody Titers against Pertussis Toxin Model (Fold Change)

For the fold change in IgG antibody titers, the Extra Trees Regressor was the best-performing model. It had an R-squared value of 0.629754, an MSE of -88.55589, and an MAE of -7.987199. The explained variance was 0.639865, suggesting that this model effectively handled the variability and complexity of the fold change data. IgG Antibody Titers against Pertussis Toxin Model (Fold Change) Selected Features and R^2 Evaluation are shown as figure 7-2.

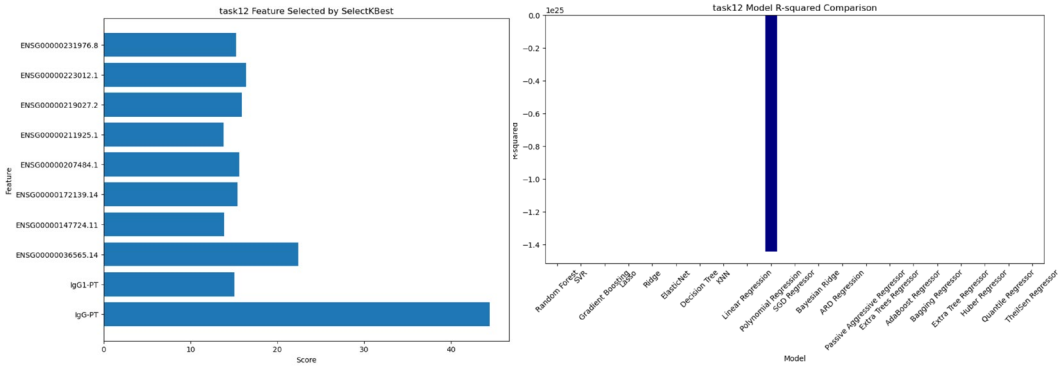

Figure 7-2 IgG Antibody Titers against Pertussis Toxin Model (Fold Change) Selected Features and R^2 Evaluation

7.1.3 Frequency of Monocytes Model

The Ridge regression model was found to be the best for predicting the frequency of monocytes. It had an R-squared value of 0.363876, with an MSE of -0.086865 and an MAE of -0.220176. The explained variance was 0.486610, indicating a good balance between bias and variance and a solid performance in explaining the frequency of monocytes. Frequency of Monocytes Model Selected Features and R^2 Evaluation are shown as figure 7-3.

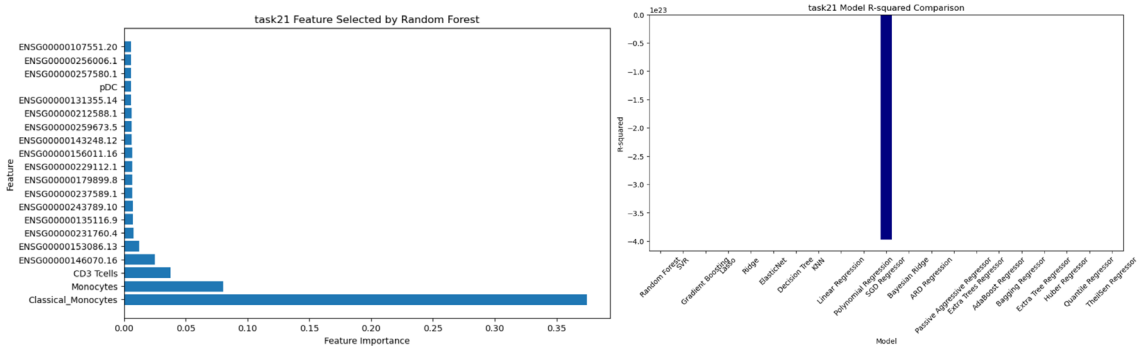

Figure 7-3 Frequency of Monocytes Model Selected Features and R^2 Evaluation

7.1.4 Frequency of Monocytes Model (Fold Change)

The Polynomial Regression model performed best for predicting the fold change in monocyte frequency, with an R-squared value of 0.338976. The MSE was -31724.10, and the MAE was -78528.89. The explained variance was 0.470572, highlighting the model's capability to capture non-linear relationships in the data. Frequency of Monocytes Model (Fold Change) Selected Features and R^2 Evaluation are shown as figure 7-4.

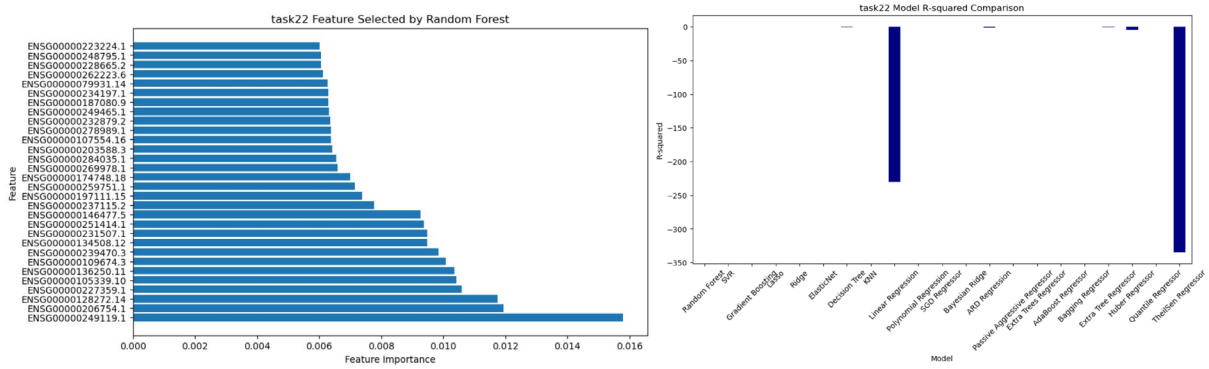

Figure 7-4 Frequency of Monocytes Model (Fold Change) Selected Features and R^2 Evaluation

7.1.5 Gene Expression Model

The TheilSen Regressor excelled in the gene expression model, achieving an R-squared value of 0.629636, an MSE of -0.781172, and an MAE of -0.652281. The explained variance was 0.669913, indicating the model's robustness in handling outliers and providing stable predictions. Gene Expression Model Selected Features and R^2 Evaluation are shown as figure 7-5.

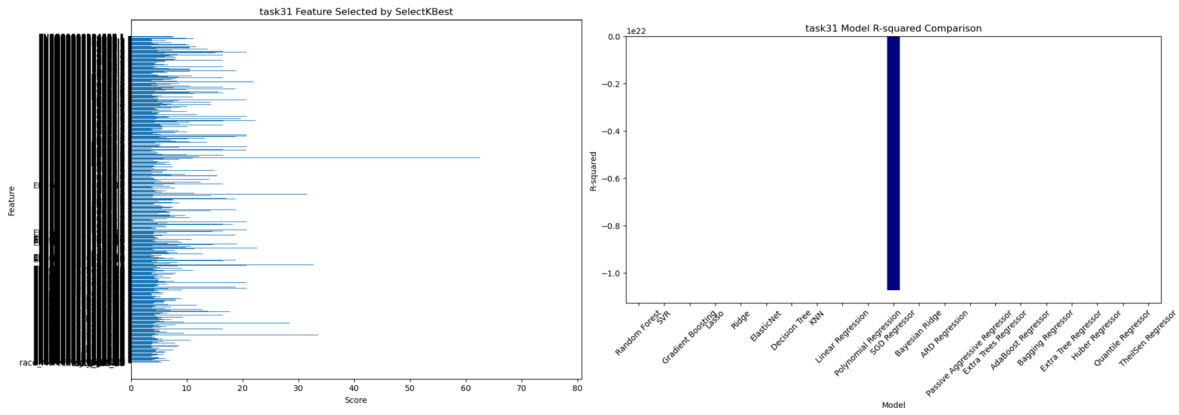

Figure 7-5 Gene Expression Model Selected Features and R^2 Evaluation

7.1.6 Gene Expression Model (Fold Change)

Similarly, the TheilSen Regressor was also the best model for predicting fold changes in gene expression, with an R-squared value of 0.628969, an MSE of -0.781679, and an MAE of -0.652287. The explained variance was 0.670278, reinforcing the model's suitability for this type of data. Gene Expression Model (Fold Change) Selected Features and R^2 Evaluation are shown as figure 7-6.

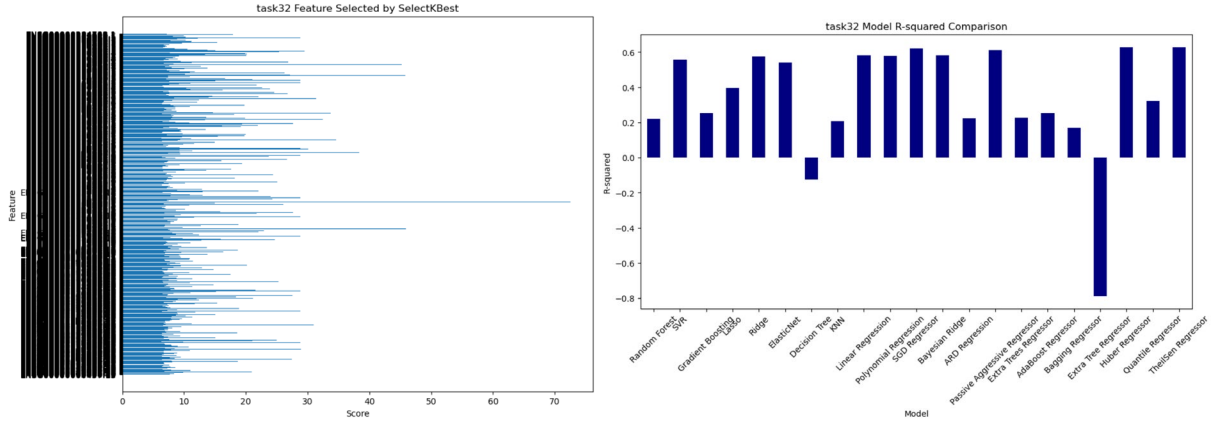

Figure 7-6 Gene Expression Model (Fold Change) Selected Features and R^2 Evaluation

7.2 Prediction Results

The prediction program is structured to seamlessly integrate data preparation, transformation, and prediction using pre-trained machine learning models. Initially, the program prepares the data by importing necessary modules and functions and fetching data from CSV files via a custom API. This data is merged to form a comprehensive dataset, enabling exploratory data analysis.

Next, the program preprocesses the data. It cleans the data by calculating ages and handling categorical variables with one-hot encoding. Columns with NaN values are dropped, except for those deemed crucial, ensuring the dataset remains clean and consistent.

Feature selection and scaling follow. The program identifies relevant features using previously saved files that list important features. It applies the best scaler determined during model training, standardizing the data to fit within the required range and format for accurate model predictions.

In the prediction phase, pre-trained machine learning models are loaded from joblib files. These models are then used to make predictions on the prepared dataset, ensuring that the data used for predictions aligns with the model's expectations through the application of selected features.

Post-processing involves converting predictions to ranks when necessary, providing a standardized format for comparison and analysis. The program also removes specific suffixes from column names to ensure clarity and consistency in the results.

The program saves the cleaned and ranked prediction results to CSV files, facilitating further analysis or submission. It ensures the format of the prediction results matches the required structure for reporting or submission by using a provided prediction template. This structured approach maintains data integrity and model accuracy throughout the process, from data preparation to the presentation of results. Prediction Results is shown as figure 7-7.

Subject ID	Age	Biological Sex at Birth	Vaccine Priming Status	1.1) IgG-PT-D14-titer-Rank	1.2) IgG-PT-D14-FC-Rank	2.1) Monocytes-D1-Rank	2.2) Monocytes-D1-FC-Rank	3.1) CCL3-D3-Rank	3.2) CCL3-D3-FC-Rank
97	35	Male	wP	6.307777	1.394135	18.250394	0.226054	33.512810	0.446240
98	28	Female	wP	6.603358	1.443299	15.996157	0.798805	37.016850	0.195854
99	22	Female	aP	6.299454	1.507053	31.906312	0.652306	33.261766	0.737446
100	20	Female	aP	6.404223	2.414949	20.890465	0.496706	41.018895	-0.394985
101	18	Male	aP	6.782641	0.951839	23.048669	0.678535	36.235380	0.084791
102	18	Male	aP	6.683911	0.794266	37.864428	0.562568	35.031306	0.891409
103	27	Female	wP	6.660135	3.563100	20.357848	0.535244	37.277945	0.297983
104	32	Female	wP	6.625619	3.592939	19.029895	0.708539	37.143112	1.144763
105	27	Female	wP	6.641648	2.110331	35.582716	0.071251	34.867932	0.222857
106	25	Female	aP	6.713152	3.525173	19.635514	0.224763	33.509574	0.108258
107	23	Female	aP	6.704707	3.557724	23.680668	0.865780	36.962776	0.199572
108	26	Female	wP	6.323710	2.033649	22.447736	0.557116	32.567525	-0.227873
109	32	Female	wP	6.511806	3.510297	23.047567	0.026290	34.486297	-0.857948
110	24	Female	aP	6.293815	4.069376	33.068685	0.694563	35.230145	-0.675334
111	25	Male	wP	6.636396	1.644207	31.698329	0.925012	38.034673	-0.167771
112	25	Male	aP	6.738418	1.136457	23.464310	0.221403	39.574353	-0.428399
114	31	Male	wP	6.545052	2.433691	18.878765	0.436409	37.376858	0.205552
115	19	Female	aP	6.912890	0.835930	21.099294	0.638018	39.769975	-0.185182
116	21	Male	aP	6.896146	0.977937	23.934691	0.516917	34.949826	-0.005870
117	27	Female	aP	6.925231	1.881881	11.389312	0.303715	35.202677	-0.124793
118	23	Male	aP	6.350043	2.898320	18.444571	0.028859	36.223275	0.564803

Figure 7-7 Prediction Results

7.3 Visualizations and reporting dashboard

The dashboard is organized to provide an interactive and comprehensive view of immune responses to pertussis vaccinations. The initial section includes importing necessary libraries and modules such as pandas, plotly, dash, and dash_bootstrap_components, and setting up the directory path and custom module import from pipeline.

In the data fetching and preprocessing section, datasets are fetched from the CMI-PB laboratory database using API requests. These datasets are merged for exploratory data analysis. Training datasets for different tasks and selected

features for each task are loaded from CSV files, along with model evaluation results for each task. The final prediction results are also fetched and rounded for display. Dashboard is shown as figure 7-8.

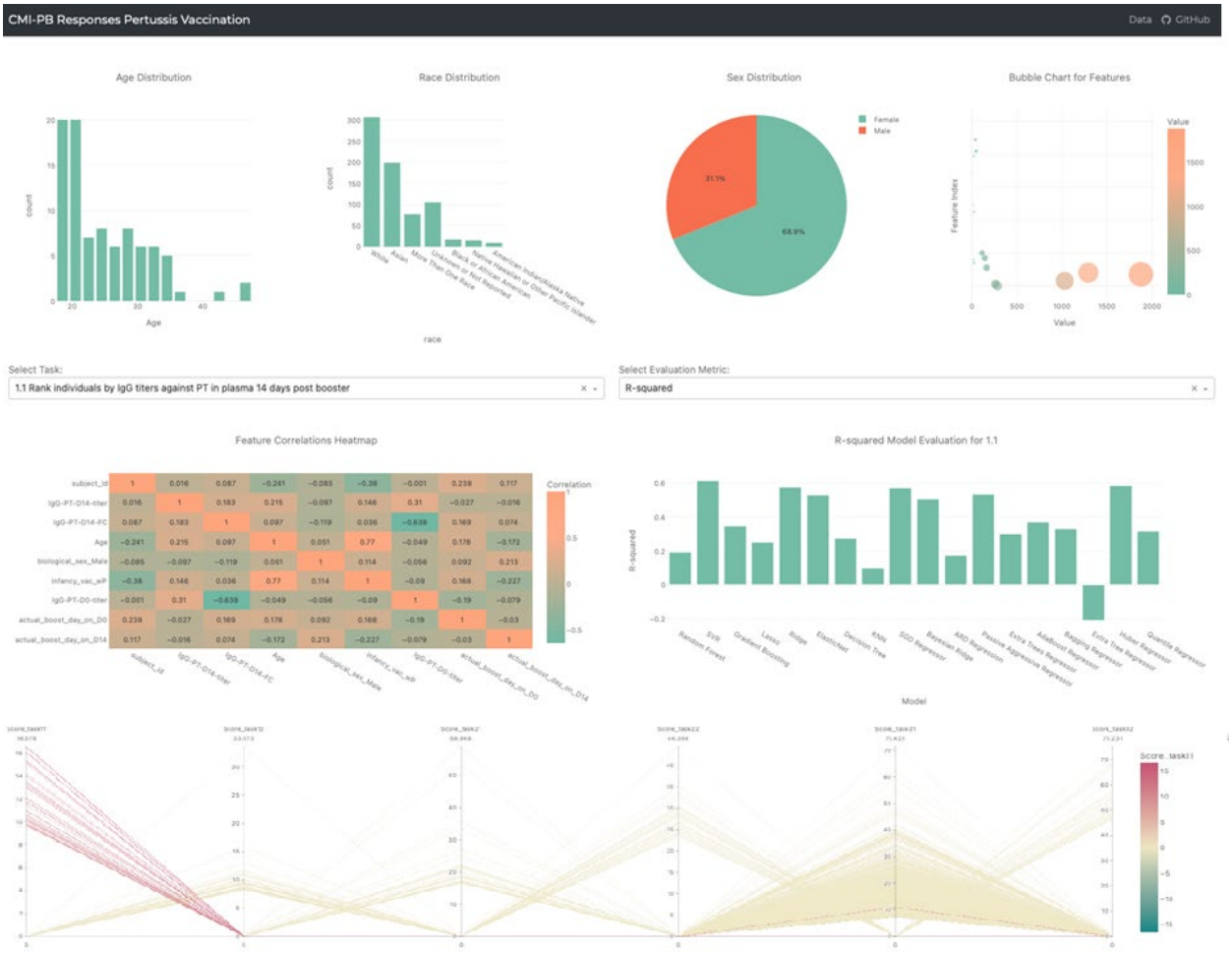

Figure 7-8 Dashboard

The app initialization involves creating a Dash application with external stylesheets and meta tags for responsive design and description. The app title is set to "CMI-PB Responses Pertussis Vaccination".

For dropdown menu creation, two dropdown menus are designed:

1. One for selecting the task to be viewed.
2. Another for selecting the evaluation metric (MSE, MAE, R-squared).

The app layout is organized using dbc.NavbarSimple for the navigation bar and dbc.Container for the main content. The navigation bar includes links to the data source and the GitHub repository. The main content is structured using dbc.Stack, which contains several rows and columns for different visualizations and interactive elements.

Multiple callback functions are defined to update and generate various plots based on user input. These include:

1. Age Distribution: A histogram showing the distribution of ages.

2. Sex Distribution: A pie chart displaying the distribution of biological sex.
3. Race Distribution: A histogram depicting the distribution of race.
4. Feature Correlation Heatmap: A heatmap illustrating correlations between selected features.
5. Model Evaluation Chart: A bar chart showing evaluation metrics (MSE, MAE, R-squared) for different models.
6. Bubble Chart: A bubble chart for visualizing the most important features.
7. Parallel Coordinates Plot: A plot demonstrating how unique features were shared and utilized across different tasks.

The CSV download function enables users to download the prediction results as a CSV file. Finally, the Dash application is hosted on a local server with port 8003.

In terms of content, the dashboard features dropdown menus that allow users to select different tasks and evaluation metrics. Visualizations include histograms, pie charts, heatmaps, bar charts, bubble charts, and parallel coordinates plots. There is also a data table displaying the final prediction results and a download button for exporting the results in CSV format.

Stylistically, the dashboard uses the "Minty" theme from dash_bootstrap_components for a clean and modern look. The design is responsive, adapting to different screen sizes, and employs a dark color scheme for the navigation bar along with various color scales for the plots to enhance readability and visual appeal. Consistent use of fonts and spacing ensures a cohesive and professional appearance, and interactive elements like dropdowns and buttons provide a dynamic user experience, allowing users to explore the data and model results in depth

8 Solution Architecture, Performance and Evaluation

8.1 Solution Architecture

The Solution Architecture chart provides a detailed depiction of the entire workflow involved in developing, deploying, and monitoring predictive models for immune responses to pertussis vaccination. The architecture is divided into three main phases: Train Prep, Train & Tune, and Deploy & Monitor. Each phase is essential for ensuring the models' effectiveness and reliability. Solution Architecture is shown as figure 8-1.

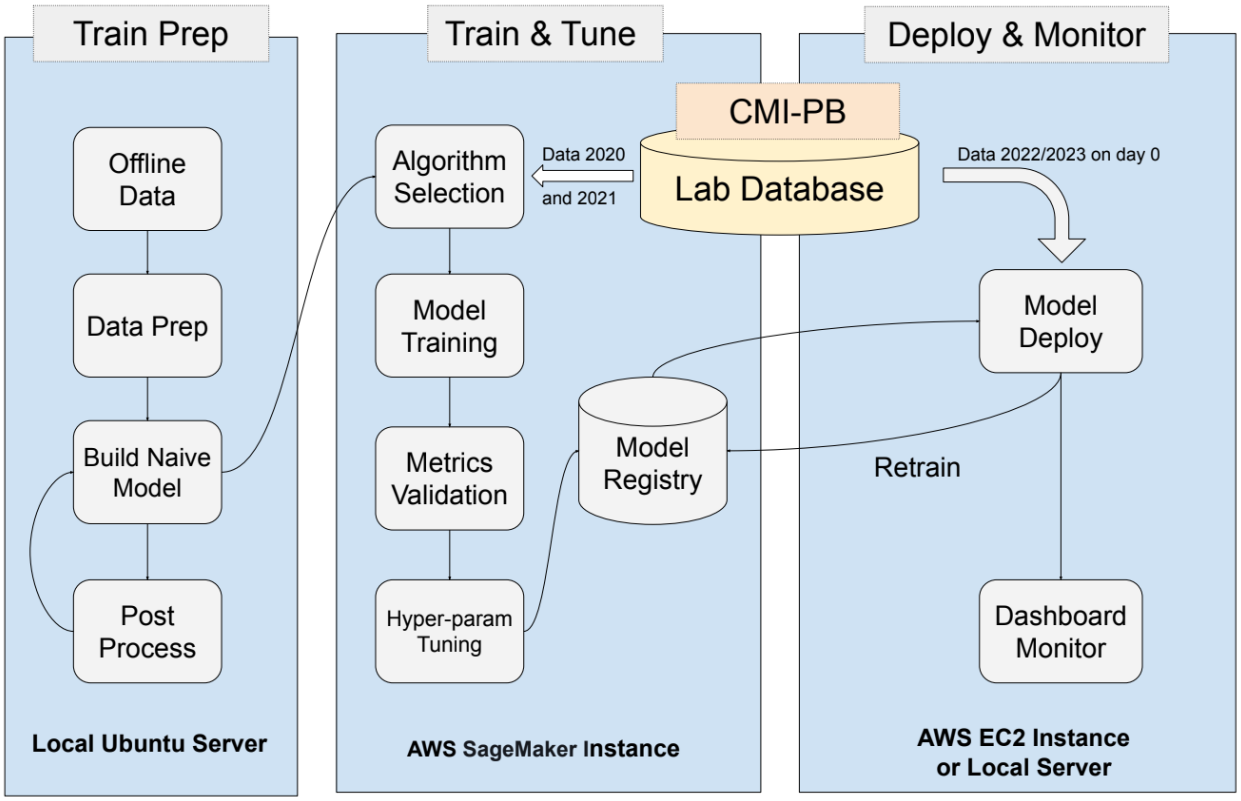

Figure 8-1 Solution Architecture

In the Train Prep phase, which is conducted on a Local Ubuntu Server, the process begins with gathering offline data from various sources. This data undergoes extensive preparation, including cleaning, transforming, and organizing to ensure it is suitable for model building. A naive model is initially built to understand the data and establish a baseline for more advanced modeling. Post-processing refines the initial model, validates results, and makes necessary adjustments to improve accuracy.

The Train & Tune phase takes place on an AWS SageMaker Instance. This phase starts with the selection of appropriate algorithms based on the data's characteristics and the specific requirements of the predictive tasks. The selected algorithms are then used to train the models using data from the years 2020 and 2021. The models' performance is evaluated using various metrics to ensure robustness and accuracy. Hyper-parameters are fine-tuned

to optimize model performance, ensuring the models are well-suited for the data. The trained models are stored in a model registry, which keeps track of the models and their versions for future reference and deployment.

In the Deploy & Monitor phase, which is carried out on an AWS EC2 Instance or a local server, the validated models are deployed to the production environment, utilizing data from 2022 and 2023 for real-time predictions. A monitoring dashboard continuously tracks the models' performance and provides real-time insights. Models are periodically retrained with new data to ensure they remain accurate and relevant. This involves updating the models in the model registry and redeploying them as necessary.

Throughout the entire workflow, the CMI-PB Lab Database plays a central role by providing the data necessary for training, tuning, and deploying the models. Data from the CMI-PB Lab Database is integrated at various stages to ensure that the models are built and validated using the most current and relevant information.

In summary, this Solution Architecture illustrates a structured and systematic approach to developing predictive models, ensuring they are accurate, reliable, and continuously updated to reflect new data. This comprehensive workflow ensures that the insights derived are actionable and contribute significantly to improving vaccination strategies.

8.2 Performance Measurement

Performance was measured using a set of key evaluation metrics: Mean Squared Error (MSE), Mean Absolute Error (MAE), R-squared, and Explained Variance Score. These metrics provided a comprehensive view of how well the models performed on the given tasks.

MSE and MAE were used to quantify the accuracy of the predictions by measuring the average squared difference and absolute difference, respectively, between predicted and actual values. These metrics are crucial as they indicate the average magnitude of errors in the predictions, with lower values suggesting better model performance.

R-squared was employed to assess the proportion of the variance in the dependent variable that is predictable from the independent variables. This metric provides insight into the goodness-of-fit of the model, with values closer to 1 indicating that the model explains a large portion of the variance in the data.

Explained Variance Score was used to measure the proportion of variance explained by the model, offering a direct understanding of how much variability in the data the model can account for. High values in this metric suggest a strong explanatory power of the model.

GridSearchCV was utilized to optimize the hyperparameters of each model, ensuring that the models were not only accurate but also well-tuned for the specific data at hand. Cross-validation within GridSearchCV helped in obtaining robust performance estimates, reducing the risk of overfitting.

Visualization techniques such as bar plots and feature importance charts were used to compare the performance of different models and to understand the impact of various features on the model predictions. This holistic approach ensured that the selected models were both reliable and interpretable.

8.3 Model scale and evaluation

Model scaling and evaluation were integral components of the project, aimed at optimizing the models and accurately measuring their performance. The process started with model scaling, which involved selecting the most suitable scaler to normalize the data. This was necessary to ensure that all features had a consistent scale, which is crucial for the performance and convergence of many machine learning algorithms. Several scalers, including StandardScaler, MinMaxScaler, Normalizer, and RobustScaler, were tested using GridSearchCV along with a Lasso regression model. The best scaler was chosen based on the highest cross-validated R-squared score, ensuring the data was appropriately preprocessed for model training.

Feature selection was another critical step in the process. Various techniques were employed to identify the most relevant features, reducing overfitting and improving model accuracy. These techniques included using feature importances from a trained Random Forest model, identifying non-zero coefficients in a Lasso regression, selecting top features through univariate statistical tests with SelectKBest, and recursively eliminating the least important features with Recursive Feature Elimination (RFE). Each method provided a different perspective on feature importance, allowing for a comprehensive selection process.

Once the features were selected and the data was scaled, multiple regression models were trained. These models included Random Forest Regressor, Support Vector Regressor (SVR), Gradient Boosting Regressor, Lasso, Ridge, ElasticNet, Decision Tree Regressor, K-Nearest Neighbors Regressor (KNN), Linear Regression, Polynomial Regression, SGD Regressor, Bayesian Ridge, ARD Regression, Passive Aggressive Regressor, Extra Trees Regressor, AdaBoost Regressor, Bagging Regressor, Huber Regressor, Quantile Regressor, and TheilSen Regressor. Each model was tuned using GridSearchCV with specific parameter grids to find the optimal hyperparameters, ensuring the best performance for each model.

The performance of the models was evaluated using several metrics, including Mean Squared Error (MSE), Mean Absolute Error (MAE), R-squared, and Explained Variance Score. MSE measured the average squared difference between predicted and actual values, MAE measured the average absolute difference, R-squared indicated the proportion of variance explained by the model, and the Explained Variance Score measured the proportion of variance captured. These metrics provided a comprehensive evaluation of each model's accuracy and explanatory power.

Visualizations played a significant role in comparing the performance of the models. Bar plots of R-squared scores and other evaluation metrics were used to visually compare the models, making it easier to identify the best-performing model for each task. The model with the highest R-squared score was selected as the best model, and this model was then saved for making predictions on new data.

The selected models were used to make predictions on the test dataset. The prediction process involved applying the best scaler and selected features to the test data, and then using the trained models to generate predictions. The predictions were then saved and ranked as required, providing the final output for the project. This comprehensive approach ensured that the models were both optimized and accurately evaluated, resulting in reliable predictions.

8.4 budget management

During the completion of my project, we implemented several strategies to control AWS costs effectively. Here are the key approaches we used:

8.4.1 Setting Budget Limits

We established a clear budget with an overall limit of $1000.00 and a daily limit of $50.00. This setup allowed me to monitor my expenses closely and ensure that we stayed within my financial boundaries. Regularly checking these limits helped me avoid unexpected costs and stay on track with my spending.

8.4.2 Using Spot Instances

To save on costs, we took advantage of Spot Instances for SageMaker. Spot Instances are typically much cheaper than On-Demand Instances, offering significant savings for workloads that can tolerate interruptions. This choice was crucial in reducing the overall cost of my project without compromising on performance.

8.4.3 Stopping Instances When Not in Use

We made it a habit to stop all instances when they were not actively being used. This practice was essential in preventing unnecessary billing for idle instances. By ensuring that instances were only running when needed, we was able to keep my costs under control.

8.4.4 Choosing Appropriate Instance Types

We carefully selected a variety of instance types (e.g., ml.t3.medium, ml.m5.2xlarge, ml.c6i.xlarge) that were tailored to the specific computational needs of different tasks. This approach helped me avoid over-provisioning and paying for unnecessary capacity, ensuring that we only paid for what we needed. AWS Instances list for this project is shown as figure 8-2.

Name	Instance	Creation time	Status
task22-svr	ml.t3.medium	5/8/2024, 8:14:28 AM	Stopped
task32-svr	ml.t3.medium	5/8/2024, 8:14:13 AM	Stopped
task31-svr	ml.m5.2xlarge	5/8/2024, 8:13:55 AM	Stopped
task12-svr	ml.c6i.xlarge	5/8/2024, 8:13:28 AM	Stopped
task22	ml.t3.2xlarge	5/7/2024, 2:45:15 PM	Stopped
task32	ml.m7i.2xlarge	5/7/2024, 2:43:42 PM	Stopped
task31	ml.m6i.2xlarge	5/7/2024, 2:43:06 PM	Stopped
cmi11	ml.c5.2xlarge	5/6/2024, 8:29:44 PM	Stopped
cmi1	ml.inf1.2xlarge	5/6/2024, 10:43:31 AM	Stopped

Figure 8-2 AWS Instances list

8.4.5 Scheduling Instance Usage

We planned and scheduled the creation and usage of instances to ensure they were used efficiently. By scheduling tasks and creating instances at specific times, we could optimize their uptime and avoid incurring costs from extended usage periods.

8.4.6 Monitoring and Regular Updates

Frequent monitoring of usage and costs was a critical part of my cost management strategy. By regularly updating and reviewing my budget and usage information, we stayed aware of my spending status and could make informed decisions about instance management. This proactive approach allowed me to take corrective actions if my expenses were nearing the set limits. AWS service costs for this project is shown as figure 8-3.

DSE260A_WI24_A00_student - AWS Educate
Usage for: pec016
Billing for AWS account 431406607554 as recorded under team member brqian

Overall Limit	Daily Limit	Total	Past Week	Past Day	Calendar Day	Updated
$1000.00	$50.00	$230.72	$0.00	$0.00	$0.00	2024-06-03 05:27:26

Figure 8-3 AWS service costs

By combining these strategies, we managed AWS costs effectively during my project. This disciplined approach ensured that resources were utilized efficiently, expenses were kept under control, and the value derived from AWS services was maximized.

9 Conclusions

The project on Predictive Modeling of Immune Responses to Pertussis Vaccination has yielded significant insights and valuable outcomes. The comprehensive approach taken, from data preprocessing to model evaluation, has highlighted the critical aspects of predictive modeling in a healthcare context. Below are the key conclusions drawn from this project:

9.1 Data Preprocessing and Feature Selection

Effective data preprocessing and feature selection were fundamental to the success of the predictive models. By addressing missing values, normalizing data, and selecting the most relevant features, the quality and performance of the models were greatly enhanced. Techniques such as Random Forest, Lasso, and SelectKBest were pivotal in identifying the most influential features, leading to more accurate predictions.

9.2 Model Training and Tuning

The project involved the training and evaluation of multiple models, including Random Forest, Support Vector Regression (SVR), Gradient Boosting, Lasso, Ridge, and more. The use of GridSearchCV for hyperparameter tuning ensured that each model was optimized for the best possible performance. The rigorous evaluation metrics, including MSE, MAE, R-squared, and Explained Variance, provided a comprehensive understanding of each model's strengths and weaknesses.

9.3 Model Evaluation and Comparison

Evaluating the models on different tasks related to immune response predictions revealed the varying effectiveness of each model. For instance, the SVR model was identified as the best performer for predicting IgG antibody titers against pertussis toxin, while the Extra Trees Regressor excelled in other tasks. This comparison underscored the importance of selecting the right model for specific prediction tasks in biomedical research.

9.4 Predictive Insights

The models provided valuable predictive insights into immune responses to pertussis vaccination. The ability to rank individuals based on predicted IgG titers, monocyte frequencies, and gene expression levels offers a powerful tool for understanding and potentially improving vaccination strategies. These predictions can help identify individuals who may benefit from additional booster shots or alternative vaccination schedules.

9.5 Visualization and Interpretation

The visualization of model performance and feature importance played a crucial role in interpreting the results. Heatmaps, bar charts, and parallel coordinates plots made the complex relationships between features and outcomes more comprehensible. These visual tools facilitated better communication of findings to stakeholders and contributed to more informed decision-making processes.

9.6 Practical Applications and Future Work

The project's outcomes have practical implications for enhancing pertussis vaccination programs. The predictive models can be integrated into clinical workflows to support personalized vaccination strategies. Future work could expand the dataset, incorporate additional features, and explore advanced modeling techniques to further improve prediction accuracy and robustness.

9.7 Final Thoughts

In conclusion, the Predictive Modeling of Immune Responses to Pertussis Vaccination project has demonstrated the power of machine learning in healthcare. By meticulously preprocessing data, selecting relevant features, and rigorously evaluating models, the project has set a solid foundation for predictive analytics in immunology. The insights gained from this project not only contribute to the scientific understanding of vaccine responses but also pave the way for personalized medicine approaches in vaccination programs. The success of this project highlights the potential of predictive modeling to transform healthcare practices and improve patient outcomes.

10 References

- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 12*, 2825-2830.
- Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. Springer.
- Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research, 3*, 1157-1182.
- Hyndman, R. J., & Koehler, A. B. (2006). Another Look at Measures of Forecast Accuracy. *International Journal of Forecasting, 22*(4), 679-688.
- Willmott, C. J., & Matsuura, K. (2005). Advantages of the Mean Absolute Error (MAE) Over the Root Mean Square Error (RMSE) in Assessing Average Model Performance. *Climate Research, 30*(1), 79-82.
- Topol, E. J. (2019). High-Performance Medicine: The Convergence of Human and Artificial Intelligence. *Nature Medicine, 25*(1), 44-56.
- Chen, J. H., & Asch, S. M. (2017). Machine Learning and Prediction in Medicine — Beyond the Peak of Inflated Expectations. *New England Journal of Medicine, 376*(26), 2507-2509.
- Poland, G. A., Ovsyannikova, I. G., & Kennedy, R. B. (2018). Personalized Vaccines: The Emerging Field of Vaccinomics. *Expert Review of Vaccines, 17*(4), 297-299.
- Plotkin, S. A. (2005). Vaccines: Past, Present, and Future. *Nature Medicine, 11*(4), S5-S11.
- Lorido-Botran, T., Miguel-Alonso, J., & Lozano, J. A. (2014). A Review of Auto-Scaling Techniques for Elastic Applications in Cloud Environments. *Journal of Grid Computing, 12*(4), 559-592.
- McKinney, W. (2010). Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference* (pp. 51-56).

11 Appendices

11.1 DSE MAS Knowledge Applied to the Project

In our project, we applied various skills and knowledge acquired from the DSE MAS course. Key courses that contributed significantly to our project include Python Data Analysis, Database Management Systems, Statistics and Probability with Python, Machine Learning, and Data Visualization.

We used Python extensively for data pre-processing and analysis, leveraging libraries such as Pandas and NumPy to efficiently process and analyze large amounts of immune response data. The knowledge from the Database Management System course enabled us to build and query relational databases, effectively manage and integrate multiple datasets through PostgreSQL and PostgREST API. Statistical methods and probability theory were used to evaluate our models, using metrics such as MAE, MSE, and R-squared values to ensure that the predictions were robust and accurate.

The machine learning techniques we learned in the course were essential in developing the predictive models. We employed various algorithms, including random forests, gradient boosting, and linear regression, and performed hyperparameter tuning to optimize model performance. Data visualization skills were essential in making complex data easier to understand. We created multiple visualization tools and dashboards to clearly present the prediction results and model performance to provide insights to team members and stakeholders.

The integration of these courses allowed us to build and refine predictive models that can predict immune responses to pertussis vaccines, thereby helping to personalize vaccination strategies and improve the overall effectiveness of the vaccine.

11.2 Link to the Library Archive for Reproducibility

https://library.ucsd.edu/dc/collection/bb8093534p