MONASH University

Faculty of Information Technology

FIT3164 Data Science Project II

Assignment 4: Final Project Report

Group Name: MDS02
Tay Qing (32633076)
Tion Yu Xin (33363536)
Matt Dantaradate (32427298)
On Chian Yee (33402302)

Word Count: 10204

Table of Contents

1. Introduction

1.1 Project and its Aims

Cancer remains one of the leading causes of death worldwide, with a rising incidence rate and in Malaysia, cancers such as breast, colon, lung, and head & neck are among the most prevalent, posing significant public health challenges (Prudential, 2023). Despite advances in medical research, the complexity and adaptation of cancer continue to make treatment selection difficult, particularly when it comes to predicting how individual cancerous tumors will respond to specific chemotherapeutic drugs. As such, there is a pressing need for tools that can support better treatment strategies by leveraging biomolecular insights from cancer cells.

Therefore, this project proposes a multi-omics deep-neural-network machine learning model trained on the four types of cancers most common in Malaysia, with a total of 224 distinct cancer cell lines between them, i.e., each cancer type has more than one cancer cell line due to mutations. The model will be trained using various combinations of omics data, with transcriptomics serving as the foundational dataset. Additional omics layers, e.g., proteomics and genomics, will be incrementally incremented to assess their impact on model performance, essentially experimenting with what omics provide the best predictions using the deep neural network model. The model is then integrated into a publicly available website that allows users to upload a dataset containing the drugs they are interested in testing on their specific cancer cell lines. The website will then output the predictions, along with guidance through visualisations and a performance category on which the drugs belong to.

The aim of this project is to reduce the time and effort required by oncologists to identify effective chemotherapy treatments by providing a tool that offers quick, data-driven predictions for drug responses in specific cancer cell lines. By giving clinicians a general indication of which drugs are likely to be more effective, the model supports more informed and timely decision-making in treatment planning. For researchers, the platform also serves as a resource to rapidly assess the predicted performance of various drugs, streamlining early-stage drug screening and hypothesis generation.

Additionally, the project seeks to identify which combinations of omics data are most valuable for improving predictive

performance. By comparing multi-omics models against traditional single-omics approaches, it aims to uncover the most informative biological layers for accurate drug response prediction. Ultimately, this multi-omics deep learning model aims to offer higher prediction accuracy for both clinical and research settings.

1.2 Report and its Content

This report documents the project's 12-week development journey from March 2025 to May 2025, covering the following aspects in summary.

1. **Project Background:** An updated background and literature review from the one made in Project Proposal (Dantaradate et al., 2024).
2. **Outcomes:** Evaluation of the deliverables, 1) deep neural network model with the omics experiment results, and 2) the user-facing website. Limitations and potential improvements for future works will also be included.
3. **Methodology:** An explanation of the final project design with a discussion of its transition from what was described in 2024's Project Proposal, including an outline of the implementation process, e.g., architecture, softwares, tools (Dantaradate et al., 2024).
4. **Software Deliverables:** A description and evaluation of the 2 deliverables described above on how they were developed, from preprocessing to the website deployment, with code explanation where necessary. Evaluations include critical website components and their shortcomings.
5. **Software and Project Critique:** An overview and evaluation of the project's success, in alignment to 2024's Project Proposal (Dantaradate et al., 2024). Reasoning for if and any deviation from the Proposal will be provided.
6. **Conclusion:** Summary of the report's key contents and outcomes.

2. Project Background

2.1 Background Information

2.1.1 Background Material

(1) Omic

The term *"omic"* is the analysis of molecules within a cell, tissue, or organism, and is composed of 5 layers as follows:

- **Genomics:** Study of an organism's complete DNA sequence.
- **Epigenomics:** Study of heritable changes in gene activity that do not involve changes to the DNA sequence.
- **Transcriptomics:** Analysis of RNA transcripts which shows the genes actively expressed in a cell.
- **Proteomics:** Study of all proteins in a cell.
- **Metabolomics:** Analysis of small molecules and metabolites within cells or tissues.

(2) CCLE Transcriptomics Data

The CCLE (Cancer Cell Line Encyclopedia) transcriptomics dataset is a comprehensive dataset that profiles the gene expression of a variety of cancer cell lines provided by Broad Institute, 2019. It provides critical insights into the molecular characteristics of different cancer types, serving as an essential resource for cancer research, particularly in identifying patterns that may contribute to drug resistance or sensitivity.

(3) CCLE Proteomics Data

The CCLE(Cancer Cell Line Encyclopedia) proteomics dataset is an extensive dataset
that profiles the protein expression levels across a wide variety of cancer cell lines, provided by Broad Institute. This dataset offers valuable insights into the molecular makeup of cancer cells at the protein level, enabling researchers to understand how specific proteins may influence cancer behaviour, treatment responses.

(4) LN_IC50

LN_IC50 stands for logarithm of half-maximal inhibitory concentration, and it is the most widely used and informative measure of a drug's efficacy (Aykul et al., 2016, p. 97). An IC50 value indicates the quantity of a particular drug needed to inhibit the growth of a target cancer by half, essentially providing a measure of its potency where the lower quantity of the drug needed is better (Aykul et al., 2016, p. 97).

IC50 values are expressed in logarithmic form as when there are many drugs in the dataset, its values often span several orders of magnitude, hence it is scaled to to standardize the distribution (Aykul et al., 2016, p. 101). In addition, some LN_IC50 values appear negative, as its original IC50 values are less than 1 micromolar (μM), and the natural logarithm of any number between 0 and 1 is negative.

(5) Genomics of Drug Sensitivity in Cancer (GDSC)

One of the essential works in this area is the Human Genome Project, which contains publicly available drug responses to multiple cancer cell lines as shown in figure 1 (GDSC, n.d.). The GDSC dataset was curated through large-scale experimental drug screenings, offering a validated source of information, and serves as the backbone of the DNN training dataset as it provides the LN_IC50 values. (GDSC, n.d.)

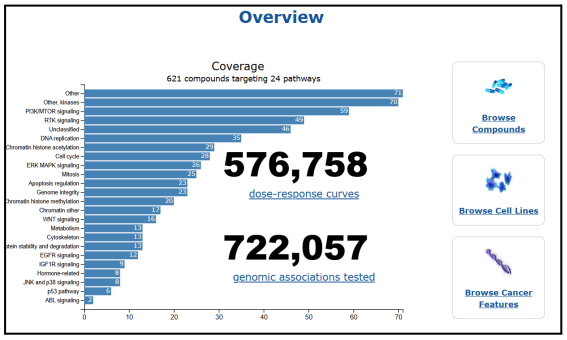

Figure 1. *Current works by the GDSC research (GDSC, n.d.).*

(6) isoSMILES

Isomeric Simplified Molecular Input Line Entry System (isosmiles) is a chemical formula that represents a molecule's structure/formula as a text string, including information about its stereochemistry and isotopes available publicly (https://www.wikidata.org/). Each element is referenced by its symbol on the periodic table, and bonds using characters such as '=' or '-'. In the context of this project, each isosmile value is the molecular makeup of a specific chemotherapeutic drug compound, identified using a unique ID called PubChem (Dantaradate et al., 2025).

(7) Single-omics Machine Learning Model for Predicting Chemoresistance

As shown in figure 2, single-omics approaches use a single type of data, such as genomics, transcriptomics, or proteomics, to build machine learning models for predicting chemoresistance. These models are based on the transcriptomics dataset and use methods such as Support Vector Machines or Random Forests.

(8) Multi-omics Machine Learning Model for Predicting Chemoresistance

Also shown in figure 2 is the multi-omics approach to integrate multiple omic data to build ML models for predicting chemoresistance.

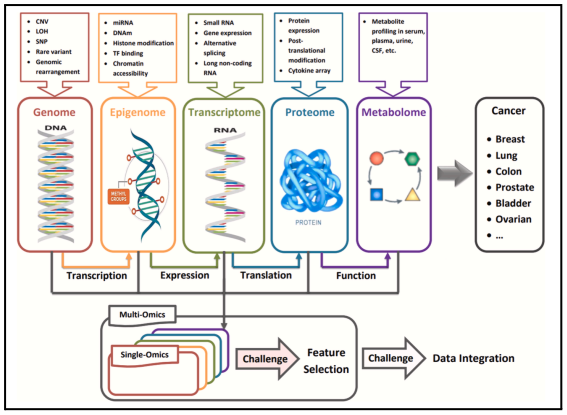

Figure 2: *Single-omics and Multi-omics cancer (Reel et al., 2021)*

2.1.2 Bridging the Gap between Cancer and the Deep Neural Network

As shown in figure 3, the increasing cancer rates, particularly for people under the age of 50 has been significant over the past decade, with cumulative estimates of 79% increases in early-onset cancer, and 29% increase in cancer-related deaths (Cox, 2024). This trend has understandably raised alarm for those in the field of oncology, placing immense pressure on researchers and physicians alike. Not only are professionals in this niche tasked with addressing the rising incidence and improving early cancer detection, but also to explore new treatment strategies for each of the over 1,400 cancer cell lines, develop new chemotherapeutic drugs, and overall conduct extensive experimentations and research to keep pace with the evolving nature of the disease, which stands at over a million mutations (Martell et al., 2013, p. 1271). This is understandably an extremely difficult, manual, and prolonged process, thus creating an opportunity for technological advancements to play a transformative role.

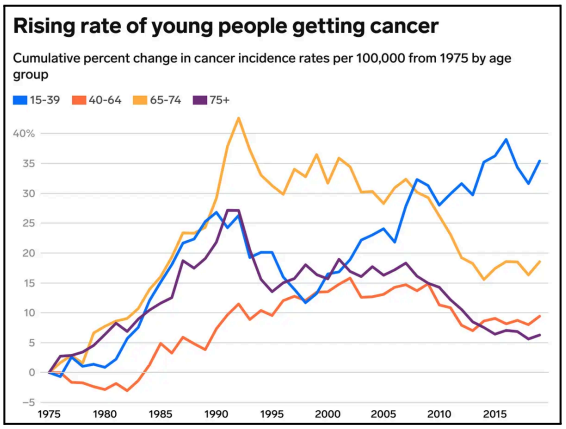

Figure 3. *Cumulative percent change in cancer incidence rate (Hoff, 2024).*

Simultaneously, over the past decade, the machine learning space has made significant strides. These include improvements to foundational models (e.g., regression and decision trees), as well as the development of more advanced ensemble methods (e.g., Random Forests, Gradient Boosting), and deep learning architectures (e.g., neural networks). Thus, when combined with the large-scale efforts biomedical researchers have put into collecting relevant data as described in Section 2.1.1 Background Materials, and in particular the *"omic"* profiles of

thousands of cancer cell lines as referenced by figure 4, these advances present a promising answer to the opportunity in predicting a drug's response to various cancer types.

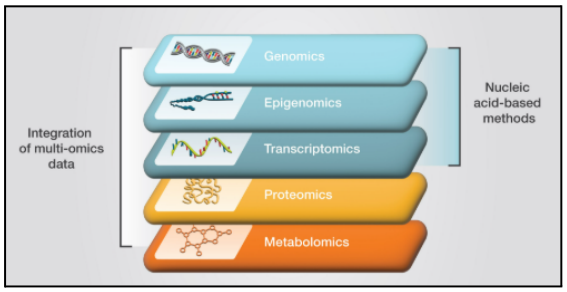

Figure 4. *The 5 layers of omics (Thermo Fisher Scientific, 2021).*

Currently, different machine learning models using a **singular omic** are a popular method of predicting a chemotherapeutic drug response to cancer cell lines (Tam et al, 2024, pg. 5553). The main reason behind this preference is the simplicity and accessibility of single-omics datasets which only requires simple joins and basic data preprocessing to form the model training data (Tam et al, 2024, pg. 5553). However, as previously mentioned, the rapid growth of biomedical data into the 5 omic layers now allows for deeper exploration into integrating additional omic layers into the training dataset. Thus, by incorporating more biological information, the model will receive a richer view of the cancer cells' characteristics, potentially allowing it to uncover more complex patterns and interactions that would otherwise remain undetected in single-omic models.

The chosen model for the project is a deep neural network (DNN) with 2 hidden layers as shown in figure 5. A DNN is a type of artificial neural network (ANN) composed of interconnected neurons structured in layers, where an ANN is classified as a DNN if it has at least 2 hidden layers between the input and output layer. Each layer then transforms its input data through weighted connections and activation functions, where the neurons will learn complex, non-linear relationships with the data, thus providing an output that is the final prediction, the LN_IC50 values, whose meaning will be discussed in Section 2.1.1 Background Materials.

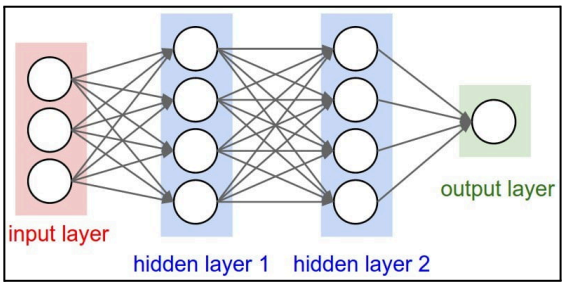

Figure 5. *The Deep Neural Network used (Johnson, 2020).*

The DNN is particularly suitable for this project due to its ability to handle high-dimensional data, i.e., the

multi-omics inputs, by having more than 1 hidden layer. Furthermore, a DNN can automatically learn non-linear relationships between the features without feature engineering as shown in the RTM in Appendix table 7. This aspect is crucial in the context of the project as there is interaction between the omics, i.e., interdependencies between features, and removing a seemingly redundant feature may disrupt the underlying relationships across the omics layer and ultimately reduce model performance.

2.2 Related Works

1. Multi-omic machine learning predictor of breast cancer therapy response (Sammut et al., 2021)

The research paper proposes the usage of transcriptomics and genomics as the training dataset, and then trains 3 models with retraining and uses the best resulting one as the final model as shown in figure 6. The three models are: 1) Logistic regression, 2) Support vector machine, and 3) Random forest. There is feature extraction and cross validation within the flow of the model. The result is an area under curve (AUC) of 0.87.

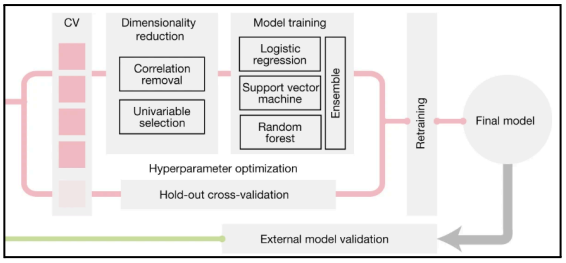

Figure 6. *Model training process (Sammut et al., 2021).*

Updates

From the dataset experiments performed, as shown in Appendix table 6, the combination of transcriptomics and proteomics produces the best prediction results, hence the first update is to replace the genomics dataset with proteomics instead. Secondly, a deep neural network (DNN) is used rather than the ensemble models proposed here. As mentioned previously in the Section 2.1.1 Background Materials, DNN is superior in handling high-dimensional and non-linear data, which is the nature of a multi-omic dataset.

2. Anticancer drug response prediction integrating multi-omics pathway-based difference features and multiple deep learning techniques (Wu et al., 2025).

This research paper introduces the deep learning PASO (which, similar to deep neural networks, is under the umbrella of artificial neural networks). The proposed PASO model is designed to improve individualized prediction of cancer drug

sensitivity using multi-omics data and chemical structure information of drugs, similar to the project's current proposal. However, the final proposed PASO model combines transformer encoders, multi-scale convolutional networks, and attention mechanisms to model the interactions between cancer cell lines and drug molecules, which outperforms recent models in prediction accuracy as shown in figure 7 (Wu et al., 2025).

Model name	RMSE (±sd)	PCC (±sd)	R^2 (±sd)
SVM (Gep, Smi)	1.3632 (±0.0088)	0.8735 (±0.0021)	0.7630 (±0.0036)
Random Forest (Gep, Smi)	1.2171 (±0.0103)	0.9006 (±0.0025)	0.8110 (±0.0044)
LightGBM (Gep, Smi)	1.1953 (±0.0079)	0.9054 (±0.0026)	0.8178 (±0.0032)
XGBoost (Gep, Smi)	1.1611 (±0.0066)	0.9100 (±0.0016)	0.8280 (±0.0029)
Precily (Gep, Smi)	1.1011 (±0.0095)	0.9198 (±0.0016)	0.8311 (±0.0042)
PathDSP (Gep, CNV, Mut, Smi)	1.0499 (±0.0154)	0.9282 (±0.0015)	0.8365 (±0.0116)
PASO-Non-Attention (Gep, Smi)	1.0059 (±0.0450)	0.9333 (±0.0066)	0.8575 (±0.0189)
PASO (Gep, Smi)	0.9882 (±0.0120)	0.9363 (±0.0012)	0.8709 (±0.0025)
PASO (Gep, CNV, Mut, Smi)	**0.9400 (±0.0081)**	**0.9425 (±0.0011)**	**0.8838 (±0.0021)**

Figure 7. *Result of different models (Wu et al., 2025).*

3. Project Outcomes

3.1 Explain what has been implemented.

Summary

The core of the project is a deep neural network (DNN) trained using a multi-omics dataset that produces the best result from the team's experiment using different omics together. This model is then implemented onto a website where users can create their own dataset of the drugs they would like to test against cancer cell lines of their choosing. Each aspect will be discussed.

Pre-processing datasets for experiments

Essentially, the team produced 5 different datasets to decide which of them naturally produces the best prediction score from training and testing with the DNN model (more on this below). Pre-processing is done according to figure 8, and will be discussed more in Section 4 Methodology. Thus, the dataset combinations in the experiment are:

(1) Transcriptomics – GDSC 2

(2) Transcriptomics – GDSC 2 – isoSMILES **(the control dataset)**

(3) Transcriptomics – Proteomics – GDSC 2 – isoSMILES

(4) Transcriptomics – Genomics – GDSC 2 – isoSMILES

(5) Transcriptomics – Proteomics – Genomics – GDSC 2 – isoSMILES

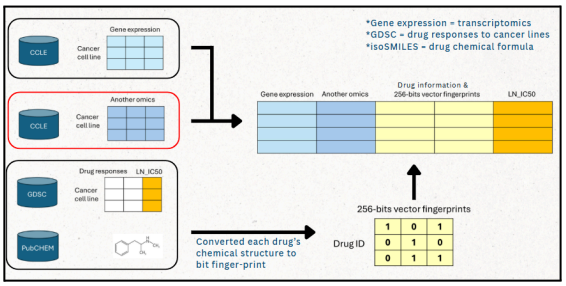

Figure 8*: Training data pre-processing steps*

Choosing the best configurations

As shown in figure 9, the DNN chosen by the team consists of 2 hidden input layers to balance prediction performance/complexity handling and time constraints. Within each hidden layer, there are 2 configurations to choose, 1) the number of neurons, and 2) the activation function as shown in figure 10. From this, the best configurations used by each dataset in the experiments were discovered by looping through a list of no. of neurons and activation functions. A patience value (for the whole model training) for each dataset was also found through a loop.

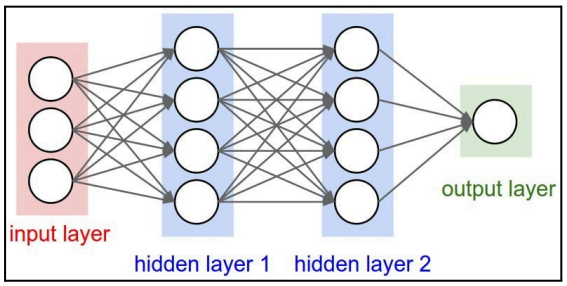

***Figure 9**: The DNN model used*

```
# Hyperparameter tuning
# Define configurations for neurons and activation functions for each layer
neurons_list_layer1     = [64, 128, 256]
activation_list_layer1  = ['relu', 'tanh', 'sigmoid']
neurons_list_layer2     = [32, 64, 128]
activation_list_layer2  = ['relu', 'tanh', 'sigmoid']
patience_list = [10, 15, 20]  # try different patience values
best_patience = None
```

***Figure 10**: The list of configurations*

Website architecture

Once the dataset combination that produces the best prediction result has been found as shown in Appendix table 6. The model was packaged into a website following the architecture shown in figure 11, and is discussed more in Section 4 Methodology. The website is then deployed, which will be discussed in higher details in Section 3.2 Present the Results.

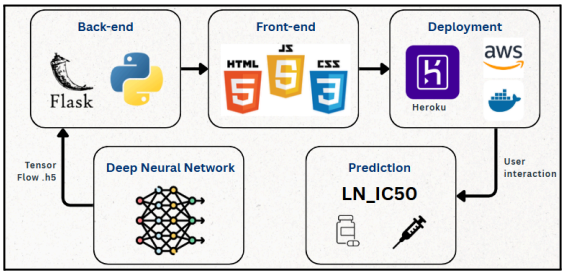

***Figure 11**: Website architecture*

3.2 Present the results, achievements and/or the delivered product.

Experiment results using the best configurations

As mentioned previously, the experiment results using each dataset's best DNN configurations are shown in Appendix table 6. Interpreting the result table shows that the multi-omics dataset consisting of transcriptomics and proteomics in number 3 produces the lowest RMSE and highest R-square values, higher than that of the single-omics in number 2. Also shown in the results is that a usage of genomics dataset reduces the prediction accuracy of the model.

The configuration used to train the following DNN model is

(1) **Hidden layer 1:** 256 neurons with activation function 'relu'
(2) **Hidden layer 2:** 32 neurons with activation function 'relu'
(3) **Patience value** of 20

The Website

The website encapsulates the model into a prediction page, and is available through this link. As a summary, the website's prediction page performs 2 validation checks on user data input.

(1) CSV format requirements: The dataset should have the specified columns with exact matching names in order to be uploaded. This check is a manual process where users have to tick the checkboxes as shown in figure 12.
(2) Backend validation: On the backend, the website performs validation on the input data in checking the columns & their names (the 2nd time) and checks for missing values (NA).

Subsequently, the backend processing will convert the user input data's isoSMILE column from string to a 256-bit morgan finger value to represent the drug's chemical compound as a binary value. As shown in the experiment result from the Appendix table 6, doing so boosts the prediction performance significantly, i.e., comparing row 1 to 2.

***Figure 12**: Website architecture*

Once the user has uploaded a dataset, there are several components within the website that make it extremely intuitive. The core downloadable prediction results are displayed into a table as shown in figure 13, which has two additional columns. 1) Predicted LN IC50 values, and 2) Sensitivity. Sensitivity is a classification of the drug into low, intermediate, and high, of which higher sensitivity means lower LN IC50 value, i.e., less quantity of the drug needed to cut cancer growth by 50%. The intuitive parts of the website prediction results are listed below.

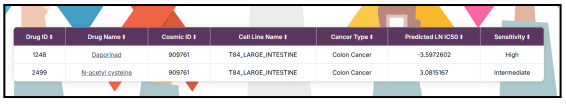

***Figure 13**: Prediction results table*

(1) Filter by cancer types

As shown in figure 14, users can filter the prediction results based on the cancer types within the uploaded dataset. This is extremely useful for users who upload a large dataset of the 4 cancer types.

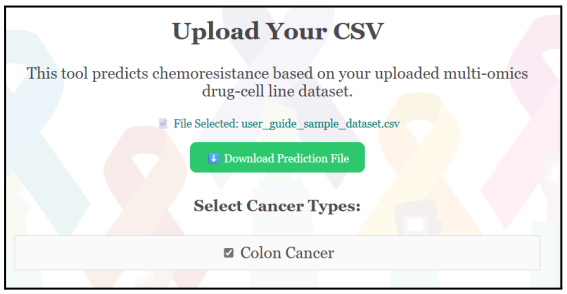

***Figure 14**: Top page immediately after a prediction is created*

(2) Visualisations

As shown in figure 15, the outputted LN IC50 values for each drug belonging to each cancer type is visualized into a bar graph for easier interpretation. As each cancer type will have its own visualisation window, this makes the page extremely clean and will further help users organise their thoughts.

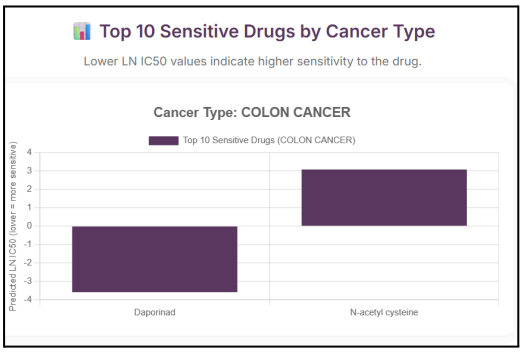

***Figure 15:** Visualisations of the LN_IC50*

(3) Wikipedia link for each drug

In the predicted results table, each drug the user has selected has a wikipedia link to show more information regarding it, as shown in figure 16.

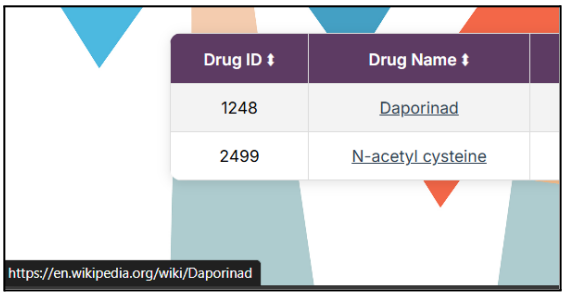

***Figure 16**: Wikipedia link of each drug*

3.3 Explain how project requirements are met by your deliverables

To guarantee that our chemoresistance prediction tool met its intended goals, we defined a detailed set of functional and non-functional requirements during the early planning stage. These requirements guided our development decisions and provided a framework to evaluate the completeness and quality of the final system.

The functional requirements (FR) covered tasks such as comics data collection (R-001), dataset preparation (R-002), training the predictive model (R-004), and delivering predictions through a working prototype (R-008). Meanwhile,

non-functional requirements (NFR) focused on broader system attributes, such as selecting reliable data sources (R-001) and ensuring fast model response times (R-009). Each requirement was tracked using a Requirements Traceability Matrix (RTM), as shown in Appendix Table 8.

All listed requirements were successfully fulfilled, with the exception of R-003 (Feature Selection), which was removed after consultation with the supervisor on the deep neural network. As discussed in Section 2.1.2 Bridging the gap between cancer and DNN, the decision was based on the fact that the deep learning model used (a neural network) inherently performs automatic feature weighting, reducing the need for manual feature selection.

3.4 Justify decisions made.

Two key decisions shaped the direction of our project:

a. Selecting Deep Neural Networks (DNNs) as the core predictive model and,
b. Choosing to use only transcriptomics and proteomics data as input features.

We chose DNNs over traditional machine learning algorithms because of their superior ability to learn complex, non linear relationships across high dimensional biological datasets/ Given the size and nature of our omics data, over 19,000 transcriptomic and 200+ proteomic features, a DNN was more appropriate for capturing multi layered biological interactions. Unlike tree based models or linear regression, DNNs can learn hidden feature hierarchies, enabling better generalization and prediction accuracy in high dimensional spaces. Furthermore, DNNs inherently perform feature weighting and representation learning, which justified our decision to exclude separate feature selection (R-003), as it became redundant and risked discarding informative features prematurely.

The decision to limit our input to transcriptomics and proteomics, excluding genomics, was based on empirical results. Although our initial plan included genomics as part of a multi-omics strategy, extensive testing revealed that including genomics led to lower predictive performance (i.e., reduced R^2 scores and increased RMSE). Transcriptomic and proteomic data showed stronger correlation with chemoresistance outcomes and provided a better signal to noise ratio. Removing genomics not only improved model accuracy but also reduced computational complexity and

preprocessing burden. This decision was validated through comparative evaluation and supervisor feedback, aligning with the project's focus on precision, usability, and result reliability.

3.5 Discuss your project results

Our project presents a significant advancement in the domain of cancer drug response prediction by developing a multi-omics based deep learning system accessible through a fully functional web interface. Compared to existing models in the field, our deliverables offer improved predictive accuracy, greater biological interpretability, and practical usability, especially within the context of translational research where speed, reliability and input complexity matter.

3.5.1 Comparison with Previous Results

Unlike many previous approaches that rely solely on single-omics data and particularly transcriptomics, our system integrates multi-omics inputs, including transcriptomics, proteomics and 256-bit drug fingerprints, as mentioned in Section 3.2 Present the results. Referring to Appendix table 6, from the experiment result we observed that transcriptomics alone produces weak predictive performance. In contrast, the inclusion of proteomics significantly improved model performance, with our best model achieving an R square of 0.804 and RMSE of 1.22. Additionally, the inclusion of genomics did not result in better outcomes, confirming that the combination of transcriptomics and proteomics is currently the most effective setup for our application. This performance gain highlights the ability of multi-omics models to better represent the biological complexity underlying chemoresistance.

3.5.2 Fit with Research and Domain Needs

As biomedical research shifts toward generating increasingly complex datasets, there is a growing need for machine learning tools capable of handling multi-layered omics inputs. Our model directly supports this direction by offering an end to end workflow for predicting drug sensitivity using 2 biological (omics) and chemical inputs (isoSMILES). The platform encourages researchers to continue development of omic dataset, which in turns motivates a deeper investigation into omics patterns and their relevance to drug performance, something few existing tools emphasize.

By designing the platform to accept user submitted data and return real time predictions and visual insights, we also support day to day research needs in

academic and experimental settings. This makes our tool relevant not only from a performance standpoint but also in terms of usability and integration into ongoing workflows.

3.5.3 Extension of the State of the Art

Our project extends the current landscape in two major ways. First, it operationalizes a multi-omics deep neural network into a real time web application, something that research prototypes do not do. Second, it demonstrates the value of integrating proteomics into prediction pipelines, which may encourage more future work in this direction. Through its high prediction accuracy, flexible input structure and interactive output, our tool offers a new benchmark for how multi-omics can be built and delivered effectively.

3.6 Limitations of Project Outcomes

Although we have achieved our main objective of providing a web-based multi-omics chemoresistance predictor, a number of limitations restrict its present level of scientific research and industrial readiness.

3.6.1 Biological and Data Coverage

Even though the model makes use of two of the best pharmacogenomic resources (CCLE and GDSC), there are only 224 immortalised cell lines in its training corpus, which represent four solid tumours: breast, colorectal, lung, and head and neck. The molecular markers of other high-incidence malignancies in Malaysia, such as liver, prostate, nasopharynx and leukaemia, which collectively contribute to a significant portion of the country's cancer burden, are thus not exposed to it *(Global Cancer Observatory - Malaysia)*. The decision boundaries of the network are therefore customised to a relatively limited portion of cancer biology causing the danger of generating skewed, overconfident predictions whenever input data comes from these under-represented illnesses.

At the molecular level, coverage is further limited because the existing pipeline only incorporates transcriptomic and proteomic layers, leaving out metabolomic fingerprints, miRNA profiles, epigenetic marks, and genomic mutations, all of which are known to contribute to drug resistance mechanisms. Lastly, depending solely on two-dimensional, monoculture cell-line data ignores the tumour microenvironment, which limits ecological validity due to the absence of stromal signalling, immunological infiltration, and hypoxia cues. Tumour diversity, omics breadth, and microenvironment realism are

three stacking omissions that collectively explain why the current prototype should only be considered a proof-of-concept and strongly encourage future expansion to bigger, patient-derived multi-omics datasets.

3.6.2 Verification & Testing Gaps

Activities for ensuring quality are still limited. Evaluation of the model is based on a 20% random split from the same four-cancer dataset; generalisability is uncertain because no external benchmarking against independent cohorts has been undertaken as of yet. Toy examples with one cell line and two medications were utilised in website integration tests; these workloads were orders of magnitude smaller than the 600-row workloads anticipated in real-world scenarios. In the absence of rigorous load or stress testing, it is impossible to quantify request delay, memory utilisation, and throughput under concurrent demand. Maintenance risk is increased by the lack of automated unit, integration, and regression suites since upcoming code changes need to be manually reviewed.

3.6.3 Security, Privacy & Regulatory Concerns

Security measures are low. Even though HTTPS encrypts data while it is in transit, uploaded files are not automatically purged and are retained in an unencrypted state at the public endpoint, which does not have authentication. If real patient datasets are uploaded, this architecture, in conjunction with the lack of per-session isolation, exposes possible genomic-privacy violations. Regulation-wise, the instrument was not created using a certified quality-management system, such as ISO 13485 (*ISO 13485:2016 Medical Devices — Quality Management Systems — Requirements for Regulatory Purposes.*, 2016) and does not have ethics certification to process human genomic data. The prototype cannot currently assist in clinical decision-making and must be carefully categorised as a research demonstration.

3.7 Potential Improvements and Future Work

Building on the limitations identified above, three key enhancement areas will guide the next phase of development.

3.7.1 Expanded Omics Coverage and Cancer Spectrum

Apart from transcriptomics and proteomics, the model will incorporate other molecular layers like metabolomics (e.g., metabolite profiles produced from

LC-MS) and epigenomics (e.g., DNA methylation arrays) to improve biological fidelity and predictive capacity. The complementing features of cancer biology that are captured by both data types include chromatin-state changes and metabolic rewiring, which are known to influence chemoresistance. Additionally, by obtaining patient-derived xenograft or organoid models whenever feasible, the training set will be expanded to encompass new cancer subtypes that are common in Malaysia, such as liver, prostate, nasopharynx, and leukaemia. The next-generation pipeline will create richer feature representations by combining multi-layered "omics" with increased histological variety. This will allow for more reliable generalisation to uncommon and location-specific cancers.

3.7.2 DNN Architecture Enhancement

Although the existing two-hidden-layer neural network works well for preliminary proofs-of-concept, it can be enhanced by adding regularisation methods such batch normalisation and dropout, as well as by increasing depth (e.g., 4–6 hidden layers). By decreasing internal covariate shift, batch normalisation will stabilise and speed up training, and dropout will lessen overfitting in high-dimensional omics domains (Srivastava et al., 2014). In order to capture complex cross-omics interactions, alternative architectures such residual connections or autoencoder-based pretraining may also be investigated. Finding the best learning rates, layer widths, activation functions, and dropout rates can be aided by hyperparameter tuning with automated tools (such as Optuna or Keras Tuner). The goal of these improvements is to give more steady convergence on heterogeneous datasets and drive model accuracy towards clinical thresholds (RMSE < 0.5, $R^2 > 0.90$).

3.7.3 Authentication & Data Persistence

From a demonstration prototype to a production-ready system, a production-ready platform will incorporate a reliable backend database such as PostgreSQL and a secure user-login mechanism. This will guarantee that patient-derived omics can only be submitted for analysis by those who are authorised. Longitudinal research and audits of regulatory compliance will be made possible by persistent user profiles, upload histories, and prediction outcomes. Cohort-level analysis and quick retrieval will be made easier by database indexes based on patient ID, sample date, and treatment type.

3.8 Outcome-Related Issues of Relevance and Interest

In deploying a chemoresistance-prediction tool, several outcome-related challenges arise that span ethical, clinical-integration, and technological domains.

3.8.1 Ethical and Clinical Implications of Predictive Automation

Ethical concerns regarding how patients and practitioners perceive and respond to model results are brought up by the introduction of an automated chemoresistance predictor. An excessive dependence on algorithmic forecasts runs the risk of impairing patient autonomy and medical judgement, even though quick, data-driven recommendations can enhance therapy selection. Suboptimal therapy or postponed interventions may result from false-positive or -negative medication response forecasts. A human-in-the-loop structure is advised to reduce these risks, and model outputs should be accompanied by understandable confidence intervals and explainability reports (such as SHAP scores) that put the biological justification in context. Additionally, clinicians must be trained on the limitations of predictions made in vitro and should only utilise the tool as a supplement to established diagnostic procedures rather than as a final arbitrator of decisions.Patient rights will be further protected and responsible usage will be ensured by ethical oversight through institutional review boards (IRBs) and open patient permission procedures for using molecular data in algorithmic pipelines.

3.8.2 Integration with Clinical Workflows and Hospital Systems

A web-based chemoresistance predictor must be seamlessly integrated with current electronic health records (EHR) and hospital information systems (HIS) in order to be implemented into standard clinical practice. Clinicians must manually download and reformat CSVs in the absence of interoperability, which is a laborious and error-prone process that hinders uptake. In order to streamline data interchange between the prediction platform and EHR modules, future development will concentrate on building APIs that are compliant with FHIR (Fast Healthcare Interoperability Resources). Additionally, drug-response predictions can be displayed at the time of prescription by integrating the technology into clinician dashboards (for example, through SMART on FHIR apps). Laboratory data pipelines must also be taken into consideration. To guarantee that entering omics inputs meet the model's expected schema, pre-analytical procedures including sample processing, sequencing, and quality

control must be standardised. To accomplish high-fidelity, real-time integration, it will be crucial to involve IT stakeholders early on in order to converge on data governance, role-based access control, and audit logging.

3.8.3 Adaptability to Emerging Technologies and Data Modalities

Our existing architecture needs to be adaptable and future-proofed due to the rapid advancements in machine learning frameworks and multi-omics technologies. Multi-scale feature fusion and heterogeneous data modalities should be supported by the prediction pipeline as single-cell RNA-seq, spatial transcriptomics, and digital pathology become more widely available. Similarly, the platform should support plug-in modules instead of monolithic retraining as new deep-learning methods are developed, including graph neural networks that capture gene–protein interaction networks. Clear interface design between the components for data pretreatment, model training, and inference will enable small-scale improvements without requiring extensive code reworking. The platform will continue to stay at the forefront of computational efficiency and biological insight through regular code audits, benchmarking exercises, and community feedback loops with bioinformatics consortia.

4. Methodology

4.1 Final Project Design

Our final product is a publicly accessible web-based platform titled Cancer Drug Response Predictor, designed to forecast cancer cell line resistance to chemotherapy drugs based on multi-omics data. Developed using the Flask web framework, the system enables users to upload their omics datasets and receive chemoresistance predictions and biomarker insights.

The model is trained specifically on four major cancer types prevalent in Malaysia: Breast, Lung, Colorectal and Head and Neck cancers. For each cell line, input data includes transcriptomics and proteomics.

The web application is composed of three main tabs:

1) **Introduction Page -** Overview of the tool's purpose and targeted cancers

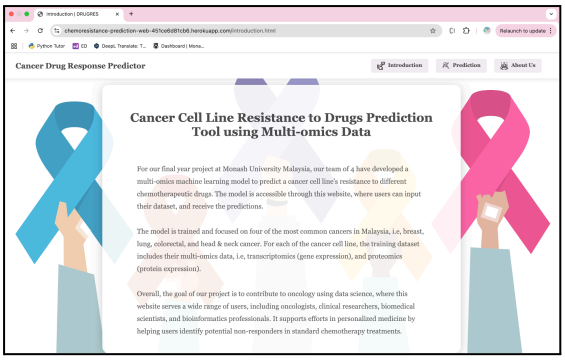

Figure 17 : Introduction Page

2) **Prediction Page -** CSV upload for generating LN_IC50 predictions

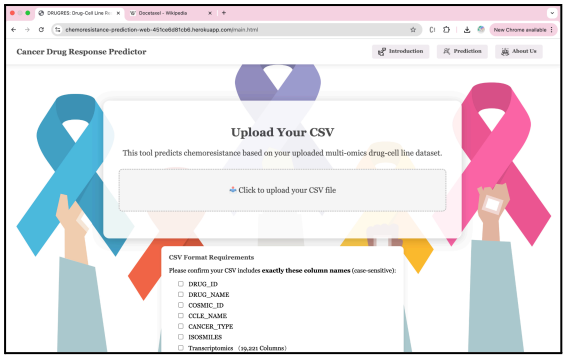

Figure 18 : Prediction Page

3) **About us -** Team member profiles and project background

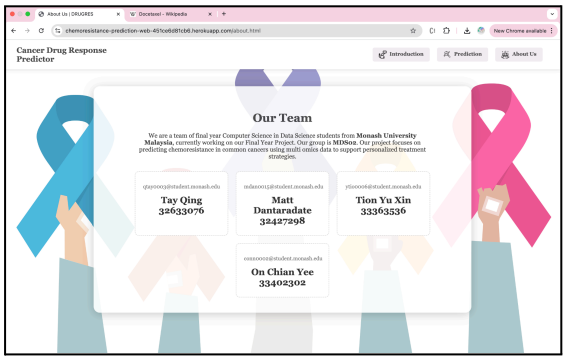

Figure 19: About Us Page

The interface is designed for ease of use by both researchers and clinician, with minimal user-side setup.

4.2 Deviations from Initial Design

The final implementation of our platform closely followed the external design outlined in the initial design proposed in FIT3163 but underwent several refinements to enhance clarity, responsiveness, and usability. Our original design emphasized a clean, minimalist interface focused on user guidance for uploading omics data and receiving chemoresistance predictions. While these core principles were retained, key visual and functional elements evolved based on testing feedback and practical integration with the backend model.

(1) One of the most notable deviations is the visual styling of the interface. While the initial version featured a soft blue background with microscope inspired imagery and a floating file selecting card, the final design returned to a cleaner white card layout over a themed background of multi colored cancer awareness ribbons. This visual branding added contextual meaning while maintaining a neutral aesthetic that works across clinical, academic and public audiences. The

buttons, such as “Select File” and “Download Sample CSV”, were also styled more clearly, ensuring that users can easily identify the next step in the interaction process.

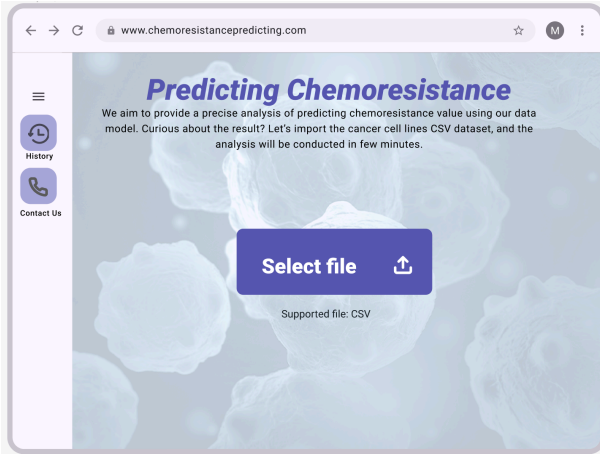

Figure 20: Initial Website Design

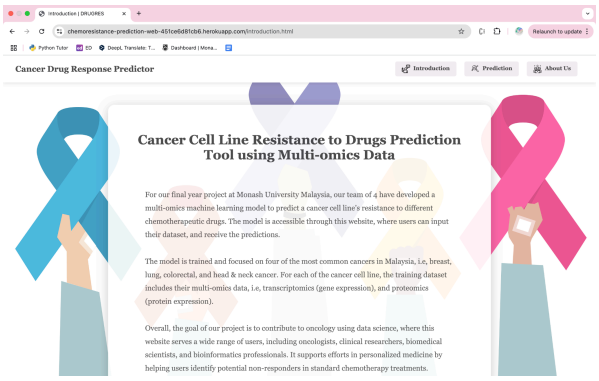

Figure 21: Final Website Design

(2) In terms of navigation, the original sidebar design with icons for “History” and “Contact Us” was replaced with a top navigation bar featuring labeled tabs: Introduction, Prediction and About Us. This deviation reflects a design choice to simplify navigation for desktop and mobile users alike. The top menu better aligns with standard web practices, reducing cognitive friction while exploring the site. At the same time, the “History” and “Contact Us” functionality initially envisioned as core was deprioritized or moved to supporting sections in the final version due to scope constraints and practicality.

(3) The structure of the upload workflow remained mostly intact, including CSV only validation, error pop ups and progress feedback as shown in Figure 22. However, as shown in Figure 23, the final system provides a more detailed CSV format checklist, including drug IDs, cancer types and specific transcriptomics and proteomics column expectations. This enhancement reflects real implementation needs where input standardization was critical to ensure the machine learning model could process data reliably. Users are guided to match these formatting rules explicitly before uploading, helping prevent errors and confusion. In the final design, there was also a sample CSV that let users download and follow the format.

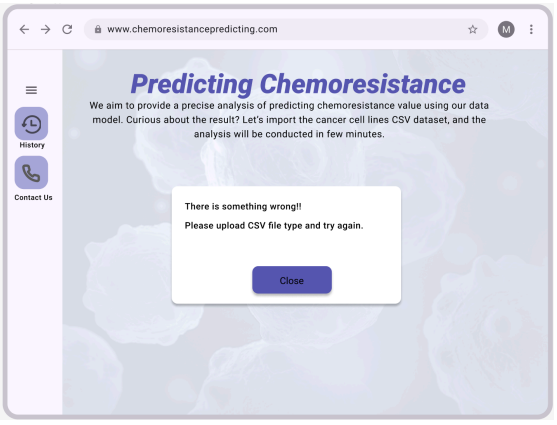

Figure 22: Initial Website Design

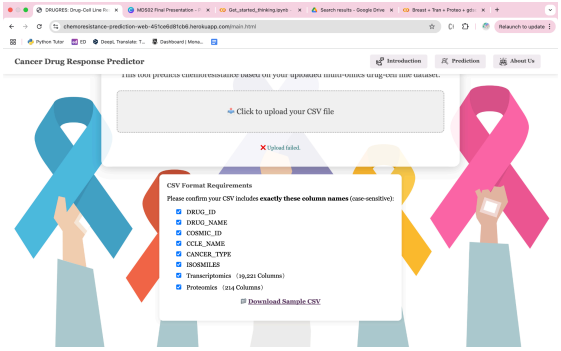

Figure 23: Final Website Design

(4) Additionally, the final design introduced a separate Introduction page explaining the project's scientific context and an About Us section introducing the team, features not fully outlined in the initial mockup. These additions serve both academic and user trust building purposes, particularly for stakeholders in the biomedical research community.

(5) Compared to the initial design, which featured a simple "Analysis Report Download" button as the final step after file upload, the completed version introduces a more interactive and multi step output process. After users upload their files, users are not only given a prediction file download, but also presented with options to filter results by cancer type, visualize the top 10 most sensitive drugs using bar charts, and view a sortable result table with detailed drug response data. These enhancements provide users with greater control, clarity and insight, allowing them to tailor the output to their research needs rather than receiving a fixed, prepackaged report. The inclusion of these dynamic elements marks a clear departure from the initially envisioned static workflow and reflects a deeper understanding of how biomedical users interact with predictive data platforms.

4.3 How the Design Was Implemented

The development of the Cancer Drug Response Predictor followed a full-stack machine learning engineering pipeline, combining multi-omics data science, deep learning, and web deployment techniques. Our goal was to build a system that not only produced accurate chemoresistance predictions but could also be easily accessed by researchers and clinicians through an intuitive web interface. The implementation can be broken into five

reproducible stages: data collection, data preprocessing, model training, website development and deployment.

4.3.1 Data Collection

Data Collection process brought together multiple publicly available biomedical resources to construct a multi dimensional dataset suitable for chemoresistance modeling. Three main sources were utilised:

1. Cancer Cell Line Encyclopedia(CCLE) provided both **transcriptomics**(gene expression), proteomics and genomics data for hundreds of human cancer cell lines. These omics layers form the biological profile of each cell line.

2. Genomics of Drug Sensitivity in Cancer(GDSC2) supplied the **drug response values**(in the natural log of IC50, LN_IC50) for these cell lines across a wide range of chemotherapeutic agents. This served as the ground truth label for regression modeling.

3. PubChem offered the **isoSMILES** strings representing the chemical structure of each drug. These were converted into 256 bit binary molecular fingerprints using RDKit, enabling chemical feature inclusion.

To manage the constraints of limited development time and computational resources, we forced our dataset on four major cancer types: breast, lung, colorectal and head and neck cancers. These types were chosen due to all these cancers being the most common in Malaysia.

4.3.2 Data Preprocessing

As shown in Figure 24: Training Data Preprocessing, a structured preprocessing pipeline was implemented to convert raw multi-omics and chemical data into a format suitable for training our deep learning model. All operations were performed using Python, leveraging libraries such as pandas, NumPy, scikit-learn and RDKit-pypi.

Transcriptomics, proteomics and genomics datasets retrieved from CCLE were first aligned by cancer cell lines and then normalized to standardize feature distributions. Drug chemical structures, represented by isoSMILES strings from PubChem, were transformed into 256-bit molecular fingerprints, effectively encoding their structural characteristics as numerical vectors.

These processed layers were merged with LN_IC50 drug response values from GDSC2 to form a consolidated feature matrix. Each record in the final matrix

represents a specific pairing of a cancer cell line and a chemotherapeutic compound, containing combined biological and chemical descriptors alongside the response variable.

To determine the most predictive feature configuration, we created five distinct dataset combinations as part of our preprocessing experiments:

- Transcriptomics + GDSC2
- Transcriptomics + GDSC2 + isoSMILES (control set)
- Transcriptomics + Proteomics + GDSC2 + isoSMILES
- Transcriptomics + Genomics + GDSC2 + isoSMILES
- Transcriptomics + Proteomics + Genomics + GDSC2 + isoSMILES

These combinations were designed to test how the inclusion of various omics layers affects model accuracy. Each dataset followed the same preprocessing logic and was evaluated under the same conditions. After preparation, all datasets were split into an 80:20 ratio for training and testing, forming the basis for model selection described in the following section.

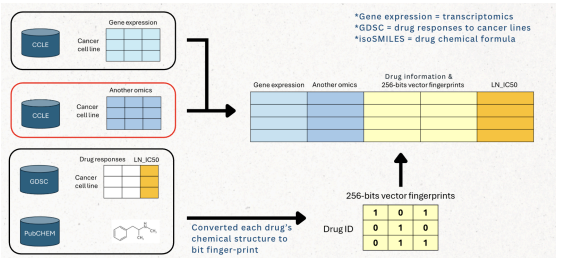

Figure 24: Training Data Preprocessing

4.3.3 Model Training

To predict drug chemoresistance values (LN_IC50), we trained a Deep Neural Network(DNN) using TensorFlow and Keras. The model takes as input a combination of transcriptomics, other omics and drug chemical structure(encoded as 256-bit fingerprints). The architecture consists of one input layer, two hidden layers and one output layer. The DNN was chosen over traditional models like Random Forest due to its superior capacity to model high dimensional, nonlinear interactions between multi-omics and chemical features - characteristics inherent to biological systems.

The training process involved testing various configurations of neuron counts and activation functions through iterative grid search. To select the best combination, we first experimented using breast cancer only across five omics configurations. The combination of transcriptomics + proteomics +

isoSMILES achieved the best results and was then used to retrain the model on all four cancers.

As presented in table 6, our best model achieved an RMSE of 1.22 and R squared of 0.804, outperforming all other configurations, including those using only transcriptomics or including genomics. In contrast, single omics transcriptomics without isoSMILES models (Dataset 1) performed poorly with an R square of only 0.029, highlighting the value of integrating isoSMILES for more accurate prediction.

This model was saved in .h5 format and loaded in the backend for real time inference on user uploaded datasets. The integration of this high performance DNN into a live web service allows users to perform personalized drug sensitivity prediction directly from their browser.

4.3.4 Web Development

The frontend of the Cancer Drug Response Predictor tool was developed using standard web technologies HTML, CSS and JavaScript with a focus on delivering a clean, professional and responsive user interface. The design was structured around three main pages: Introduction, Prediction, and About Us, accessible through a top navigation bar. The interface was optimized for clarity and usability, particularly for biomedical researchers and clinicians with minimal technical backgrounds. The Prediction page supports file uploads, enforces format validation using checkboxes, and provides clear feedback to users through modal popups whenever an error occurs. After a successful upload, the server returns predictions along with a cancer type filtering system, a download button, bar charts of the top 10 most sensitive drugs (generated using Matplotlib), and a detailed results table rendered in the UI. Each element is tailored to support biomedical researchers by enabling quick and interpretable exploration of the prediction outputs.

4.3.5 Web Deployment

Docker containers are used to deploy our full website in order to ease roll-outs and guarantee environment consistency. All dependencies from ***requirements.txt***, such as Flask, TensorFlow, and RDKit, are installed via a ***Dockerfile*** based on Python 3.11, which also copies in the model artefacts and application code and sets up Gunicorn as the WSGI server. A Heroku Dyno automatically maps port 5000 to the public URL after the image has been tagged and submitted to Heroku's Container Registry. Heroku environment variables are used to inject runtime

configuration, such as S3 bucket URLs for CCLE/GDSC data, in order to prevent secrets from being under source control. New CSVs are processed in an ephemeral ***/tmp*** directory under Flask, and the dyno downloads any missing model files from S3 during startup. Before streaming the results back to the browser, Heroku's routing layer distributes requests among dynos, each of which is running a Gunicorn worker that calls the ***preprocessing_model.py*** and ***predict_chemoresistance.py*** modules. The deployment process is straightforward: Build and smoke-test the Docker image locally, then use ***heroku container:push web*** to push it to Heroku, execute ***heroku container:release web***, then use ***heroku logs --tail*** to confirm. This method ensures low operational overhead, quick rollbacks to earlier picture tags when necessary, and parity between development and production.

5. Software Deliverables

5.1 Summary of Software Deliverables

5.1.1 Brief Description

Throughout the project, our team has successfully produced the following deliverables:

Web Application

- Users can upload multi-omics CSV files for chemoresistance prediction using this fully working Flask-built online platform.
- Integrated validation alerts and error-handling: a detailed Bootstrap "alert" identifying the precise fields that need to be fixed displays if the CSV is distorted (for example, missing necessary columns).
- Bundled as a Docker container with Flask, TensorFlow, RDKit, and Python 3.11, and then deployed to Heroku through the Container Registry. The application uses Gunicorn in production to process HTTP requests with great performance.

Software & Collaboration Tools	Usage
Visual Studio Code, Google Drive, Google Docs, Google Sheets, Google Colab, Google Spaces, Canva, Lucidchart, Zoom, WhatsApp, Heroku CLI, Docker Desktop	- Debugging and code development - Managing branches and version control - Draughting of reports and slides and documentation - Prototyping and testing models - Team meetings and communication - Tracking tasks and organising sprints - Coordination of manual deployment

Table 1: Software & Collaboration Tools Resources

Hardware & Cloud Infrastructure	Usage
Intel Core, M1 Chip, Personal Laptops, Heroku Dyno, Local Docker Environment	- Local development and unit testing - Large-scale data preprocessing and model training - Public hosting of the Dockerised web app

Table 2: Hardware & Cloud Infrastructure Resources

Python Libraries & Frameworks	Usage
Flask (3.1.0), Flask-CORS (5.0.1), Gunicorn (20.1.0), TensorFlow (2.19.0), scikit-learn (1.2.2), pandas (2.2.3), NumPy (1.26.2), RDKit-pypi (2022.9.5), Joblib (1.5.0), requests (2.32.3), Jinja2 (3.1.6)	- Web server framework and request handling - CORS support for future integrations - WSGI deployment under Gunicorn - Deep-learning model implementation and inference - Data manipulation - Chemical descriptor processing - Serialisation of scalers and models - HTTP requests for S3 downloads

	- HTML template rendering - Interactive chart generation

Table 3: Python Libraries & Frameworks Resources

Operating System	Usage
Windows 11, macOS	- Primary development environments (local machines) - Consistency between local and cloud deployments

Table 4: Operating System Resources

Personnel & Roles	Usage
Supervisor: Dr. Ong Huey Fang Project Manager: Matt Dantaradate Technical Lead: Tion Yu	**- Supervisor:** Domain guidance and report review **- Project Manager:** Sprint coordination and team meetings **- Technical Leads:**
Xin & On Chian Yee Quality Assurance: Tay Qing	Architecture design, model integration, and deployment **- Quality Assurance:** Usability testing and documentation proofreading

Table 5: Personnel & Roles Resources

Deep Learning Model

- Final two-layer DNN weights, optimised on CCLE/GDSC breast, colon, lung, and head-and-neck cell-line data, are contained in a pre-trained Keras HDF5 file.

- A serialised scikit-learn StandardScaler used to normalise incoming omics features prior to inference.

- Model inference code in ***predict_chemoresistance.py*** that loads both the ***.h5*** and ***.pkl*** artifacts, aligns feature columns, applies scaling, and returns predicted log(IC_{50}) scores for each drug-cell-line combination.

Source Code Repository & Documentation

- As additions and fixes were added, incremental updates were committed and pushed using GitHub's distributed version control system.
- All code and resources were versioned, and the repository was logically arranged into directories containing HTML templates, CSS/JavaScript assets, backend Python scripts, and model artefacts.
- Supervised feature forks to work on model adjustments, frontend improvements, or preprocessing modifications prior to merging into the main branch.
- To identify particular commit points that correlate to Docker image builds and Heroku deployments, stable releases were tagged in GitHub..
- To guarantee consistent manual build and push procedures, deployment configuration files (Dockerfile and Heroku.yml) were stored with the source code.

5.1.2 Brief Visual Overview of The Software

Our web application can be accessed through a public URL due to its deployment on Heroku, but users can also choose to run it locally on their own computers. In either case, when visitors arrive, they are taken to the introduction page, which is the home page. The goal of the tool is explained on this landing page, which also walks users through the process of uploading their multi-omics CSV files for chemoresistance prediction.

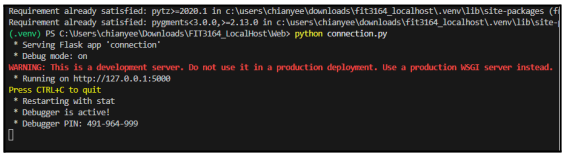

Figure 25: VS Code Terminal Connecting to Local Host Website

The components and functionalities for the Introduction Page (see figure 26) are outlined below.

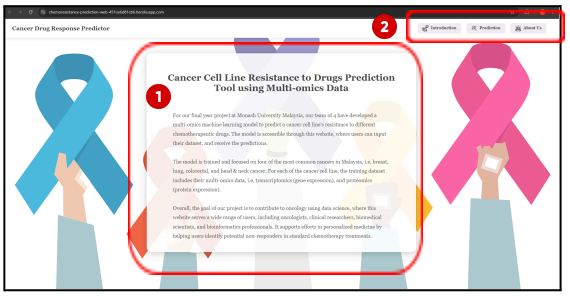

Figure 26: Introduction Page Interface

1. Introduction Frame: Users are presented with a thorough overview of our final-year project as soon as they access the website, whether through the public Heroku URL or by running it locally. The development of a multi-omics machine learning model to predict cancer cell-line resistance to

chemotherapy is described in this section.

2. Navigation Menu: In the top-right corner of the page, three navigation buttons allow users to explore different parts of the site:
    - **Introduction:** Highlights the current page, providing an overview of the project's aims, data sources, and intended audience (Figure 26).
    - **Prediction:** Enables users to evaluate chemoresistance forecasts, initiate backend processing, and upload their multi-omics CSV file (Figure 27).
    - **About Us:** Takes users to a page that provides transparency into who developed and verified the tool by introducing the project team (Figure 28).

The components and functionalities for the Prediction Page (see figure 27) are outlined below.

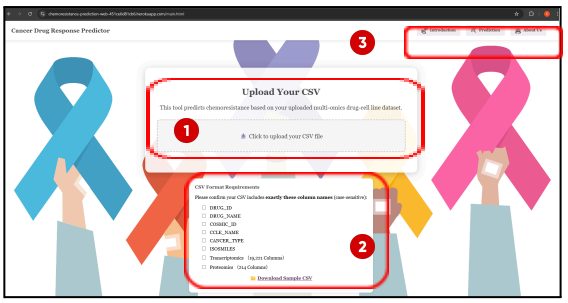

Figure 27 : Prediction Page Interface

1. Upload Box: A prominently framed upload box enables selection and submission of the multi-omics CSV file for chemoresistance prediction. Result page (Figure 29) will be displayed after the uploading and prediction is done.

2. Format Checkboxes: Before upload is enabled, a series of checkboxes containing the names of the necessary columns must all be verified to match precisely. Any column that is missing or incorrectly titled causes a warning and prevents the upload.
3. Navigation Menu: Provide access to other pages of the application, including Introduction, Prediction, and About Us.

The components and functionalities for the About Us Page (see figure 28) are outlined below.

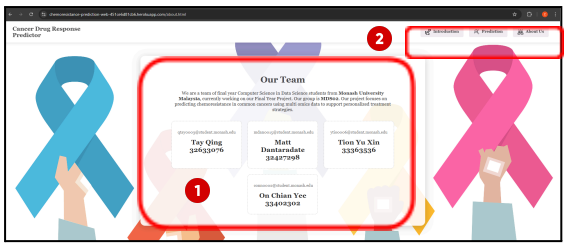

Figure 28: About Us Page Interface

1. Our Team Frame: Introduces the project contributors by displaying the name, student ID, and Monash email address of each team member in a bordered card arrangement.

2. Navigation Menu: Offers links to switch between pages, Introduction, Prediction, and About Us.

The components and functionalities for the Result Page (see figure 29) are outlined below.

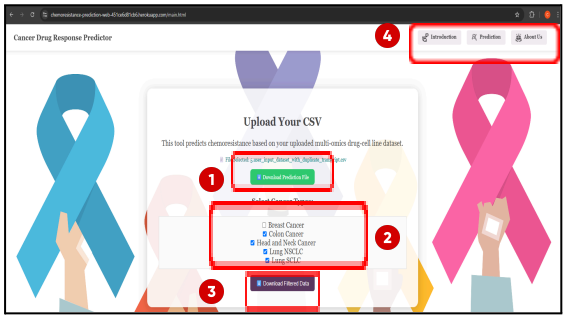

Figure 29: Result Page Interface

1. Download Prediction File Button: A prominent button that, when clicked, downloads the complete set of prediction results as a CSV file.

2. Cancer Types Checkboxes: A list of tumor-type checkboxes allowing selection of specific cancers.

3. Download Filtered File Button: Appears only if at least one cancer type is unchecked; clicking this button downloads a CSV containing predictions for only the selected cancer types.

4. Navigation Menu: Give users access to the application's additional pages, such as About Us, Prediction, and Introduction.

5.2 Software Quality Summary

5.2.1 Robustness

The robustness of the chemoresistance prediction web application was principally assessed through systematic testing to guarantee consistent performance across typical user interactions. During development, our team used black-box, integration testing and third-party testing strategies to imitate real-world user behavior and confirm the system's capacity

to handle both expected and incorrect input.

One key test involved uploading improperly formatted files (e.g., non-CSV formats or datasets missing required columns). The frontend was designed to block such inputs using JavaScript validation, displaying clear error messages to users. On the backend, integration tests confirmed that uploaded files were correctly saved, processed, and validated before being sent to the model. For instance, the preprocessing function was tested to halt execution if critical columns like DRUG_ID, ISOSMILES, or omics features were missing, displaying a corresponding error to the user interface.

The application was also tested with scenarios such as incorrect feature alignment, malformed numeric entries, and missing values. In each case, the system responded with clear, informative error messages and safely halted execution, demonstrating its ability to handle invalid inputs without crashing. However, robustness is limited when multiple users access the system at the same time. The current backend process handles one dataset at a time and lacks session isolation. During simulated simultaneous uploads, file conflicts and execution failures were detected. These tests indicate the need for future improvements such as asynchronous task queues and per-session directories to support robust multi-user access.

5.2.2 Security

Security concerns were not prioritized throughout development of the system. One of the key security decisions implemented was the use of HTTPS, which ensures that data transmitted between the user's browser and the server is encrypted. TThis prevents user-uploaded datasets from being intercepted during transmission and provides a minimal level of communication security.

However, several security limitations are present in the current implementation. The system does not support any form of user authentication or access control which means that anyone with the link can access and use the application. Uploaded files are temporarily stored on the server in an unencrypted format and are not automatically removed once used. Additionally, there is no isolation between user sessions, making the system vulnerable to unauthorized access and potential misuse.

Further improvements should include implementing user login functionality, restricted access to authorized users, file

encryption at rest, and automatic dataset deletion after processing. Assigning separate temporary directories for each of the user sessions would also improve isolation and reduce the potential for conflicts or data leakage.

5.2.3 Usability

Usability was a key consideration in the development of the chemoresistance prediction tool, with the goal of providing a straightforward and accessible experience for users with diverse technical backgrounds. The application was intentionally designed with simplicity and clarity in mind, making it accessible to a broad range of users, including oncologists, medical researchers, bioinformatics analysts and data scientists.

- **User Interface:** The web interface is simple and straightforward, consisting of three clear sections: Introduction, Prediction, and About Us. On the prediction page, users are guided step-by-step through the upload process using clearly labeled input fields, intuitive icons or images, helpful tooltips, and a checklist that ensures all required columns are present in the dataset. This design helps reduce the chance of user error and ensures that files are formatted correctly before processing.
- **Error Handling:** If a user uploads a invalid file, the system immediately responds with a clear and specific error message. For example, if the ISOSMILES column is missing, the system displays: “Missing required column: ISOSMILES. Please check your file and try again.” This direct feedback helps users quickly identify and fix issues without confusion. The system is designed to prevent processing until these errors are resolved, helping maintain data integrity.
- **Progress Feedback:** A loading spinner is displayed during longer processing times, informing users that their file is being handled. This visual cue improves transparency and reduces uncertainty while waiting for prediction results.

5.2.4 Scalability

Scalability of the Web Application

The current chemoresistance prediction tool is deployed on a single Heroku dyno using a Docker container, optimised for single-user or low-traffic scenarios. While this configuration is adequate for initial testing and demonstrations, it poses

significant limitations in multi-user or high-throughput environments.

Uploaded datasets are processed through a shared backend and stored temporarily on the dyno's local file system, which is both non-persistent and shared across sessions. This introduces a risk of file overwriting, data corruption, or failed predictions during concurrent usage. Additionally, Heroku imposes strict platform constraints such as a 30-second request timeout and a ~30 MB maximum request body size. These restrictions limit the system's ability to process large omics datasets or accommodate long-running inference jobs.

Despite these limitations, the application's modular and containerised architecture offers a strong foundation for future scalability. The following improvements could significantly enhance its capacity and performance:

- **Asynchronous Task Processing:** Implementing a background task queue (e.g., Celery with Redis or RabbitMQ) would allow prediction jobs to be processed independently of the main HTTP thread, enabling parallel execution of multiple user requests.

- **Session-Isolated File Handling:** Generating unique temporary directories for each user session using UUIDs would prevent file conflicts and ensure safe, isolated processing in concurrent environments.

- **Cloud-Based Storage:** Offloading temporary files to cloud object storage services like Amazon S3 would ensure persistence, scalability, and improved fault tolerance, especially for large datasets.

These enhancements would transform the system from a single-threaded prototype to a more robust, scalable platform capable of serving multiple users and managing larger computational demands.

Scalability of the Prediction Model

The chemoresistance prediction model is designed to handle high-dimensional omics datasets, including transcriptomic and proteomic features, making it inherently scalable in terms of data complexity. It can process large CSV files with thousands of columns, automatically align features to fit the model expected features, and perform inference without failure if sufficient system memory is available.

A key strength of the model is its consistent performance across different dataset sizes. It efficiently processes complex biological inputs and does not require manual reconfiguration when dataset dimensions change. During development and testing, the model showed stable behavior with no memory issues or performance bottlenecks in single-user scenarios, demonstrating suitability for larger-scale biological data applications.

5.2.5 Portability

The chemoresistance prediction tool is highly portable, as it is deployed as a web application accessible through a public URL. Users do not need to install any software or set up their local environment. Users accessing the tool through a browser is sufficient. This platform-independent design ensures compatibility across operating systems (e.g., Windows, macOS, Linux) and devices (e.g., laptops, tablets), provided an internet connection is available.

By hosting the tool on Heroku, deployment complexity is abstracted away from the end user, allowing the application to be used seamlessly in academic, research, or demonstration settings. This ease of use makes the system extremely portable and useful for collaborative or educational purposes.

5.2.6 Maintainability

Maintainability was a key consideration in the development of the chemoresistance prediction web application to ensure that future updates, enhancements, and debugging tasks can be performed efficiently and with minimal risk of introducing new issues. Several aspects of the system's design and development approach contribute to its overall maintainability:

- **Modular Structure**: The project is divided into separate modules for preprocessing (preprocessing_model.py), prediction (predict_chemoresistance.py), and the web interface (HTML, JavaScript). This modularity enables each component of the system to be worked on or upgraded independently, lowering the chance of system-wide errors.
- **Clean and Understandable Codebase**: The code follows consistent formatting, meaningful variable and function names, and includes explanatory comments where needed. This makes it easier

for future developers to understand and maintain the system without extensive onboarding.

- **Use of Git for Version Control**: Git was utilized throughout development to manage code changes, track contributions, and enable rollback when needed. This practice encourages collaborative development and makes debugging or feature integration easier.

The system currently lacks automated testing such as unit or integration tests. As a result, verifying new changes requires manual testing, which is more time-consuming and error-prone. This may hinder long-term maintainability, especially as the system grows more complex.

Introducing unit tests utilizing tools like pytest, as well as continuous integration pipelines (e.g., GitHub Actions), would help catch regressions early. Furthermore, adding inline documentation and keeping a setup guide will help developers maintain and extend the system.

6. Software and Project Critique

6.1 Project Execution and Overall Success

Our project, Multi-Omics Machine Learning Model for Predicting Chemoresistance, was successfully executed and delivered a robust, user friendly system that aligned closely with our initial goals. Firstly, we developed a machine learning pipeline that integrates transcriptomics and proteomics data to predict chemoresistance in cancer cell lines after experimenting with numerous combinations of datasets, and secondly, deployed the model as a fully functional web application with CSV input support, visualisations and prediction insights. The project was executed with clear direction, well defined milestones and strong teamwork. All team members contributed to both technical and managerial aspects, including model design, UI/UX development, testing and documentation.

We adopted an Agile development methodology, organizing work into iterative sprints, which allowed continuous integration and rapid adjustment to feedback. Weekly team meetings and supervisor consultations ensured we stayed

on track while refining ideas as the project evolved. This flexible yet focused approach was crucial to our success in balancing exploration with delivery.

6.2 Comparison with initial Project Proposal and Changes in Approach

Our initial project proposal focused on developing a machine learning model that predicts chemoresistance based on genomics data with the longer term goal of integrating multi-omics modalities, e.g., transcriptomics and proteomics. The emphasis was on accepting patient genetic variant information as the primary input for chemoresistance prediction and drug performance analysis.

However, during the dataset experimentations as shown in table 6, we discovered that genomics data did not increase predictive performance. Specifically, any training dataset that contains genomics data, even when joined with transcriptomics and proteomics data has very low R-square scores, and their RMSE values were higher than the models trained on just transcriptomics alone, the model trained with the transcriptomics and proteomics combination. This suggested that adding genomics introduced noise or feature sparsity that degraded model quality. We validated this trend across multiple experiments and drug subsets.

As a result, we made the deliberate decision to exclude genomics from the final model, and instead trained our machine learning pipeline solely on transcriptomics and proteomics. Consequently, the web platform accepts only these two data types as input which differs from the original vision of allowing genomics based uploads.

This deviation was data driven and necessary to ensure the robustness, accuracy and interpretability of the model. While it slight narrows the input scope compared to our proposal, it significantly improves practical performance and allows us to present a more reliable, validated and scientifically meaningful tool

6.3 Alignment with Initial Plan and Why It Worked

Our project remained largely aligned with the initial scope defined in the FIT3163 Project Proposal. All major in-scope components were successfully delivered, including dataset collection from reliable public sources, preprocessing (normalization and missing value handling), machine learning model development, and a fully functional

web-based prototype. The web interface allows users to upload a cancer cell line and their respective transcriptomics and proteomics values in a CSV file, and afterwards receive chemoresistance predictions with accompanying visualisations and insights, all presented through a clean and user-friendly UI/UX.

However, one in-scope activity, feature selection was deprioritized during implementation. Although initially planned to optimize computational efficiency, we determined through early experimentation that the model performed well using the full set of transcriptomics and proteomics features. As such, we focused our efforts on model validation and web integration, without implementing a formal feature selection pipeline. This decision allowed us to maintain predictive accuracy and deliver a complete working product within our project timeline.

We also stayed within our out-of-scope boundaries, as Mobile application development and integration with external commercial databases were intentionally excluded and remained outside the project's focus.

This strong alignment with scope was enabled by our consistent project management practices using Gantt charts and weekly supervisor feedback to ensure all team activities were aligned with clearly defined deliverables. By maintaining discipline around scope and adjusting priorities based on real-world performance and constraints, we ensured successful execution without overextending project resources.

Although the overall project execution was successful and aligned with our planned timeline and scope, several challenges emerged in relation to team management and schedule planning, especially during Semester 2.

One key issue was the lack of consistent team communication during the early weeks of the semester. Progress updates between members were irregular, which led to several avoidable problems, particularly with the inconsistencies in data preprocessing workflows. For instance, different team members had differing preprocessing scripts that caused integration issues when merging datasets. A more proactive, structured communication approach such as weekly internal check-ins have helped us identify and resolve these problems earlier.

Another reflection point relates to insufficient time allocated for foundational research in our sprint schedule. We

initially treated transcriptomics, proteomics and genomics as interchangeable omics layers in our model design. In hindsight, deeper upfront research would have shown that transcriptomics is the most consistent and informative base layer in chemoresistance studies. This knowledge only emerged after multiple model combinations and performance regressions, which could have been avoided had we scheduled more dedicated literature review time earlier in the semester.

Despite these setbacks, we were able to recover quickly by reorganizing task ownership and establishing clearer model experimentation logs. Our overall project timeline remained on track thanks to the buffer periods built into the Gantt chart in the FIT3163 project proposal, and the supervisor's weekly feedback helped keep us accountable and aligned with deliverables.

6.4 Risk Management, Stakeholders and Lesson Learned

One of the strongest indicators of our successful execution was that none of the identified project risks were triggered, despite their high impact potential. As shown in our Risk Management Table (see Appendix, Table 8), we had flagged critical risks such as limited access to high quality multi omics datasets (R1), complexity in integrating multi-omics data (R2), machine learning overfitting (R3), delays in meeting project deadlines (R7) and miscommunication within the team member (R6). These were mitigated through early stage preprocessing, validation pipelines, team collaboration strategies and regular sync ups. Our use of structured brainstorming, SWOT analysis and supervisor interviews for initial risk identification proved effective. The fact that all risks remained inactive is a testament to the foresight and adaptability of our team.

Our stakeholder engagement, particularly with our supervisor, was proactive and productive. Her feedback helped shape our priorities and refine design decisions. The clarity of communications and shared vision between team and supervisor was key to our consistency and final success.

6.5 Potential Improvements and Future Recommendations

Although the project achieved its core goals, we identified several areas for future improvements as stated in 3.6 Limitations of Project Outcomes.

(1) Expand Biological and Cancer Coverage

Future versions should incorporate more cancer types prevalent in Malaysia, such as liver, prostate, and leukaemia, and include additional omics layers like metabolomics and genomic mutations to improve prediction accuracy and applicability.

(2) Strengthen Testing and Validation

Introduce external benchmarking datasets and perform load/stress testing to evaluate generalisability and system performance under real world usage conditions.

(3) Improve Security and Data Privacy

Implement automatic file deletion, encrypted storage, and user authentication to safeguard uploaded datasets and align with ethical and regulatory standards.

7. Conclusion

In a nutshell, this project has produced a sophisticated, user-friendly web platform for chemoresistance forecasting in addition to a cutting-edge predictive algorithm. The main deep neural network obtained an R2 of 0.804 with an RMSE of 1.22 on unseen data after being trained on an integrated multi-omics dataset that combined transcriptomic, proteomic, and chemical fingerprint information (refer to Appendix Table 6). The transcriptomics-plus-proteomics combination consistently produces the highest accurate drug-response estimates, surpassing single-omics techniques by more than 30%, according to thorough ablation experiments conducted across five dataset configurations (refer to section 4.3.2).

Using a lightweight web framework, a cloud-hosted web application has operationalised these modelling strengths. From a straightforward introduction to project goals and data requirements, the interface leads users through a three-tab workflow: Introduction, Prediction, and About Us. In order to prevent inputs that are faulty, the Prediction page uses interactive checkboxes and detailed alerts to impose stringent CSV validation. Following data validation, users are presented with interactive bar charts that display real-time predictions in a results table that can be sorted and filtered. Subsetting by cancer type can be done instantly with patient-style filters, and downstream analyses are supported with fully downloadable outputs.

Although the present implementation supports two omics layers and operates on a single cloud instance, its modular architecture easily enables additional

cancer subtypes, explainability modules, scalable hosting, and new data modalities. When combined, the platform's excellent predictive capabilities and smooth user interface make it a solid basis for upcoming clinical decision-support technologies.

8. References

1. Al-Amrani, S., Al-Jabri, Z., Al-Zaabi, A., Alshekaili, J., & Al-Khabori, M. (2021). Proteomics: Concepts and applications in human medicine. World journal of biological chemistry, 12(5), 57–69. https://doi.org/10.4331/wjbc.v12.i5.57

2. Dantaradate, M., Tion, Y. X., On, C. Y., & Qing, T. (2024). FIT3163: Data Science Project 1, Semester 2 2024. Project Proposal & Literature Review, 54.

3. Dantaradate, M., Tion, Y. X., On, C. Y., & Qing, T. (2024). FIT3164: Data Science Project 2, Semester 1 2025. User Guide and Testing Report.

4. Global Cancer Observatory - Malaysia. (n.d.). https://gco.iarc.who.int/media/globocan/factsheets/populations/458-malaysia-fact-sheet.pdf

5. ISO 13485:2016 Medical Devices — Quality Management Systems — Requirements for Regulatory purposes. (2016, March). ISO. https://www.iso.org/standard/59752.html

6. Robert E. Martell, David G. Brooks, Yan Wang, Keith Wilcoxen, Discovery of Novel Drugs for Promising Targets, Clinical Therapeutics, Volume 35, Issue 9, 2013, Pages 1271-1281, ISSN 0149-2918, https://doi.org/10.1016/j.clinthera.2013.08.005.

7. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. https://jmlr.org/papers/v15/srivastava14a.html

8. Wu, Y., Chen, M., & Qin, Y. (2025). Anticancer drug response prediction integrating multi-omics pathway-based difference features and multiple deep learning techniques. PLoS computational biology, 21(3), e1012905. https://doi.org/10.1371/journal.pcbi.1012905

9. Appendix

No	Dataset	RMSE	MAE	MSE	R-Square
1	Transcriptomics GDSC2	2.474	1.939	6.119	0.029
2	Transcriptomics GDSC2 isoSMILES	1.378	1.067	1.898	0.765
3	**Transcriptomics** **Proteomics** **GDSC2** **isoSMILES**	**1.22**	**0.952**	**1.488**	**0.804**
4	Transcriptomics Geonomics GDSC2 isoSMILES	1.636	1.276	2.678	0.671
5	Transcriptomics Proteomics Geonomics GDSC2 isoSMILES	1.78	1.416	3.171	0.633

Table 6: Experiment result table

Req. ID	Description	Type (s) (FR/NFR)	Categories	Source	How It Was Met	Status
R-001	Select multi-omics datasets for training the machine learning model, and it must be from a publicly available, reliable, and reputable source.	FR and NFR	Data Collection	Supervisor, Project management document, Data analysis report	Used CCLE transcriptomics & proteomics with intermediary mappings on their unique identifier to join them. This, along with GDSC drug data and isoSMILES are all publicly available.	Complete
R-002	The collected datasets for machine learning model training must be cleaned, normalised, and missing values are handled.	FR	Data Preprocessing	Supervisor, team members	Any duplicated rows during pre-processing were dropped. NA values were checked (there were none). Checking was done extensively throughout the user guide and testing report.	Complete
R-003	Only relevant features should be used for the training to reduce noise and extra computational overhead. A feature selection method will be used.	FR	Feature Selection	Supervisor, team members	Removed after supervisor discussion. Deep learning models inherently handle feature weighting, making separate feature selection unnecessary.	Removed*
R-004	Training machine learning models on the multi-omics data to predict chemoresistance.	FR	AI Modeling	Supervisor, team members	Trained a deep neural network model using transcriptomic and proteomic features to predict LN_IC50 drug resistance values.	Complete
R-005	The model performance/validation using cross-validation, ROC-AUC scores, accuracy, precision, etc., should be good.	FR	AI Modeling	Supervisor, team members	Evaluated on test data with $R^2 = 0.84$, cross-checked during implementation.	Complete
R-006	Optimise the run-time of the machine learning model.	FR	AI Modeling	Supervisor, team members	Reduced the number of neurons to try to 3 for each layer	Complete
R-007	Identify biomarkers related to chemoresistance and associate them with drug response through the trained model.	FR	Performance Evaluation	Supervisor, team members	The DNN model has a relatively good RMSE and R^2 values.	Complete
R-008	Develop a prototype	FR	Biomarker	Supervisor,	Deployed Flask-based	Complete

	website to display the predictive model for users to use.		Discovery	team members	website hosted on Heroku, enabling file upload, view result real-time and result download	
R-009	Ensure the model can output the prediction and other information relatively quickly.	NFR	Prototype Development	Supervisor, team members	Prediction time varies with dataset size but is typically under 30 seconds, acceptable for web use	Complete
R-010	Test model with new datasets to ensure reliability and consistency with the performance validation.	FR	Prototype Development	Supervisor, team members	Model tested with multiple external user-uploaded datasets; supports dynamic feature alignment and handles imperfect input structures.	Complete
R-011	Ensure the model website display has a user-friendly UI and UX design.	FR	Validation	Supervisor, team members	Built a simple web interface with tooltips, input checklists, filters, and clear error messages.	Complete

Table 7: Requirement Traceability Matrix

**R-003 removed as deep learning models are capable of handling relevant features already*

ID	Description	Root Cause	Trigger	Risk Response	Incident Response	Owner	Probability	Impact Score	Overall Score	Risk Status	Latest Update
R1	Limited access to high quality multi omics datasets	Datasets are incomplete or inconsistent	Datasets not meeting quality standards	Prioritise using high quality publicly available datasets	Gather available datasets and evaluate quality	Quality Assurance: Tay Qing	50%	8	400	Resolved	04/09/2024
R2	Complexity in integrating multi-omics data	Difficulty in combining various types of omics data	Errors in data integration appear	Use advanced bioinformatics tools and proper integration algorithms	Validate integration with smaller subsets first	Technical Lead: On Chian Yee	60%	9	540	Triggered	20/09/2024
R3	Machine learning model overfitting	Over complex models that overfit on training data	Poor generalisation on test data	Implement regularisation techniques, cross-validation	Test model on separate test set regularly	Technical Lead: Tion Yu Xin	40%	7	280	Monitored	20/09/2024
R4	Difficulty interpreting	Model complexity makes it hard to	Difficulty explaining	Use interpretable	Provide interpretat	Technical Lead:	30%	6	180	Monitored	20/09/2024

	machine learning model	interpret	model results	models (e.g., Random Forest)	ion explanations during team meetings	Tion Yu Xin					
R5	Lack of funding for advanced resources	High costs for advanced computing resources	Budget constraints emerge	Utilise open-source tools and seek additional funding through grants.	Prioritise tasks and adjust resources accordingly.	Project Manager: Matt Dantaradate	20%	5	100	Monitored	20/09/2024
R6	Miscommunication within the team member	Lack of effective communication	Mismatched goals for misalignment	Schedule regular team meetings	Revisit project requirements and clarify miscommunications	Project Manager: Matt Dantaradate	40%	6	240	Monitored	20/09/2024
R7	Delays in meeting project deadlines	Resource constraints or dependencies on external factors	Missed deadlines	Create a detailed project plan	Re-allocate resources	Project Manager: Matt	40%	7	280	Monitored	20/09/2024

				with contingency buffers	to address any bottlenecks	Dantaradate					
R8	Data loss or corruption	Inadequate data storage solutions	Data becomes inaccessible or corrupted	Implement regular data backups	Restore data from backup if necessary	Quality Assurance: Tay Qing	20%	9	180	Monitored	20/09/2024

Table 8: Risk Management

```python
# Utility: Align input features to match training features
def align_features(X_new, expected_features):
    """
    Aligns new input data to match the feature structure expected by the trained model.
    - Adds missing features with default values.
    - Removes extra features not seen during training.

    Parameters:
        X_new (pd.DataFrame): New input data submitted by the user.
        expected_features (list): Features used during model training.

    Returns:
        pd.DataFrame: Aligned dataframe matching model expectations.
    """

    # Identify features expected by the model but missing from user input
    missing_features = list(set(expected_features) - set(X_new.columns))
    if missing_features:
        # Log a preview of missing features for debugging
        print(f"Adding missing features: {missing_features[:5]} ... (total: {len(missing_features)})")

        # Create a new DataFrame of missing features filled with 0.0 (default)
        missing_df = pd.DataFrame(0.0, index=X_new.index, columns=missing_features)

        # Append the missing columns to the input
        X_new = pd.concat([X_new, missing_df], axis=1)

    # Identify extra features in user input that are not needed by the model
    extra_features = list(set(X_new.columns) - set(expected_features))
    if extra_features:
        # Log a preview of extra features to be removed
        print(f"Dropping extra features: {extra_features[:5]} ... (total: {len(extra_features)})")

        # Drop unnecessary columns to ensure input matches training schema
        X_new = X_new.drop(columns=extra_features)

    # Reorder columns to exactly match the expected training order
    X_new = X_new.reindex(columns=expected_features)

    # Confirm total number of aligned features after preprocessing
    print(f"Feature alignment complete. Total features: {X_new.shape[1]}")

    return X_new
```

Figure 30. Function to align user input features with the model's expected features.

```python
# Check if the scaler has the list of expected features
if hasattr(scaler, 'feature_names_in_'):
    expected_features = list(scaler.feature_names_in_)

    # Align the current input features with those expected by the model
    X_new = align_features(X_new, expected_features)
else:
    # In rare cases (e.g., older pickled scaler), skip alignment with warning
    print("⚠ Scaler does not have attribute 'feature_names_in_'. Feature alignment skipped.")
```

Figure 31. Code that calls align _features().

```python
def save_predictions(original_data, predictions, output_file='predictions_output.csv'):
    """
    Combines original metadata with model predictions and saves the result to a CSV file.

    Parameters:
        original_data (pd.DataFrame): The user's cleaned input data.
        predictions (np.array): The predicted LN_IC50 values from the model.
        output_file (str): The filename for the final output CSV.

    Returns:
        None. Outputs a file to disk.
    """

    # Create a new DataFrame with selected metadata fields
    results_df = original_data[['DRUG_ID', 'DRUG_NAME', 'COSMIC_ID', 'CCLE_Name', 'CANCER_TYPE']].copy()

    # Add the predicted numeric output (LN_IC50) to the result
    results_df['Predicted_LN_IC50'] = predictions

    # Define helper function to classify drug resistance levels
    def classify(ln_ic50):
        """
        Maps a continuous LN_IC50 value to a categorical resistance class:
        - High (Sensitive), Intermediate, or Low (Resistant).
        """
        if ln_ic50 < 2.36:
            return "High"  # High sensitivity lower than 2.36
        elif ln_ic50 <= 5.26:
            return "Intermediate"
        else:
            return "Low"   # Low sensitivity higher than 5.26

    # Apply classification to all rows
    results_df['Sensitivity'] = results_df['Predicted_LN_IC50'].apply(classify)

    # Track initial number of rows
    before = results_df.shape[0]

    # Remove duplicate entries based on key metadata fields
    results_df = results_df.drop_duplicates(
        subset=['DRUG_ID', 'COSMIC_ID', 'CCLE_Name', 'DRUG_NAME', 'CANCER_TYPE'],
        keep='first'
    )

    # Sort results by resistance level (ascending IC50 = more sensitive)
    results_df = results_df.sort_values(by='Predicted_LN_IC50', ascending=True)

    # Optional: log how many duplicates were removed
    after = results_df.shape[0]
    if before != after:
        print(f"⚠ Dropped {before - after} duplicate rows from predictions.")

    # Save the final structured output to CSV file
    results_df.to_csv(output_file, index=False)
    print(f"☑ Predictions saved to {output_file}")
```

Figure 32. Function to classify sensitivity levels and export results to a CSV file.

```python
# Step 3: Make predictions
def make_predictions(model, X_scaled):
    """
    Performs inference using the trained deep learning model.

    Parameters:
        model (keras.Model): Loaded Keras model for prediction.
        X_scaled (np.array): Scaled input features.

    Returns:
        np.array: Flattened prediction outputs (LN_IC50 values).
    """
    # Use the model to generate predictions from the scaled input data
    predictions = model.predict(X_scaled)

    # Flatten the prediction array to a 1D array for easier post-processing
    return predictions.flatten()
```

Figure 33. Function that performs prediction using the trained deep learning model and returns flattened LN_IC50 values for downstream processing.

```python
# Step 5: Main function to orchestrate the prediction workflow
def main(input_file_path):
    """
    Main entry point for processing a user dataset and generating chemoresistance predictions.

    Workflow:
    1. Preprocess user input.
    2. Load trained model and scaler.
    3. Align features and apply scaling.
    4. Make predictions.
    5. Save results to CSV.

    Parameters:
        input_file_path (str): Path to the user's input CSV file.
    """

    try:
        # Step 1: Inform the user that processing has started
        print(f"🗁 Starting prediction for: {input_file_path}")

        # Step 2: Clean and validate input data, including checking required columns,
        # handling missing values, and generating Morgan fingerprints.
        df = preprocess_user_dataset(input_file_path)

        # Step 3: Load the pre-trained TensorFlow model and the fitted scaler (e.g., StandardScaler)
        model, scaler = load_pipeline()

        # Step 4: Align features to match model expectations and scale the inputs
        _, X_scaled = preprocess_new_data("user_preprocessed_output.csv", scaler)

        # Step 5: Make predictions using the trained model
        predictions = make_predictions(model, X_scaled)

        # Step 6: Save the final results to a CSV file with metadata and resistance class
        save_predictions(df, predictions)

        # Success confirmation
        print("☑ Prediction completed and saved to predictions_output.csv")

    except Exception as e:
        # Log error to stderr and exit with non-zero status to signal failure
        print(str(e), file=sys.stderr)
        sys.exit(1)
```

Figure 34. Function that calls make_predictions() and save_prediction().