National College of Ireland

Explainable Multi-Omics AI Framework for Breast Cancer Stage Classification: Integrating Gene Expression, Proteomics, and Mutation Data.

MSc Research Project

MSc in Artificial Intelligence

Sriram Samala

Student ID: 23338032

School of Computing

National College of Ireland

Supervisor: Arundev Vamadevan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Sriram Samala
Student ID:	23338032
Programme:	MSc in Artificial Intelligence **Year:** 2024-2025
Module:	Researc practicum
Lecturer:	Arundev vamadevan
Submission Due Date:	11/08/2025
Project Title:	Explainable Multi-Omics AI Framework for Breast Cancer Stage Classification: Integrating Gene Expression, Proteomics, and Mutation Data.
Word Count:	..... **Page Count:** .....

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Sriram Samala

Date: 11/08/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□

Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

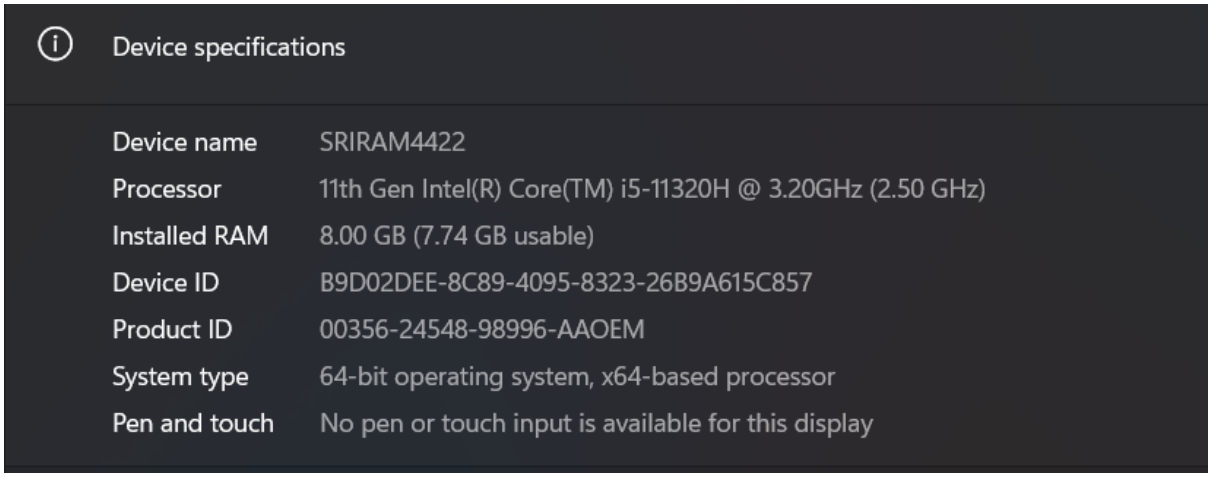
Explainable Multi-Omics AI Framework for Breast Cancer Stage Classification: Integrating Gene Expression, Proteomics, and Mutation Data.

Sriram Samala
Student ID: 23338032

1 Introduction

This research aims to present an explainable AI pipeline that performs breast cancer stage classification using multi-omics data from TCGA-BRCA. The multi-omics data included gene expression, proteomics, and mutation profiles. The preprocessing, feature selection, and dimensionality reduction using a deep autoencoder occurred in the same pipeline, as did the classification using XGBoost. The molecular drivers behind the predictions were interpreted using SHAP and LIME, which promote clinical trust and future applications of precision oncology.

2 System Specifications



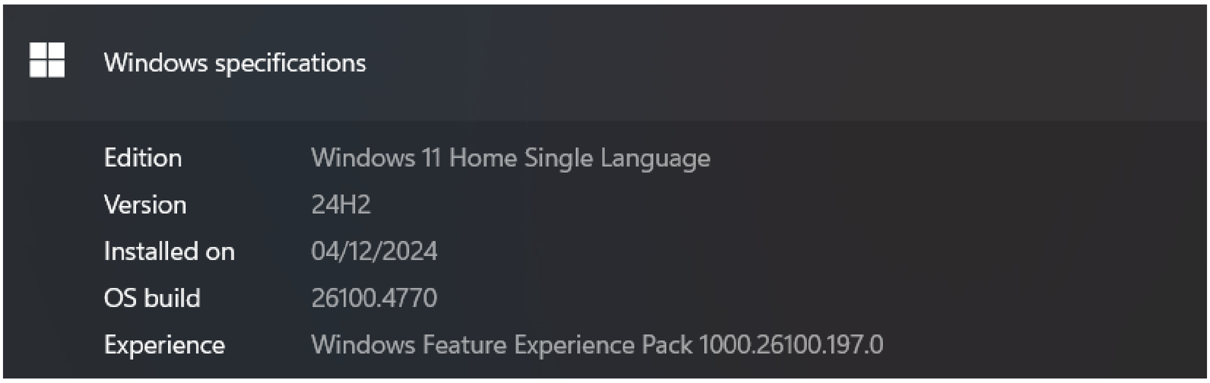


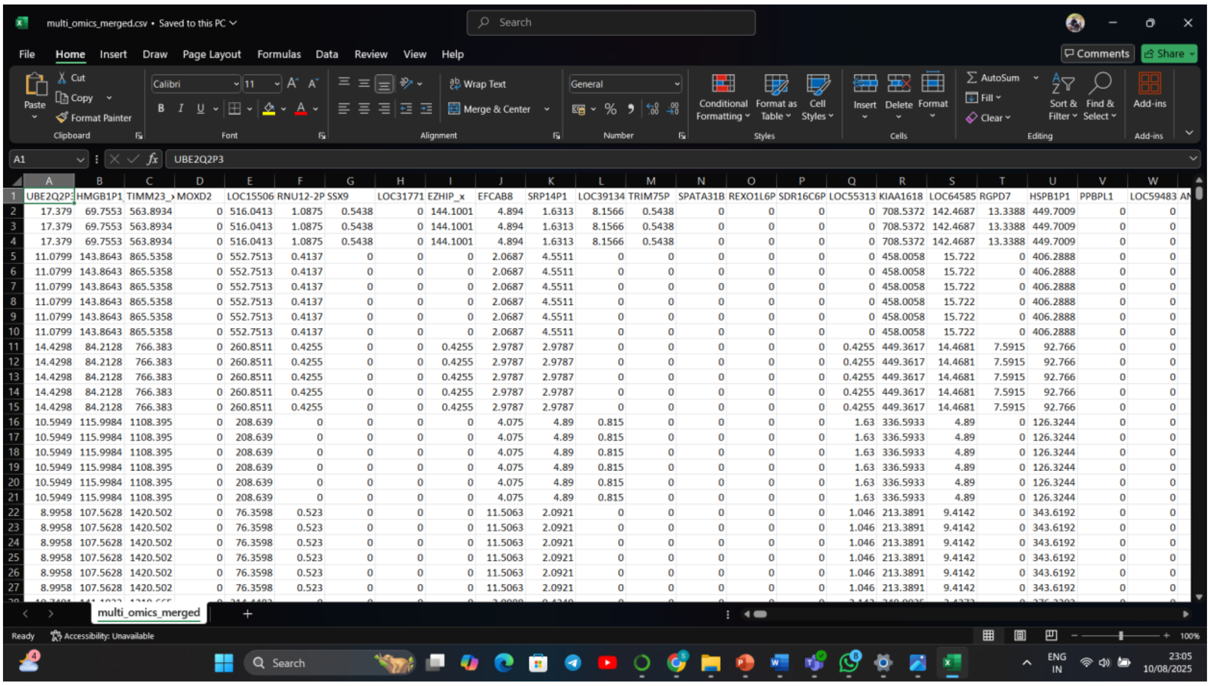
Figure 1. System specifications

Software Used:

- Microsoft excel: Used for custom dataset description.
- Jupyter notebook: Used for all processing and as code runtime environment
- Device: Used to store models, datasets and handle while runtime

3 Dataset specification

The dataset for the current study was obtained from TCGA-BRCA cohort and includes three omics layers: gene expression measures from RNA-seq, protein abundance values from RPPA assays, and binary somatic mutation data. All three modalities were aligned by patient ID for accurate multi-omics alignment. Quality control procedures involved dropping samples or features with more than 25% missing data, mean imputation for the remainder of the missing data, and standardization to a common scale. The dataset was preprocessed with approximately 3,500 breast cancer tissue samples and around 20,000 molecular features included after processing. Clinical stage labels were binarized to categorize as Early (Stage I/II) and Late (Stage III/IV) to inform classification tasks in the AI pipeline.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	UBE2Q2P3	HMGB1P1	TIMM23_x	MOXD2	LOC15506	RNU12-2P	SSX9		LOC31771	EZHIP_x	EFCAB8	SRP14P1	LOC39134	TRIM75P	SPATA31B	REXO1L6P	SDR16C6P	LOC55313	KIAA1618	LOC64585	RGPD7	HSPB1P1	PPBPL1	LOC59483
2	17.379	69.7553	563.8934	0	516.0413	1.0875	0.5438	0	144.1001	4.894	1.6313	8.1566	0.5438	0	0	0	0	708.5372	142.4687	13.3388	449.7009	0	0
3	17.379	69.7553	563.8934	0	516.0413	1.0875	0.5438	0	144.1001	4.894	1.6313	8.1566	0.5438	0	0	0	0	708.5372	142.4687	13.3388	449.7009	0	0
4	17.379	69.7553	563.8934	0	516.0413	1.0875	0.5438	0	144.1001	4.894	1.6313	8.1566	0.5438	0	0	0	0	708.5372	142.4687	13.3388	449.7009	0	0
5	11.0799	143.8643	865.5358	0	552.7513	0.4137	0	0	0	2.0687	4.5511	0	0	0	0	0	0	458.0058	15.722	0	406.2888	0	0
6	11.0799	143.8643	865.5358	0	552.7513	0.4137	0	0	0	2.0687	4.5511	0	0	0	0	0	0	458.0058	15.722	0	406.2888	0	0
7	11.0799	143.8643	865.5358	0	552.7513	0.4137	0	0	0	2.0687	4.5511	0	0	0	0	0	0	458.0058	15.722	0	406.2888	0	0
8	11.0799	143.8643	865.5358	0	552.7513	0.4137	0	0	0	2.0687	4.5511	0	0	0	0	0	0	458.0058	15.722	0	406.2888	0	0
9	11.0799	143.8643	865.5358	0	552.7513	0.4137	0	0	0	2.0687	4.5511	0	0	0	0	0	0	458.0058	15.722	0	406.2888	0	0
10	11.0799	143.8643	865.5358	0	552.7513	0.4137	0	0	0	2.0687	4.5511	0	0	0	0	0	0	458.0058	15.722	0	406.2888	0	0
11	14.4298	84.2128	766.383	0	260.8511	0.4255	0	0	0.4255	2.9787	2.9787	0	0	0	0	0	0.4255	449.3617	14.4681	7.5915	92.766	0	0
12	14.4298	84.2128	766.383	0	260.8511	0.4255	0	0	0.4255	2.9787	2.9787	0	0	0	0	0	0.4255	449.3617	14.4681	7.5915	92.766	0	0
13	14.4298	84.2128	766.383	0	260.8511	0.4255	0	0	0.4255	2.9787	2.9787	0	0	0	0	0	0.4255	449.3617	14.4681	7.5915	92.766	0	0
14	14.4298	84.2128	766.383	0	260.8511	0.4255	0	0	0.4255	2.9787	2.9787	0	0	0	0	0	0.4255	449.3617	14.4681	7.5915	92.766	0	0
15	14.4298	84.2128	766.383	0	260.8511	0.4255	0	0	0.4255	2.9787	2.9787	0	0	0	0	0	0.4255	449.3617	14.4681	7.5915	92.766	0	0
16	10.5949	115.9984	1108.395	0	208.639	0	0	0	0	4.075	4.89	0.815	0	0	0	0	1.63	336.5933	4.89	0	126.3244	0	0
17	10.5949	115.9984	1108.395	0	208.639	0	0	0	0	4.075	4.89	0.815	0	0	0	0	1.63	336.5933	4.89	0	126.3244	0	0
18	10.5949	115.9984	1108.395	0	208.639	0	0	0	0	4.075	4.89	0.815	0	0	0	0	1.63	336.5933	4.89	0	126.3244	0	0
19	10.5949	115.9984	1108.395	0	208.639	0	0	0	0	4.075	4.89	0.815	0	0	0	0	1.63	336.5933	4.89	0	126.3244	0	0
20	10.5949	115.9984	1108.395	0	208.639	0	0	0	0	4.075	4.89	0.815	0	0	0	0	1.63	336.5933	4.89	0	126.3244	0	0
21	10.5949	115.9984	1108.395	0	208.639	0	0	0	0	4.075	4.89	0.815	0	0	0	0	1.63	336.5933	4.89	0	126.3244	0	0
22	8.9958	107.5628	1420.502	0	76.3598	0.523	0	0	0	11.5063	2.0921	0	0	0	0	0	1.046	213.3891	9.4142	0	343.6192	0	0
23	8.9958	107.5628	1420.502	0	76.3598	0.523	0	0	0	11.5063	2.0921	0	0	0	0	0	1.046	213.3891	9.4142	0	343.6192	0	0
24	8.9958	107.5628	1420.502	0	76.3598	0.523	0	0	0	11.5063	2.0921	0	0	0	0	0	1.046	213.3891	9.4142	0	343.6192	0	0
25	8.9958	107.5628	1420.502	0	76.3598	0.523	0	0	0	11.5063	2.0921	0	0	0	0	0	1.046	213.3891	9.4142	0	343.6192	0	0
26	8.9958	107.5628	1420.502	0	76.3598	0.523	0	0	0	11.5063	2.0921	0	0	0	0	0	1.046	213.3891	9.4142	0	343.6192	0	0
27	8.9958	107.5628	1420.502	0	76.3598	0.523	0	0	0	11.5063	2.0921	0	0	0	0	0	1.046	213.3891	9.4142	0	343.6192	0	0

Figure 2 : multi omics dataset

4. Project Development

After collecting all data and storing it on a drive or storage place, the jupyter/colab notebook can be launched. Click on File, followed by Open notebook. As all the code files are stored in an .ipynb file, it can be opened easily and will contain all previous run results. You can mount a drive or use the notebook storage space for uploading and accessing the dataset. There is an option to run one by one or run them simultaneously.

4.1 Importing Library:

All the required Python libraries and packages are displayed in Figure 4. The jupyter notebook platform comes with several pre-installed versions and libraries. If necessary, please install the required libraries in your environment. These libraries are freely available and can be easily installed from the official Python site.

```
import pandas as pd, numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix
import xgboost as xgb
import shap
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_selection import SelectKBest, f_classif
from sklearn.pipeline import Pipeline
from sklearn.feature_selection import VarianceThreshold
from imblearn.pipeline import make_pipeline as imblearn_pipeline
from imblearn.over_sampling import SMOTE
import time
import warnings
from sklearn.model_selection import StratifiedKFold, cross_val_score
from xgboost import XGBClassifier
import lime
import lime.lime_tabular
from sklearn.impute import SimpleImputer
import joblib
from sklearn.decomposition import PCA
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input, Dense
from tensorflow.keras.optimizers import Adam
from sklearn.metrics import roc_auc_score, average_precision_score
from sklearn.metrics import precision_recall_curve
```

Figure 3: Packages used in the Project

4.2 Importing Files:

```
expr = pd.read_csv(r"C:/Users/SRIRAM/Downloads/data_mrna_seq_v2_rsem.txt", sep="\t", engine="python")
```

Figure 4: Importing Files

4.3 Preprocessing and data validation:

Data validation is important part of exploration. Figure 5 shows code implementation for Data validation techniques used.

```
import pandas as pd
import numpy as np
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler
from sklearn.feature_selection import SelectKBest, f_classif

# Dropping columns with too many NaNs (>25%)
na_threshold = 0.25
valid_cols = merged.isnull().mean() <= na_threshold
merged_filtered = merged.loc[:, valid_cols]

# Separate features and target
X = merged_filtered.drop(columns=['Sample_ID', 'label'], errors='ignore
y = merged_filtered['label']

# Impute missing values using mean
imputer = SimpleImputer(strategy='mean')
X_imputed = imputer.fit_transform(X)

# Clip negative values to avoid NaNs during log transform
X_imputed_clipped = np.clip(X_imputed, a_min=0, a_max=None)

# Log transform to reduce skew
X_log = np.log2(X_imputed_clipped + 1)
print(" Log transform done. NaNs after log2?", np.isnan(X_log).sum())
```

```python
# Standardize features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X_log)

# Feature selection (top 500 features by F-score)
selector = SelectKBest(score_func=f_classif, k=500)
X_selected = selector.fit_transform(X_scaled, y)

# Retrieve selected feature names for SHAP/LIME later
selected_indices = selector.get_support(indices=True)
selected_feature_names = X.columns[selected_indices]

print("Final shape for model training:", X_selected.shape)
```

```
 Log transform done. NaNs after log2? 0
Final shape for model training: (3549, 500)
```

Figure 5: Data preprocessing for multi-omics

4.5 Modelling:

Model 1 on single omics:

```
# Train and evaluate
print("\n Training Logistic Regression")
start = time.time()
pipeline.fit(X_train, y_train)
y_pred = pipeline.predict(X_test)
print(classification_report(y_test, y_pred))
print(" Training time:", round(time.time() - start, 2), "seconds")

# Cross-validation on full data
scores = cross_val_score(pipeline, X, y, cv=5)
print(f" CV Accuracy: {scores.mean():.4f} ± {scores.std():.4f}")
```

```
 Training Logistic Regression
              precision    recall  f1-score   support

         0.0       0.82      0.66      0.73       717
         1.0       0.40      0.61      0.49       270

    accuracy                           0.65       987
   macro avg       0.61      0.64      0.61       987
weighted avg       0.71      0.65      0.66       987

 Training time: 25.53 seconds
 CV Accuracy: 0.5579 ± 0.0262
```

Model 2 on single omics:

```
Training Regularized Random Forest...
 Classification Report (Test Set):
              precision    recall  f1-score   support

         0.0       0.94      0.85      0.89       717
         1.0       0.68      0.85      0.76       270

    accuracy                           0.85       987
   macro avg       0.81      0.85      0.82       987
weighted avg       0.87      0.85      0.86       987

 ROC-AUC Score: 0.9297
 Training time: 0.48 seconds
 CV Accuracy: 0.8631 ± 0.0130
```

Model 3 on single omics(best model performance):

```
Training XGBClassifier...

 Classification Report (Test Set):
              precision    recall  f1-score   support

         0.0       0.94      0.99      0.96       717
         1.0       0.98      0.82      0.90       270

    accuracy                           0.95       987
   macro avg       0.96      0.91      0.93       987
weighted avg       0.95      0.95      0.95       987


 ROC-AUC Score: 0.9884

 CV Accuracy: 0.9326 ± 0.0061
 Training time: 1.27 seconds
```

Model 4 on multi-omics:

```
 scale_pos_weight set to: 2.62
 Training XGBoost with class balancing...
 Model and generated feature names saved as 'trained_xgboost_tcga_model.pkl'

 Classification Report (Test Set):
              precision    recall  f1-score   support

         0.0       1.00      0.99      1.00       514
         1.0       0.98      1.00      0.99       196

    accuracy                           1.00       710
   macro avg       0.99      1.00      0.99       710
weighted avg       1.00      1.00      1.00       710

 Confusion Matrix:
[[511   3]
 [  0 196]]
 ROC-AUC: 0.9999
 Training time: 1.34 seconds
 CV ROC-AUC: 0.9986 ± 0.0010
```

```
Classification Report (Autoencoder Features):
              precision    recall  f1-score   support

         0.0       0.99      0.95      0.97       514
         1.0       0.89      0.97      0.93       196

    accuracy                           0.96       710
   macro avg       0.94      0.96      0.95       710
weighted avg       0.96      0.96      0.96       710

ROC-AUC: 0.9951560390693241
CV ROC-AUC: 0.9856 ± 0.0052
```

References

Nicora, G., Vitali, F., Dagliati, A., et al. (2020). Integrated multi-omics analyses in oncology: A review of machine learning applications. *Briefings in Bioinformatics*, 21(2), 459–475. doi:https://doi.org/10.1093/bib/bbz038.

Hunter, D.J., Reddy, K.S., Cobbs, E.L., et al. (2022). Artificial intelligence in early cancer diagnosis. *Cancer*, 128(4), 674–688. doi:https://doi.org/10.1002/cncr.33959.