Reproduction of Research: Reproducible Survival Prediction with SEER Cancer Data

Elizabeth Amundsen
University of Illinois, Urbana - Champaign
Department of Computer Science
201 North Goodwin Avenue Urbana, IL 61801
ecreigh2@illinois.edu

Abstract

Reproducibility is a crucial aspect of all scientific research. Without objective verification from an alternative, non-related source, results of the research can be suspect. The objective of this project is to replicate the results of a research paper that was design to be highly reproducible and therefore showing research reproducibility is not an impossible goal.

Video — https://mediaspace.illinois.edu/media/t/1_y4wyz28m

Source Code — https://github.com/Kaeldra1304/mcs598-SEERCancerStudy

PyHealth Pull Request — https://github.com/sunlabuiuc/PyHealth/pull/382

Introduction

The research study *Reproducible Survival Prediction with SEER Cancer Data* (Hegselmann et al. 2018) was selected for this project. A copy can be found at (Proceedings of Machine Learning Research 2025a) and (Proceedings of Machine Learning Research 2025b). The study performs a literature review and experiment on cancer survival prediction at the 1-year and 5-year marks. It found no repeatable research studies through its literature review so a replicable experiment was designed using cancer data from the Surveillance, Epidemiology, and End Results (SEER) (National Cancer Institute 2025d). It's data selection files and source code are available in a code repository on GitHub (Hegselmann, S. 2018). The authors emphasize the importance of reproducibility for biomedical machine learning studies in order to compare and validate new machine learning approaches. They strived to set new methodology standards for future research through the inclusion of data cohort and source code.

Scope of Reproducibility

As mentioned in the project proposal, only the experiment portion of the paper was implemented. A majority of the experimentation steps were approximately reproduced but could not be exactly replicated due to data availability.

Dataset Processing. The SEER cancer data is a live database and only keeps the last 3 years of datasets so the exact data used by the paper was unavailable. Data sources have been dropped/modified and variables have been removed/modified/added since the paper was published. Additionally, the data formatting has been revised. Details of the differences are outlined in the Data section.

Model. All models were implemented using scikit-learn and Keras python libraries obtained from the original study's source code. Library upgrades may have produced minor changes. While hyperparameter ranges were discussed, the final model parameters were not documented so it is unknown if model training was accurately replicated. Details on hyperparameter tuning are in the Training section.

Baselines. The baseline model, scikit-learn's DummyClassifier(), produced very similar results between the paper and this replication, though it was not an exact match. This was mostly due to a difference in data. See Results section.

Methodology

Environment

Python version: 3.10.11

Dependencies

absl-py	2.1.0	bleach	1.5.0
cycler	0.11.0	edward	1.3.5
enum34	1.1.10	graphviz	0.8.2
html5lib	0.9999999	Keras	2.13.1
Markdown	3.7	matplotlib	3.7.2
numpy	1.24.3	pandas	1.5.3
protobuf	4.25.6	pydot-ng	1.0.0
pyparsing	3.2.1	python-dateutil	2.8.2
pytz	2022.7	PyYAML	6.0.1
scikit-learn	1.3.1	scipy	1.11.2
six	1.17.0	tensorflow	2.13.0
webencodings	0.5.1	Werkzeug	3.1.3
tensorflow-tensorboard	1.5.1		

A comprehensive environment list is documented in requirements.txt in the code repository (Amundsen, E. 2025). Most of the source code was compatible with the upgraded libraries. Only calls to tensorflow in main.py needed to be updated to use "tf.compat.v1."

Data

Several steps were done to gather data from the Surveillance, Epidemiology, and End Results (SEER) (National Cancer Institute 2025d) for replication of this experiment. First, permission to access the data had to be requested and approved. Second, the SEER*Stat program was downloaded and setup to connect to the database. Next, the database query was configured to select the proper data. Lastly, the data files were converted to a formatted usable by the existing source code.

Data Access. Approval for data access must be requested through a form on the U.S. National Cancer Institute website (National Cancer Institute 2025a). Then, the SEER*Stat software must be installed on the PC from the SEER*Stat Installation → Download SEER*Stat section (National Cancer Institute 2025b). Finally, following the steps provided in the approval email, a log-in and password can be set up to use SEER*Stat. Finally, following the steps provided in the approval email, a log-in and password can be set up to utilize SEER*Stat to access the SEER database.

Data Download. Ideally, the author-provided cohort selection files (Breast_2004_2009_275167.ss, Lung_2004_2009_229011.ss) (Hegselmann, S. 2018) would have been used to gather data from SEER*Stat but they failed to load due to missing tables and data fields. Following the author's description of cohort selection, which used a SEER tutorial (National Cancer Institute 2025c) with 2 minor modifications: a cancer type filter ('Breast', 'Lung and Bronchus') and year of diagnosis range ('2004', '2005', '2006', '2007', '2008', '2009'), also failed to produced data in the format expected by the source code. It resulted in calculations of survivability in different time periods instead of a list of cases and their features. Therefore, a new case listing query on a different database was required to gather data for replication of this study.

The database "Incidence - SEER Research Data, 17 Registries, Nov 2023 Sub (2000-2021)" replaced the original database "Incidence - SEER 18 Regs Research Data + Hurricane Katrina Impacted Louisiana Cases, Nov 2016 Sub (2000-2014) <Katrina/Rita Population Adjustment>". Inquires with the SEER*Stat team confirmed the original database was unavailable as only the last 3 years of databases are retained. The databases have two main differences: data was dropped from the Detroit Metropolitan site and all data from Louisiana was included. This resulted in different case counts for both cancer types: 263,060 Breast cancer cases (95.6% of 275,167) and 216,655 Lung and Bronchus cancer cases (94.6% of 229,011).

New cohort files (Breast_noDetroit_allLA_ASCII-out.ss, Lung_noDetroit_allLA_ASCII-out.ss) (Amundsen, E. 2025) were created from a case listings template, configured to select the proper variables from the database, and used to generate two matrices in SEER*Stat. Each matrix was converted to ASCII format by highlighting all columns and selecting "Matrix" → "Display As" → "Unformatted". Then exported into three files (*.txt, *.dic, *.sas) using "Matrix" → "Export" → "Results as Text File" into the data folders.

Data Descriptions. Since loading the study's cohort files failed, the 2016 Data Record Description (National Cancer Institute 2016) and the 2023 Data Record Description (National Cancer Institute 2023) were used to produce a variable feature map, feature_mapping_table.pdf, between the two databases. Variable names, descriptions, and value sets were compared to best replicate the original data selection. This map was used when configuring the new SEER*Stat cohort files.

Of the original 133 variables used in the study, 107 were available in SEER*Stat, 10 had similar meaning without matching value sets, 2 variables were added as replacements, and 16 were dropped. Originally, 25 variables were gathered but then excluded from the models for various reasons. These were also excluded from the replicated experiment. The full feature mapping of all variables with notes for excluded/dropped/modified/added variables, feature_mapping_table.pdf, can be found in GitHub (Amundsen, E. 2025). Detailed variable descriptions can be found in the 2023 Data Record Description (National Cancer Institute 2023).

Data Pre-Processing. Once the data was downloaded from the SEER*Stat program, several changes were required to make it compatible with the original source code due to file format changes and differences in the selected variables. Data pre-processing scripts and results are on GitHub (Amundsen, E. 2025). First, script sas_matrix_formatter.py was written to do two steps, reformat the *.sas variable definition file into its historical format and convert the *.txt data file into two files, CASES.csv and INCIDENCES.txt. (See Extension section for details) The three output files were copied to the relevant data folders for use by the experiment source code. Next, revisions were made to pipeline.py and data.py to implement the variable feature mapping (see Data Description section for details). Finally, the checksum in seer.py was updated to reflect the change in data line length caused by the different number of features.

LLM Data Pre-Processing. The Large Language Model (LLM) Gemini 2.0 Flash (Google 2025) was asked to help with data pre-processing in two ways. First, an analysis of compatible features between the 2016 and 2023 datasets. Second, a script to convert the *.sas file into it's historic formats for use in existing source code.

For the analysis, the LLM was asked to compare the 2016 Data Record Description (National Cancer Institute 2016) to the 2023 Data Record Description (National Cancer Institute 2023) for the variables listed in pipeline.py (Amundsen, E. 2025), which was required conversion to pipeline.txt for the LLM. The LLM failed to return a complete analysis of compatible features due to file size limitations, only returning a comparison on 20 of the 133 variables.

For writing the script, the LLM was asked to write a script that would convert the SEER*Stat exported file, matrix.sas, to the same format as the author provided file, read.seer.research.nov2016.sas. Both were required to be converted to text files for the LLM. The LLM failed the first 3 attempts by improperly reading the historic format, resulting in an empty format array and no output.

Model

This project replicates the biomedical machine learning experiment in the paper *Reproducible Survival Prediction with SEER Cancer Data* (Hegselmann et al. 2018). The author provided source code for all models it tested in a code repository on GitHub (Hegselmann, S. 2018).

Model Descriptions.

Two feature configurations and four model types were combined to run six models for the experiment: Dummy Classifier (BASE), Logistic Regression (LOG REG), Logistic Regression with 1-n encoded inputs (LOG REG 1-N ENC), Multilayer Perceptron (MLP), Multilayer Perceptron with 1-n encoded inputs (MLP 1-N ENC), and Embedded Multilayer Perceptron with 1-n encoded inputs (MLPEmb 1-N ENC). All models required training through hyperparameter tuning since no pre-trained models or hyperparameter values were provided.

Input Features. Feature choice was done in two different ways. First, all attributes were kept as continuous numbers and normalized by mean and variance to create continuous inputs. Second, a mix of normalized, continuous inputs and 1-n encoded inputs was used to account for categorical or non-interval attributes or those with special codes. In this second method, an attribute would be occasionally split into two inputs, a continuous input to represent the value and a 1-n encoding to indicate if the input is a special code.

Output Target Generation. Target label generation created two series of Boolean indicators (1 or 0) to represent whether the patient had survived to the 1-year or 5-year mark. The algorithm compared each case's survival month attribute to the 12 or 60 month value to determine True or False for death by the 1-year or 5-year mark.

Baseline Model. The baseline model uses Scikit-learn's Dummy-Classifier with strategy='most_frequent' (scikit-learn developers 2025a). It predicts using the most frequent class label in the observed targets.

Logistic Regression Model. The logistic regression model was implemented using Scikit-learn's LogisticRegression (scikit-learn developers 2025b) with adjustable regularization parameter, C. It functions as a binary classifier by predicting the probability of a positive class using a logistic function.

Multilayer Perceptron Model. The multilayer perceptron model uses Kera's Sequential class (Keras Team 2025b) to group a linear stack of layers into a model. Specifically, the author implements one densely-connected NN layer with relu activation and one dropout layer at each depth then finishes it with a densely-connected NN layer with sigmoid activation. There are three adjustable parameters: depth, layer width, and dropout probability. The number of epochs is also configurable.

Embedded Multilayer Perceptron Model. The embedded multilayer perceptron model uses Kera's Model class (Keras Team 2025a) to customize a multilayered model. Specifically, the author implemented one densely-connected NN layer with relu activation and one dropout layer at each depth, embedded with a concatenated and flattened 1D conventional neural network, then finishes it with a densely-connected NN layer with sigmoid activation. There are four adjustable parameters: depth, layer width, dropout probability, and number of neurons in embeddings. The number of epochs is also configurable.

LLM Model Implementation. As the source code was provided by the author, the LLM Genimi 2.0 Flash (Google 2025) was used to better understand the models and review other possible implementations. The LLM provided code that used scikit-learn (sklearn) for model training, data preprocessing, and evaluation. While the authors used sklearn for a majority of their models and evaluation, they also used Keras models for their MLPs. For the best chance at replicability, the author's source code was used for this project.

Training

Model training was performed through hyperparameter tuning. For this, the data was randomly split into training, validation, and testing sets using the ratios 80%, 10%, and 10%. Tuning results were reported on the validation dataset.

Script model_tuner.py runs a batch of experiments by iterating over the combinations of hyperparameters for the user-inputted model, cancer type, and survival duration. Script collect_cluster_results_csv.py was run to collect the results into a *.csv file. This script was modified to write the results to file using the author's original script as a base. Tuning scripts and results are on GitHub (Amundsen, E. 2025).

The sum of evaluation metrics, Area Under the Receiver Operating Characteristic Curve (AUC) and F1 score (F1), was used to determine the best model of each type and those models' hyperparameters were used for the final analysis.

Hyperparameter Tuning. In this study, hyperparameter values could vary between data set, model types, and target survival duration resulting in 20 different sets of parameters to determine. The baseline model did not require tuning.

LOG REG, LOG REG 1-N ENC	C	$1x10^x$ for x = [-2, -1, ..., 10]
MLP, MLP 1-N ENC	depth	[1, 2, 3, 4]
	width	[20, 50, 100, 200]
	dropout	[0.0, 0.1, ..., 0.5]
	epochs	[20, 50, 100]
MLPEmb 1-N ENC	depth	[1, 2, 3, 4]
	width	[20, 50, 100, 200]
	dropout	[0.0, 0.1, ..., 0.5]
	epochs	[20, 50, 100]
	emb nodes	[3, 5, 10]

Table 1: Tuning Parameters by Model.

Computational Requirements. All hyperparameter tuning, model training, and model testing was performed on a Windows 11 PC with an 13th Gen Intel Core i7-13700K processor, 32GB of RAM, and an NVMe SSD.

Runtime Analysis. Hyperparameter tuning took the most computation time due to the high number of combinations tested. 20 batch processes (1 for each tunable model/cancer type/survival duration combination) were run 3 to 5 at a time using the model_tuner.py script. The table below shows the total run time for all runs by model and the average run time for each hyperparameter combination.

		Run Times	
Model	Runs	Total (hh:mm)	Avg (min)
LOG REG	52	00:19	0.36
LOG REG 1-N ENC	52	00:48	0.80
MLP	1152	118:09	6.15
MLP 1-N ENC	1152	146:04	7.52
MLPEmb 1-N ENC*	162	38:05	14.10

Table 2: Run Times by Model.

*Exhaustive hyperparameter tuning of Embedded Multilayer Perceptron 1-N Enc models would originally require 3,456 runs. At approximately 14 minutes per run, it would have required 34 days to complete. So, this model was trained using a subset of the possible combinations as determined by the hyperparameter tuning for the Multilayer Perceptron 1-N Enc models.

		Breast Cancer		Lung Cancer	
Model, Parameters		1-yr	5-yr	1-yr	5-yr
LOG REG	C	0.1	0.01	$1x10^{4}$	$1x10^{10}$
LOG REG 1-N ENC	C	0.1	0.1	0.1	1.0
MLP	depth	1	1	3	3
	width	100	100	100	100
	dropout	0.2	0.2	0.2	0.5
	epochs	20	20	100	100
MLP 1-N ENC	depth	1	1	3	4
	width	20	20	20	20
	dropout	0.2	0.2	0.3	0.5
	epochs	20	20	100	100
MLPEmb 1-N ENC	depth	2	1	3	4
	width	50	20	20	20
	dropout	0.3	0.4	0.2	0.4
	epochs	20	20	100	50
	neurons	3	10	10	3

Table 3: Selected Tuning Parameters by Model.

Loss Functions. Each model perform predictions slightly differently. The Base model simply returns the most frequent class without calculating loss with a loss function. The Logistic Regress models' default algorithm 'lbfgs' uses L2 regularization and log loss for its loss function, which measures the weighted difference between the predicted probability and the true class label. Similarly, the Multilayer Perceptron and Embedded Multilayer Perceptron models also uses binary cross-entropy but with Adam optimization, a type of stochastic gradient descent.

LLM Model Training. The LLM Gemini 2.0 Flash (Google 2025) was used to create a script that runs a batch of python commands iterating through the two datasets, five tuneable models, and all hyperparameter tuning values using inline parameters. The prompt needed to be adjusted multiple times to account for dependencies between the parameters. Ultimately, the script was a good base but needed to be separated and parameterized so that it could be run on multiple processes due to runtime requirements.

Evaluation

After tuning was completed, the best parameters were input into the script run_final_models.py, which used the "–test" inline parameter to run the models on the test dataset. The script collect_cluster_results_csv.py was run again to collect the final model results into a *.csv file. Final model execution and evaluations scripts and results are on GitHub (Amundsen, E. 2025).

Model performance was calculated using 3 metrics, Area Under the Receiver Operating Characteristic Curve (AUC), F1 score (F1), and Accuracy (ACC), and documented in a table for each cancer type for all 6 model types for 1-year and 5-year survival targets.

Further analysis was performed on input attribute importance. The top ten attributes were evaluated for each cancer type using a comparison of weights for logistic regression and ablation analysis for MLPs to determine the importance of these attributes in cancer survival prediction. These results were presented in a set of bar charts, one for each cancer type and survival target, generated by the script attribute_importance_graphs.py.

LLM Evaluation Metrics. The LLM Gemini 2.0 Flash (Google 2025) was asked to analyze the metrics and evaluations performed by the research paper. It accurately outlined the testing metrics of AUC, F1, and ACC (accuracy) but was not able to understand the attribute importance evaluation. The script it wrote to parse the testing results folder structure failed to find the results text files. Ultimately, the results were parsed by collect_cluster_results_csv.py and a custom graphing script, attribute_importance_graphs.py.

Results

The main hypothesis of this paper revolved around replicability of biomedical machine learning research studies. It strived to achieve this goal through data cohort selection documentation and source code publication. Even with these methodology improvements, the results of this replication did not match the author's results in either model performance or attribute importance analysis. Model performance comparisons were done using AUC and F1 scores. Attribute importance was calculated as the absolute value of the logistic regression weights (or sum of absolute value of 1-n encoded features' weights) and through ablation analysis for MLP models.

Model Performance - Breast Cancer

All models predicting survival from breast cancer at the 1-year mark performed better than the original study. Results

were nearly as good for predicting survival from breast cancer at the 5-year mark, as 16 out of 18 of the models performed better than the original study. All values are very similar to each other making these improvements marginal. This replication also shows that 1-n encoding is always an improvement over continuous values, which is the same results as the original study.

Interestingly, none of the best models matched. The researchers found logistic regression with 1-n encoding performed best in AUC and F-1 for 1-year predictions and F1 for 5-year predictions with embedded multilayer perceptron with 1-n encoding performing best in AUC for 5-year prediction. This replication found that embedded multilayer perceptron with 1-n encoding performed best in both AUC and F1 for 1-year predictions while regular multilayer perceptron with 1-n encoding performed best in both AUC and F1 for 5-year predictions.

The difference in scores and model selection are most likely due to the different data cohorts, features selected, and hyperparameter values used in model training and testing.

	1-yr survival		
Model	AUC	F1	ACC
BASE	0.5	0.9853	0.9709
LOG REG	0.9434	0.9869	0.9743
LOG REG 1-N ENC	0.9506	0.9875	0.9755
MLP	0.9473	0.9871	0.9748
MLP 1-N ENC	0.9483	0.9877	0.9759
MLPEmb 1-N ENC	**0.9509**	**0.9877**	0.9759

Table 4: Survival Prediction Performance, Breast Cancer, 1 Year.

	5-yr survival		
Model	AUC	F1	ACC
BASE	0.5	0.9362	0.8800
LOG REG	0.8789	0.9509	0.9107
LOG REG 1-N ENC	0.9025	0.9549	0.9185
MLP	0.9005	0.9547	0.9179
MLP 1-N ENC	**0.9041**	**0.9552**	0.9188
MLPEmb 1-N ENC	0.9010	0.9548	0.9182

Table 5: Survival Prediction Performance, Breast Cancer, 5 Years.

Model Performance - Lung and Bronchus Cancer

Only 8 of the 18 models predicting survival from lung and bronchus cancer at the 1-year mark performed better than the original study. But, all models predicting survival from lung and bronchus cancer at the 5-year mark performed better than the original study. While all values are similar, the 5-year survival prediction values are different enough to indicate significance in these improvements. This replication also shows that 1-n encoding is always an improvement over continuous values, which is the same results as the original study.

Interestingly, three of the best models matched. The researchers found embedded multilayer perceptron with 1-n encoding performed best in AUC and F-1 for 1-year predictions and F1 for 5-year predictions with logistic regression with 1-n encoding performing best in AUC for 5-year prediction. This replication found that embedded multilayer perceptron with 1-n encoding performed best in both AUC and F1 for 1-year predictions, logistic regression with 1-n encoding performed best in AUC for 5-year predictions, and regular multilayer perceptron with 1-n encoding performed best in F1 for 5-year predictions.

The difference in scores and model selection are most likely due to the different data cohorts, features selected, and hyperparameter values used in model training and testing.

	1-yr survival		
Model	AUC	F1	ACC
BASE	0.5	0.0	0.5594
LOG REG	0.8201	0.6968	0.7561
LOG REG 1-N ENC	0.8414	0.7171	0.7691
MLP	0.8382	0.7156	0.7669
MLP 1-N ENC	0.8424	0.7254	0.7683
MLPEmb 1-N ENC	**0.8469**	**0.7282**	0.7730

Table 6: Survival Prediction Performance, Lung Cancer, 1 Year.

	5-yr survival		
Model	AUC	F1	ACC
BASE	0.5	0.0	0.8449
LOG REG	0.8965	0.6287	0.8968
LOG REG 1-N ENC	**0.9125**	0.6509	0.9031
MLP	0.9060	0.6501	0.8973
MLP 1-N ENC	0.9097	**0.6720**	0.9013
MLPEmb 1-N ENC	0.9118	0.6653	0.9000

Table 7: Survival Prediction Performance, Lung Cancer, 5 Years.

Attribute Importance - Breast Cancer

The top 10 attributes were determined based on the summed importance across all models for breast cancer survival prediction at the 1-year and 5-year marks. 5 of the 10 features (ER Status Recode and Adjusted AJCC M, T, N, and Stage) were the same between this replication and the original study for both prediction time frames. 3 of the 10 features were very similar. Median Household Income and Rural-Urban Code were replacements for the unavailable State-Country Recode feature that ranked very high in the original study. And, SEER Combined Summary Stage is nearly the same as the Historic SSG 2000 Stage feature from the original study. In the 1-year prediction, Insurance Recode is not listed since that feature was dropped from the 2023 database. In the 5-year prediction, 2 Histologic features gained importance over the original study.

Despite nearly 20% of the attributes being different, 90% of the 1-year prediction features and 80% of the 5-year prediction features were similar enough to match the original

study. This shows that the AJCC codes, staging, and specific breast cancer attributes are very important to breast cancer survival predictions.

Top 10 Attributes, Breast Cancer at 1 Year

1. Median household income inflation adj to 2022
2. Rural-Urban Continuum Code
3. Breast - Adjusted AJCC 6th Stage (1988-2015)
4. Breast - Adjusted AJCC 6th M (1988-2015)
5. Breast - Adjusted AJCC 6th T (1988-2015)
6. Breast - Adjusted AJCC 6th N (1988-2015)
7. SEER Combined Summary Stage 2000 (2004-2017)
8. ER Status Recode Breast Cancer (1990+)
9. CS version input original (2004-2015)
10. PR Status Recode Breast Cancer (1990+)

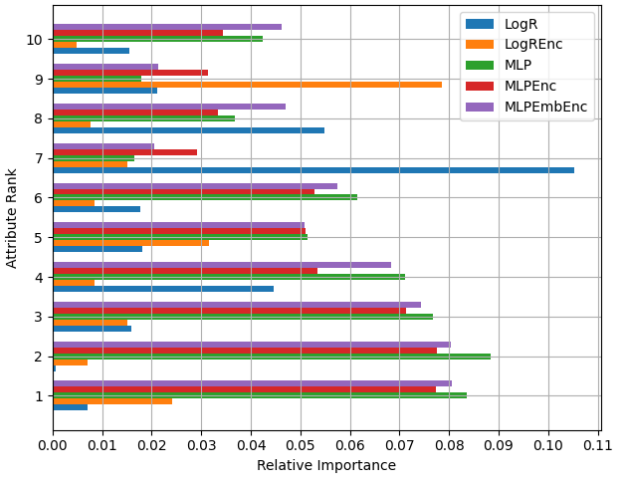

Figure 1: Top 10 Attributes, Breast Cancer at 1 Year

Top 10 Attributes, Breast Cancer at 5 Years

1. Breast - Adjusted AJCC 6th Stage (1988-2015)
2. Breast - Adjusted AJCC 6th M (1988-2015)
3. Breast - Adjusted AJCC 6th T (1988-2015)
4. Breast - Adjusted AJCC 6th N (1988-2015)
5. Median household income inflation adj to 2022
6. SEER Combined Summary Stage 2000 (2004-2017)
7. Rural-Urban Continuum Code
8. Histologic Type ICD-O-3
9. ER Status Recode Breast Cancer (1990+)
10. Histology recode - broad groupings

Attribute Importance - Lung and Bronchus Cancer

The top 10 attributes were determined based on the summed importance across all models for lung and bronchus cancer survival prediction at the 1-year and 5-year marks. 3 of the 10 features (CS Mets at Diagnosis, Derived AJCC M, AYA site recode) were the same between this replication and the

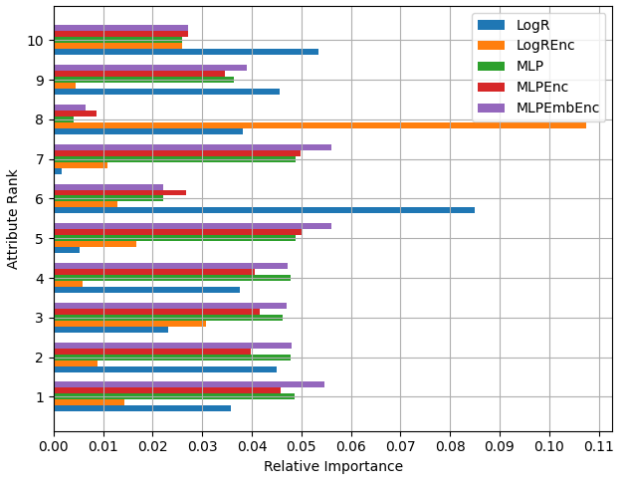

Figure 2: Top 10 Attributes, Breast Cancer at 5 Years

original study for both prediction time frames while Histologic Type ICD-O-3 matched for 1-year predictions and CS Version Input Current matched 5-year predictions. 2 of the 10 features were very similar, ICCC site codes. Despite replacing WHO 2008 codes with IARC 2017 codes, it did not affect their importance since they are still listed in the top 10. For 5-year prediction, SEER Combined Summary Stage is nearly the same as the Historic SSG 2000 Stage feature from the original study. One major difference is that Median Household Income and Rural-Urban Code (replacements for the unavailable State-Country Recode feature) were not ranked as important attributes despite State-Country being ranked very high in the original study.

Unlike the breast cancer survival predictions, the nearly 20% of the attributes being different had a greater affect on lung and bronchus cancer survival predictions. Only 60% of the 1-year prediction features and 70% of the 5-year prediction features were similar enough to match the original study's attribute importance analysis. This shows that while metastases, staging, and histology attributes are very important to lung and bronchus cancer survival predictions, some staging and extension aspects had a higher impact in this replication study than in the original study.

Top 10 Attributes, Lung Cancer at 1 Year

1. CS mets at dx (2004-2015)
2. Derived AJCC M, 6th ed (2004-2015)
3. CS version input original (2004-2015)
4. Histologic Type ICD-O-3
5. ICCC site recode 3rd edition/IARC 2017
6. CS extension (2004-2015)
7. SEER Combined Summary Stage 2000 (2004-2017)
8. ICCC site recode extended 3rd edition/IARC 2017
9. Derived AJCC Stage Group, 6th ed (2004-2015)
10. AYA site recode 2020 Revision

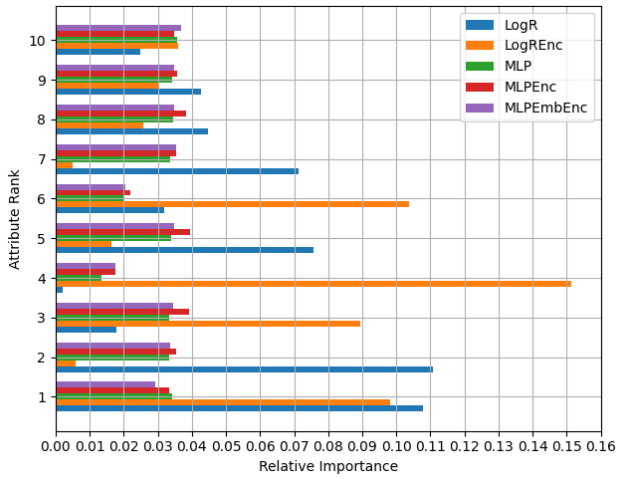

Figure 3: Top 10 Attributes, Lung Cancer at 1 Year

Top 10 Attributes, Lung Cancer at 5 Years

1. CS mets at dx (2004-2015)
2. Histologic Type ICD-O-3
3. Derived AJCC M, 6th ed (2004-2015)
4. SEER Combined Summary Stage 2000 (2004-2017)
5. ICCC site recode extended 3rd edition/IARC 2017
6. CS version input current (2004-2015)
7. CS extension (2004-2015)
8. ICCC site recode 3rd edition/IARC 2017
9. CS version input original (2004-2015)
10. AYA site recode 2020 Revision

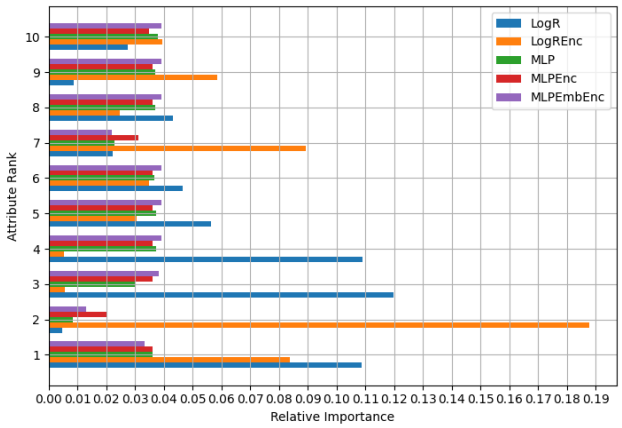

Figure 4: Top 10 Attributes, Lung Cancer at 5 Years

Extensions

LLM Brainstorm. The LLM Gemini 2.0 Flash (Google 2025) was used to brainstorm possible extensions or ablations for this research paper focusing on the areas of datasets, loss functions, models, or evaluations. It came up with many valid ideas particularly in the areas of Dataset Extensions, Loss Function and Model Extensions, and Evaluation Extensions.

In the area of Dataset Extensions, it recommended looking into additional cancer types, analyzing different SEER data versions, refining predictions with cancer subtype analysis, and using data from other sources to validate models. Many of these options are already possible through selecting different data cohorts in the SEER*Stat program. Due to the arduous data pre-processing required to attempt to replicate this paper, analyzing different SEER data versions was selected as an extension for this project.

In the area of Loss Function and Model Extensions, it recommended investigating alternative loss functions like focal loss, weighted loss, and survival-specific types of loss functions, performing systematic attribute analysis to highlight clinical variable contributions, implementing advanced machine learning models such as CNNs, RNNS, SVMs, ensemble models, and explainable AI models, and optimizing hyperparameter tuning. Some of these ideas are already implemented in the research code base, though not mentioned in the paper. Due to the high run times in hyperparameter tuning, integrated hyperparameter tuning was considered as an extension for this project though time constraints did not allow for its implementation.

In the area of Evaluation Extensions, it recommended analyzing calibration, time-dependent AUC, clinical utility, and socioeconomic factors. Due to the high impact of location-based attributes in both the original study and this replication, exploration of socioeconomic factors was selected as an extension for this project.

Implementation - SEER*Stat Data Format. The first extension to the code base was to make it compatible with different SEER data versions. The original study focused on data from the 2016 ASCII database. The goal of this extension was to expand the datasets to allow for data from the SEER*Stat program to also be used. The script sas_matrix_formatter.py was created to read-in the three SEER*Stat output files (*.txt, *.sas, *.dic) and then write the files expected by the original source code (INCIDENCES.txt, CASES.csv, matrix_reformatted.sas). The script's output was verified in two ways. First, by comparing a handwritten *.sas file to the script-written file and ensuring they matched. Second, the first ten lines of the INCIDENCES.txt file was re-written back into the SEER*Stat file format and compared to the original SEER*Stat file to make sure there was no data loss. SEER*Stat formatting scripts and results are on GitHub (Amundsen, E. 2025).

Implementation - Socioeconomic Impact. The second extension to the research study was to analyze the socioeconomic impact on survivability of breast cancer and lung and bronchus cancer. This analysis was inspired by the high attribute importance of two socioeconomic features, Median Household Income and Rural-Urban Code (National Cancer Institute 2025e). The script feature_analysis.py (Amundsen, E. 2025) was written to generate two scatter plots of the raw data with statistics overlaid. The first plot compares Median Household Income vs. Survival Months and the second

compares Rural-Urban Code vs. Survival Months. These charts were verified by comparison to manually created excel graphs.

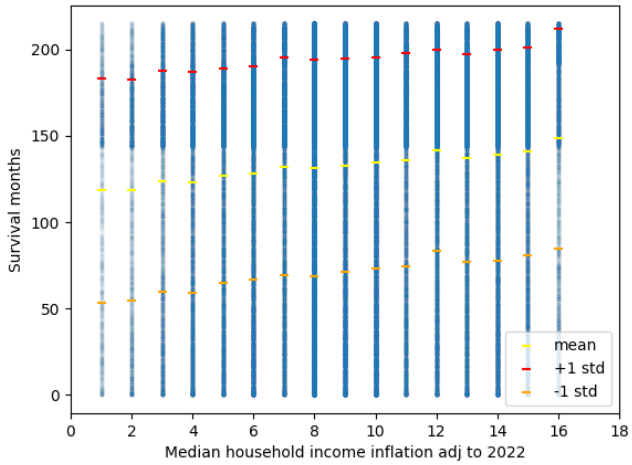

Figure 5: Median Household Income vs. Survival, Breast Cancer.

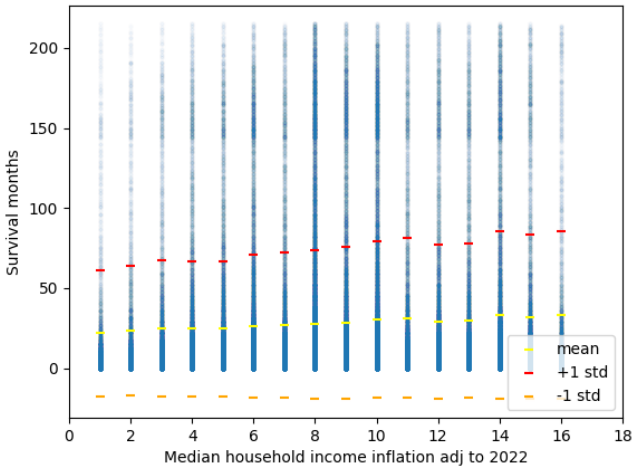

Figure 6: Median Household Income vs. Survival, Lung Cancer.

These charts show that there is almost no correlation between socioeconomic factors and Breast Cancer and Lung and Bronchus Cancer survivability due to the wide range of survival months in every socioeconomic bracket. Median Household Income had an R-value of 0.0791 and 0.0553, respectively, while Rural-Urban Code had an R-value of -0.0325 and -0.0260, respectively.

Discussion

The paper *Reproducible Survival Prediction with SEER Cancer Data* (Hegselmann et al. 2018) was not exactly reproducible as shown by the very similar but not equivalent model performance and attribute importance results. Several factors made it irreproducible, mainly that the data cannot be repeatability extracted. The SEER database is a live database with copies of past databases only being kept for 3 years. So,

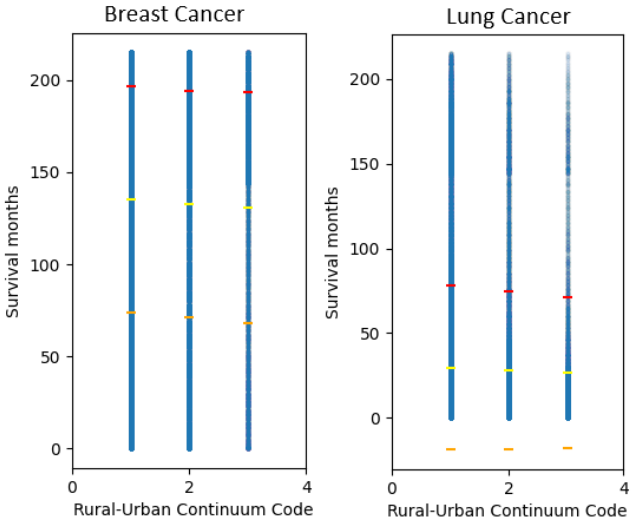

Figure 7: Rural-Urbana Code vs. Survival Months.

a study based on this data source has a limited time frame for repeatability. Within the database, changes to the data include data collection sites being added or removed, cases being added yearly, variables being dropped due to outdated information, variables being recoded/reformatted with different values, variables being renamed so that mapping requires detailed variable description analysis, and new variables being added.

Parts of replicating this paper were fairly straightforward. Access to the SEER*Stat database was easy to acquire through a simple form on their website. Also, the author's published source ran with minimal updating.

There were several difficult aspects to replicating this paper. First, the data cohort selection took three tries to complete since the published cohort files were of an outdated format and the cited tutorial no longer produced a list of cases. It required manual configuration to achieve a similar (though not exact) data cohort. Second, a manual variable feature mapping was required to identify as many similar attributes between the 2016 database used in the original study and the 2023 database currently published, which was done through a detailed variable description comparison. Lastly, while the hyperparameter tuning was not hard, it was very time consuming, requiring 5 days running 3 - 5 batch processes at a time to select the best model parameters.

The biggest area of improvement for reproducibility of this paper is to stabilize the dataset either through publication of past datasets beyond the 3 year storage provided by the SEER*Stat time or through yearly research updates to reflect the changes in the data source. Additionally, a variable feature map tool or yearly documentation that contains all changes between databases would improve the replicability of this paper.

Author Contributions

All work was performed by Elizabeth Amundsen (ecreigh2).

References

Amundsen, E. 2025. mcs598-SEERCancerStudy - Replication of the Paper: Reproducible Survival Prediction with SEER Cancer Data. https://github.com/Kaeldra1304/mcs598-SEERCancerStudy. Accessed: 2025-04-25.

Google. 2025. Gemini Flash 2.0 [Large Language Model]. https://gemini.google.com/app. Accessed: 2025-04-16.

Hegselmann, S.; Gruelich, L.; Varghese, J.; and Dugas, M. 2018. Reproducible Survival Prediction with SEER Cancer Data. *Proceedings of Machine Learning Research*, 85: 49–66.

Hegselmann, S. 2018. MLHC 2018 - Reproducible Survival Prediction with SEER Cancer Data. https://github.com/stefanhgm/MLHC2018-reproducible-survival-seer. Accessed: 2025-03-17.

Keras Team. 2025a. The Model class. https://keras.io/api/models/model/#model-class. Accessed: 2025-04-25.

Keras Team. 2025b. The Sequential class. https://keras.io/api/models/sequential/. Accessed: 2025-04-25.

National Cancer Institute. 2016. SEER RESEARCH DATA RECORD DESCRIPTION - NOV 2016. https://seer.cancer.gov/data-software/documentation/seerstat/nov2016/TextData.FileDescription.pdf. Accessed: 2025-04-16.

National Cancer Institute. 2023. SEER RESEARCH DATA RECORD DESCRIPTION - NOV 2023. https://seer.cancer.gov/data-software/documentation/seerstat/nov2023/TextData.FileDescription-nov2023.pdf. Accessed: 2025-04-16.

National Cancer Institute. 2025a. Request SEER Incidence Data. https://seerdataaccess.cancer.gov/seer-data-access. Accessed: 2025-03-17.

National Cancer Institute. 2025b. SEER*Stat Software. https://seer.cancer.gov/seerstat/. Accessed: 2025-03-17.

National Cancer Institute. 2025c. SEER*Stat Survival Exercise 3: Cause-Specific Survival. https://seer.cancer.gov/seerstat/tutorials/survival3/webprint/. Accessed: 2025-03-17.

National Cancer Institute. 2025d. Surveillance, Epidemiology, and End Results Program. https://seer.cancer.gov/. Accessed: 2025-03-17.

National Cancer Institute. 2025e. Time-dependent County Attributes. https://seer.cancer.gov/seerstat/variables/countyattribs/time-dependent.html. Accessed: 2025-04-25.

Proceedings of Machine Learning Research. 2025a. Reproducible Survival Prediction with SEER Cancer Data. https://proceedings.mlr.press/v85/hegselmann18a/hegselmann18a.pdf. Accessed: 2025-04-25.

Proceedings of Machine Learning Research. 2025b. Reproducible Survival Prediction with SEER Cancer Data, Appendix. https://proceedings.mlr.press/v85/hegselmann18a/hegselmann18a-supp.pdf. Accessed: 2025-04-25.

scikit-learn developers. 2025a. DummyClassifier. https://scikit-learn.org/stable/modules/generated/sklearn.dummy.DummyClassifier.html. Accessed: 2025-04-25.

scikit-learn developers. 2025b. LogisticRegression. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html. Accessed: 2025-04-25.