University of Hertfordshire UH

School of Physics, Engineering and Computer Science

MSc Data Science Project
7PAM2002-0509-2024
Department of Physics, Astronomy and Mathematics

Data Science FINAL PROJECT REPORT

Project Title:

Predicting Survival Time in Lung Cancer Patients using Survival Analysis

Student Name and SRN:

Aghana Kanakkanchery Shanmughan

Student ID: 23103031

Supervisor: Pushp Tiwari (Raj)

Date Submitted: 28/08/2025

Word Count: 4965

GitHub Link: https://github.com/AGHANAKS/Final-Year-Project.git

DECLARATION STATEMENT

This report is submitted in partial fulfilment of the requirement for the degree of Master of Science **in Data Science** at the University of Hertfordshire.

I have read the detailed guidance to students on academic integrity, misconduct and plagiarism information at Assessment Offences and Academic Misconduct and understand the University process of dealing with suspected cases of academic misconduct and the possible penalties, which could include failing the project or course.

I certify that the work submitted is my own and that any material derived or quoted from published or unpublished work of other persons has been duly acknowledged. (Ref. UPR AS/C/6.1, section 7 and UPR AS/C/5, section 3.6)

I did not use human participants in my MSc Project.

I hereby give permission for the report to be made available on module websites provided the source is acknowledged.

Student Name: Aghana Kanakkanchery Shanmughan

Student Name signature: 

Student SRN number: 23103031

UNIVERSITY OF HERTFORDSHIRE

SCHOOL OF PHYSICS, ENGINEERING AND COMPUTER SCIENCE

Abstract

This study demonstrates the prediction of lung cancer survival with the NCCTG dataset to determine accuracy of models through C-index and Brier scores of the models during the analysis. The study considered the following: (1) the effect of gender on lung cancer survival; (2) the effect of assessing the patients themselves (pat_karno) instead of physicians on the predictions if the patient assessment will improve the prediction more than physician assessment; (3) the most significant clinical predictors that would affect lung cancer survival; (4) Cox PH, Random Survival Forest (RSF), and Weibull AFT models regarding prediction of survival. The study found that the effect of gender added another risk with males more hazardous, as well as pat_karno had some improvement on predictions of the RSF model in regards to survival. The ECOG score and age were the strongest predictors among types of models, and Random Survival Forest was the leading predictor compared to the other models. In the conclusion, RSF model was the leading predictor of lung cancer survival information, and pat_karno offered better predictions.

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Overview

Lung cancer is one of the most significant challenges in oncology, accounting for 1.8 million deaths each year and remaining the leading cause of cancer deaths overall (Chaddad et al., 2017). Improved efforts in early detection and targeted therapies have not improved the prognosis of lung cancer, which still has a less than 20% 5-year overall survival. Common approaches utilize the TNM stage and histopathologic classifications, without taking into account the array of interacting clinical, molecular and lifestyle factors (Mohanty et al., 2020). There is a need for new analytic methodologies to model survival risk using multi-dimensional data, directly impacting clinical decision-making and treatment.

Lung cancer biology is complex; additionally, the considerable demand of clinical care has posed challenges for survival estimation models. New evidence shows significant differences based on sex in survivorship - females survive longer than males universally across all four subtypes of lung cancer May et al.,(2023). The degrees of variation in sex-based survivorship may be indicators of biologic differences because of treatment effects, and differences in tumor biology.

1.2 Short Description of Idea

The objective of the study is to model lung cancer survival time with the NCCTG Lung Cancer dataset and examine gender differences, and age. We will also evaluate the Karnofsky and ECOG performance scores, as well as predict survival performance using Cox Proportional Hazards regression, Random Survival Forest, and Accelerated Failure Time (AFT) methods, and review several traditional approaches to survival prognosis in the study. The overall aim is to produce useful prognostic information for the prognostication and clinical management of patients with a lung cancer diagnosis, by a careful accounting of both demographic and clinical characteristics of the patients.

1.3 Research Questions:

- How does gender (sex) influence the survival probability of lung cancer patients?
- Can patient self-assessments (Pat_karno) improve the accuracy of survival time predictions for lung cancer patients compared to physician assessments alone, using survival analysis techniques?
- Which clinical factors (age, sex, ECOG score, etc.) are most statistically significant in predicting lung cancer survival?
- Which survival analysis model (Cox Proportional Hazards, Random Survival Forest, or AFT model) provides the most accurate predictions for lung cancer patient survival, and how do their results compare?

1.4 Research Objectives

- To compare survival probabilities between male and female lung cancer patients using Kaplan-Meier analysis.
- To determine whether patients aged ≥70 years have a significantly different survival rate compared to younger patients.

- To evaluate the impact of physician-assessed performance scores (Karnofsky & ECOG) on survival outcomes.
- To identify the most influential predictors of survival using Cox Proportional Hazards regression.
- To compare the predictive accuracy of traditional survival models, such as Cox-PH with machine learning approaches such as Random Survival Forest in estimating lung cancer survival probabilities.
- To develop survival models such as Cox Proportional Hazards, Random Survival Forests and Accelerated Failure Time to predict survival time using the NCCTG lung cancer dataset.
- To assess the prognostic value of patient self-assessments (Pat_karno) by comparing model performance with and without this feature.
- To evaluate model accuracy using metrics such as the concordance index and Brier score.

1.5 Research Aims

- Utilizes Survival Analysis methods to predict survival time of lung cancer patients.
- Aims to identify clinical factors affecting survival and compare the effectiveness of various survival models.
- Using the Kaplan-Meier survival curves to visually and statistically compare survival probabilities for male and female patients, and for patients aged 70 years or older versus younger patients.
- Uses Cox Proportional Hazards regression model for multivariate analysis to identify statistically significant predictors.
- Compares established methods like Cox Proportional Hazards to advanced machine learning techniques like Random Survival Forests.
- Aims to determine the most applicable survival model for future use in clinical settings for lung cancer prognosis.

1.6 Ethical Issues

- The NCCTG dataset is ethically bound since it is from clinical trials, and respectful use for reuse is needed. Our reuse is accompanied with verification of its licensing term, attribution requirements, and privacy protection, because the consequences of disrespectful use may have potentially violated the ethical protocols of the original clinical trial or breach the data-sharing agreements with the original participants.
- Sensitive health data requires secure handling to prevent breaches, even in de-identified public sets.
- Imbalanced demographics may skew predictions; fairness audits are essential.

1.7 Dataset

The Lung Cancer Dataset of the North Central Cancer Treatment Group (NCCTG) was obtained from a clinical study published in the Journal of Clinical Oncology Loprinzi, et al. (1994). The dataset comes from a prospective cohort study of 228 patients with advanced lung cancer.

Data Collection Method:
Survival Times: Time-to-death or censoring (study dropout).
Clinical Covariates: Age, sex, ECOG performance status (physician-assessed), Karnofsky scores (physician-and patient-assessed).

Assessment Tools:
ECOG: 0 (fully active) to 5 (dead) 1.
Karnofsky: 0 (dead) to 100 (completely healthy)

Key variables
ID: A unique identifier assigned to each patient.
TIME: This variable records the survival time in days, denoting the duration from the start of the observation period to either the patient's death or the point of censoring.
Y: This binary variable indicates the censoring status, with codes typically representing 1 for censored and 2 for dead (or 0 for censored/start of observation and 1 for death).
Age: The patient's age at the time of study entry, recorded in years.
Sex: The biological sex of the patient, categorized as F for female and M for male.
ECOGPH: The ECOG (Eastern Cooperative Oncology Group) performance status, as assessed by the physician. This categorical scale typically ranges from 0 (fully active) to 4 (bedbound), or sometimes 0 to 5 (dead).
KarnoPH: The Karnofsky performance status, a continuous scale from 0 (dead) to 100 (completely healthy), as rated by the physician.
KarnoPAT: The Karnofsky performance status, as assessed by the patient.

Source: https://monolixsuite.slp-software.com/monolix/2024R1/ncctg-lung-cancer-data-set

2. Literature Review

2.1 Survival Probability, Censoring, and Hazard Functions

Survival analysis uses time-to-event data to analyze survival probability, censoring, and the hazard function. The survival function S(t), or the probability that a given individual is alive or event-free beyond time t (Clark et al., 2003). Censoring, the case where an event is never observed for all individuals, can take many forms, with right censoring being the most common. The hazard function represents the instantaneous risk of an event occurring at time t conditionally on the subject surviving to time t.

2.2 Clinical Factors Influencing Lung Cancer Patient Survival

2.2.1 Gender and Survival Probability

Research indicates that lung cancer survival differs by gender; women are more likely to survive surgery compared to men (Clark et al., 2003; May et al., 2023). The difference in outcomes is accounted for by extraneous factors, differences in smoking, and biological differences (Extermann & Wedding, 2012). Males are at greater risk and diagnosed at a more advanced disease stage, while females do appear to have greater antitumor activity among their tumour-associated macrophages (May et al., 2023).

2.2.2 Impact of Age on Survival Duration

About lung cancer survival, both age-related factors are also certainly complex and paradoxical. Older patients often face comorbidities, weaker organ function, poor nutrition,

and reduced immunity, all of which worsen survival. If you were to simply randomize an older population, you might find that they may do worse (Corso et al., 2015).

2.2.3 Influence of Physician-Assessed Performance Scores on Survival

In oncology, physician-rated performance status scales such as the Karnofsky Performance Score and Eastern Cooperative Oncology Group (ECOG) Performance Status are helpful in providing an assessment of a patient's ability to perform activities necessary for daily living, self-care, and work. These performance status scales also record restrictions on activity levels and in studies, patients with low performance status have worse overall survival. Performance status scores show a strong correlation to prognostic factors of overall survivorship.

2.3 Existing works

Al Mamlook et al. (2020) used Cox regression and Kaplan-Meier to compare the two scoring systems (KPS and ECOG) on 228 patients in NCCTG lung cancer data set. They answered that patients with more scores of deaths have more risks to die, but calorie intake is not a good factor for survival. The analysis revealed that KPS was less perceptive, whereas ECOG offered excellent information to the doctors to plan treatment.

Perera and Dwivedi (2020) emphasised the usefulness of Weibull survival models in cancer research, noting their ability to capture different hazard functions more effectively than standard approaches. Their work shows that parametric models can identify meaningful clinical predictors and handle complex survival data. This supports applying Weibull models, alongside Cox regression and Random Survival Forests, to improve lung cancer survival analysis.

Chen et al. (2022) predicted progression-free survival in small-cell lung cancer (SCLC) using CT-based radiomic features. From 186 cases, 1,218 features were extracted and reduced to 11 key predictors with machine learning. Using Random Survival Forests, their model achieved a C-index of 0.7 and mean C/D AUC of 0.8 on the independent test set, and performed better than single clinical features and radiomic features computed from lung or mediastinal window.

Germer et al. (2024) compare the performance of Cox regression and novel machine learning methods with lung cancer data from the Schleswig-Holstein Cancer Registry. Four models were compared: (CoxPH), (RSF), and two NN architectures from DeepSurv and TabNet approaches. The authors used concordance index, Brier score, and AUC-ROC score to evaluate system performance. The best performing model was CoxPH, while the RSF was the best performing model (C-I 0.7) based on a dataset including tumour size, lymph node, and metastasis statuses.

Gencer (2025) compared Cox Proportional Hazards and Accelerated Failure Time models for lung cancer survival, focusing on cell type, prior therapies, and treatments. Using the Akaike Information Criterion (AIC), the study found that small cell type showed the poorest survival, while AFT models, particularly the Weibull AFT, provided the best fit. Patients with previous therapy and treatment had higher survival probabilities.

3. Methodology & Implementation

3.1 Data Preparation

3.1.1 Checking for missing values

Before assessing a dataset, a preliminary systematic check for missing values following a pre-determined protocol is essential; missing data can produce an important bias, compromise an important inference from statistical analysis. By checking out the missing values in dataset we have found no missing values in dataset.

3.1.2 Conversion of categorical values into numerical

Categorical variables in the dataset, such as gender (sex), must be laboriously converted into numerical values, to be useful for quantitative modelling purposes. For this we have shifts textual labels ('M'/'F') to a binary integer format (1 and 0, for male and female respectively).

3.1.3 Creating a dead column from status (0=dead, 1=alive)

In the research the status column of the dataset originally had patient outcomes coded as either censored (0) or dead (1). To make the event of interest, death, explicit for the sake of survival modelling, we created a new binary column, dead.

3.2 Correlation Analysis

3.2.1 Heatmap correlation

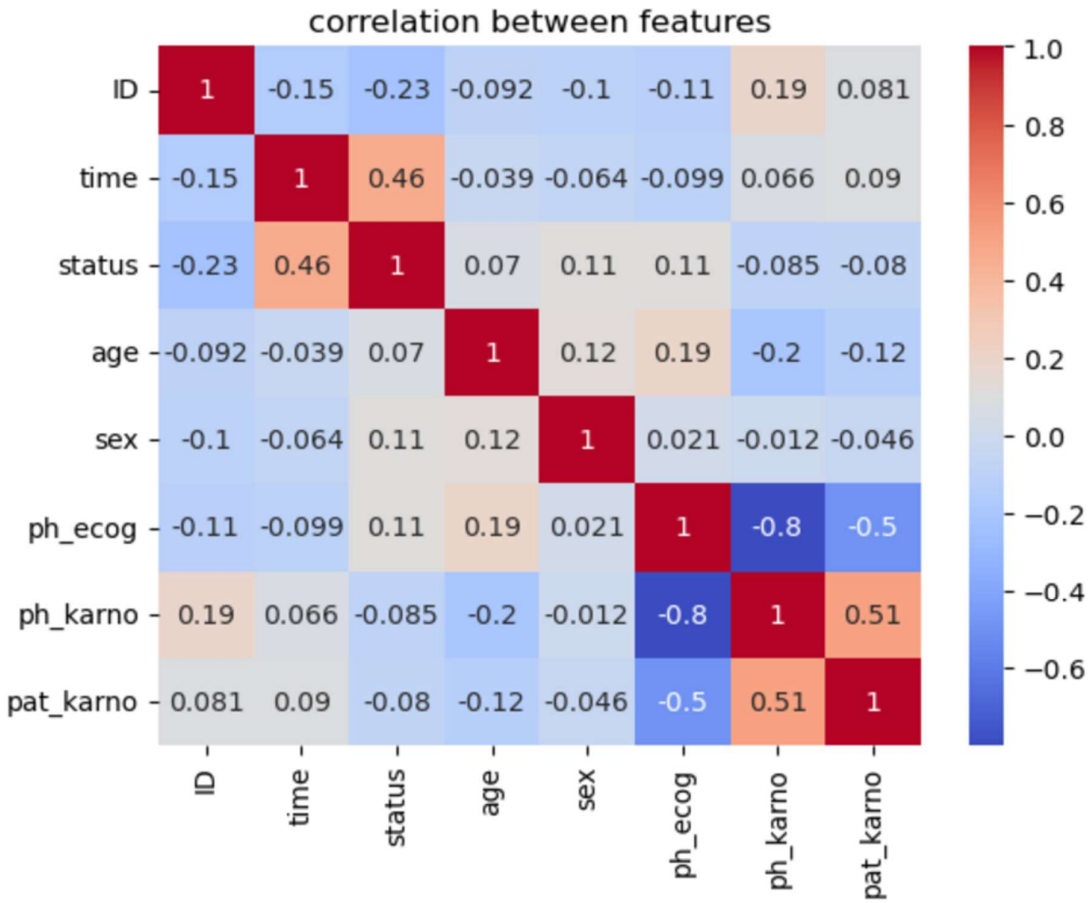


Figure 1: Heatmap Correlation analysis

The heatmap indicated that ph_ecog and ph_karno had a high negative correlation (-0.80); this reflects inverse relationship. A moderate positive correlation was found between ph_karno and pat_karno (0.51), while survival time had a moderate correlation with vital status (0.46). Age and gender had weak correlations with the other predictors, so only weakly posed linear relationships exist; we will examine the differences in age using Kaplan-Meier, and there may be some non-linear relationship with age.

3.3 Survival Analysis Fundamentals

3.3.1 Kaplan-Meier Estimation

The Kaplan-Meier estimator is a non-parametric statistical estimator for the survival (probability of an individual living) from lifetime data. Here, the KM model is used to estimate the proportion of patients living for a defined amount of time, following treatment or diagnosis of the cancer.

Kaplan-Meier Equation

$$\mathbf{S(t_i) = S(t_{i-1}) * \left(1 - \frac{d_i}{n_i}\right)}$$

3.4.2 Survival probability with time

Timeline	KM_Estimate
0.0	1.000000
5.0	0.995614
11.0	0.982456
12.0	0.978070
13.0	0.969298

Table 1: Survival probability with time

The Kaplan-Meier survival analysis displays a high initial survival probability (1.000 at baseline) and does not wane until 5 months (0.9956), suggesting an effective intervention in the early time frame. After 5 months, there is a steady decline with the probabilities decreasing to (11 months) 0.9825, (12 months) 0.9781, and (13 months) 0.9693. The continual decrease emphasizes the time-dependent mortality burden of lung cancer- in this analysis, there was no evident plateau at later stages beyond 11 months- which indicates the important windows for clinical intervention.

3.4.3 Survival Probability with Confidence Intervals

	KM_estimate_lower_0.95	KM_estimate_upper_0.95
0.0	1.000000	1.000000
5.0	0.969277	0.999381
11.0	0.953935	0.993379
12.0	0.948120	0.990813
13.0	0.936682	0.985244
...	...	...
840.0	0.030728	0.123060
883.0	0.017866	0.108662
965.0	0.017866	0.108662
1010.0	0.017866	0.108662
1022.0	0.017866	0.108662

Table 2: Survival probability with confidence intervals

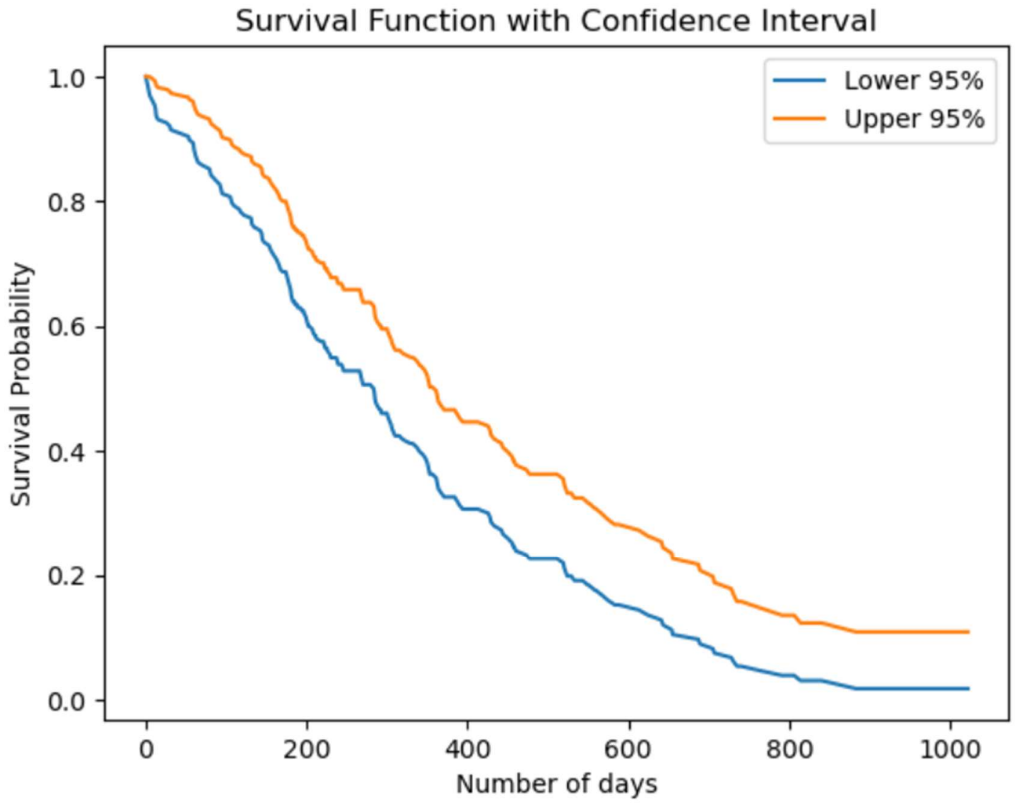


Figure 2: Survival function with confidence interval

At baseline, the survival probability was 1.00 (95% CI: 1.00-1.00). Early survival was high (5-day probability: 95% CI 0.969-0.999), but gradually declined, particularly by day 13 (95% CI: 0.937-0.985). By day 840, survival probability dropped sharply (95% CI: 0.031-0.123), and while it seemed to stabilize at unacceptably low levels by the end of our study (day 1,022; 95% CI: 0.018-0.109), the width of the CIs over time (e.g., day 5 CI width: 0.030 vs. day 840 width:

0.092) continued to widen and increase uncertainty due to relatively fewer at-risk patients remaining, as was anticipated from the limitations in understanding longitudinal cancer survival data.

3.4 Calculating the median time to an event

Timeline	KM_ Estimate-conditional median duration remaining to event
0	310
5	305
11	309
12	308
13	316
…	…
840	Inf
883	Inf
965	Inf
1010	Inf
1022	Inf

Table 3: Median time to an event

This table offers time updated predictions for patients who have already survived to a designated time interval (e.g., for patients who survived to 310 days at diagnosis). If a patient survives 13 days, their remaining median survival is estimated to be 316 days and their prognosis has improved with exceptionally short-term survival - this information is useful for adjusting feasibility of follow-up and for designating natural subgroups in studies of biological resilience.

3.5 Estimating Hazard Rates using Nelson-Aalen

Timeline	**NA_ Estimate**
0.0	0.000000
5.0	0.004386
11.0	0.017660
12.0	0.022125
13.0	0.031114
...	...
840.0	2.641565
883.0	2.891565
965.0	2.891565
1010.0	2.891565
1022.0	2.891565

Table 4: Hazard Rates using Nelson-Aalen

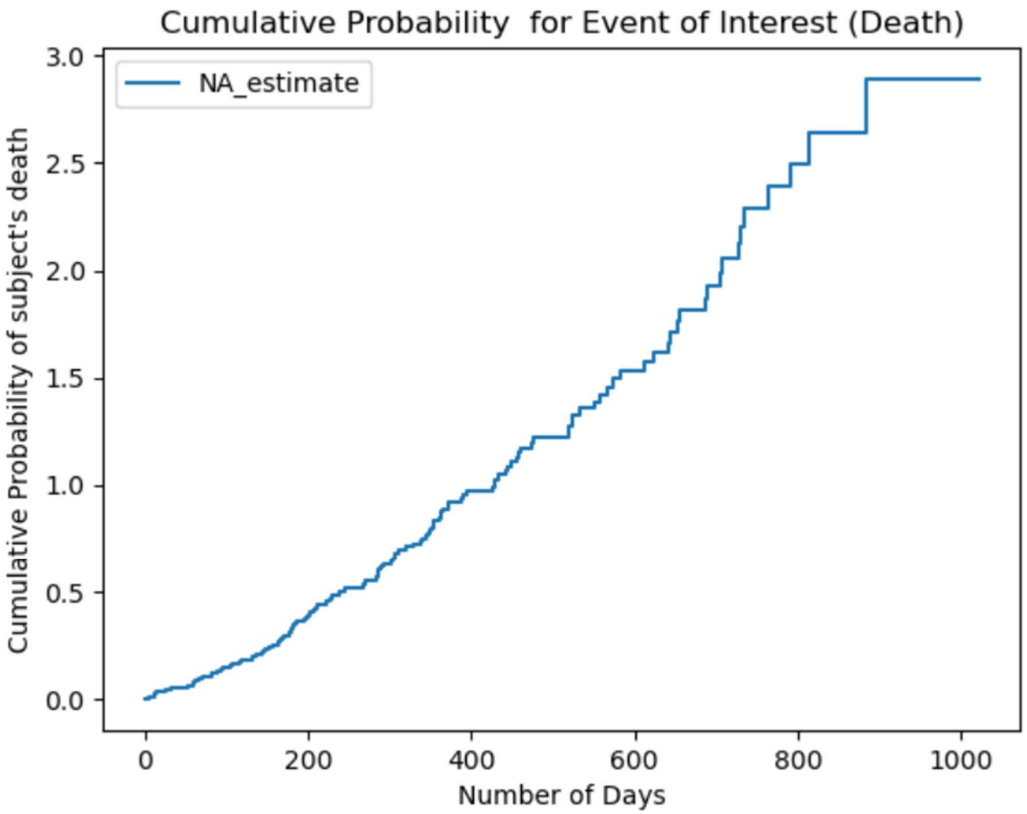


Figure 3: Cumulative probability of event of interest (Death)

The Nelson-Aalen estimator was used to assess cumulative hazard rates for lung cancer patients, showing clear early risk dynamics. The cumulative hazards increased between 0 and 0.022 at 12 months and 0.031 at 13 months, demonstrating a consistent pattern of risk of early mortality. After month 10, the estimate reached 2.89, indicating risk levels had stabilized for long-term patients.

3.6 Comparison of the different groups of the attributes using the Kaplan Meier Curve

3.6.1 Division of two groups by sex

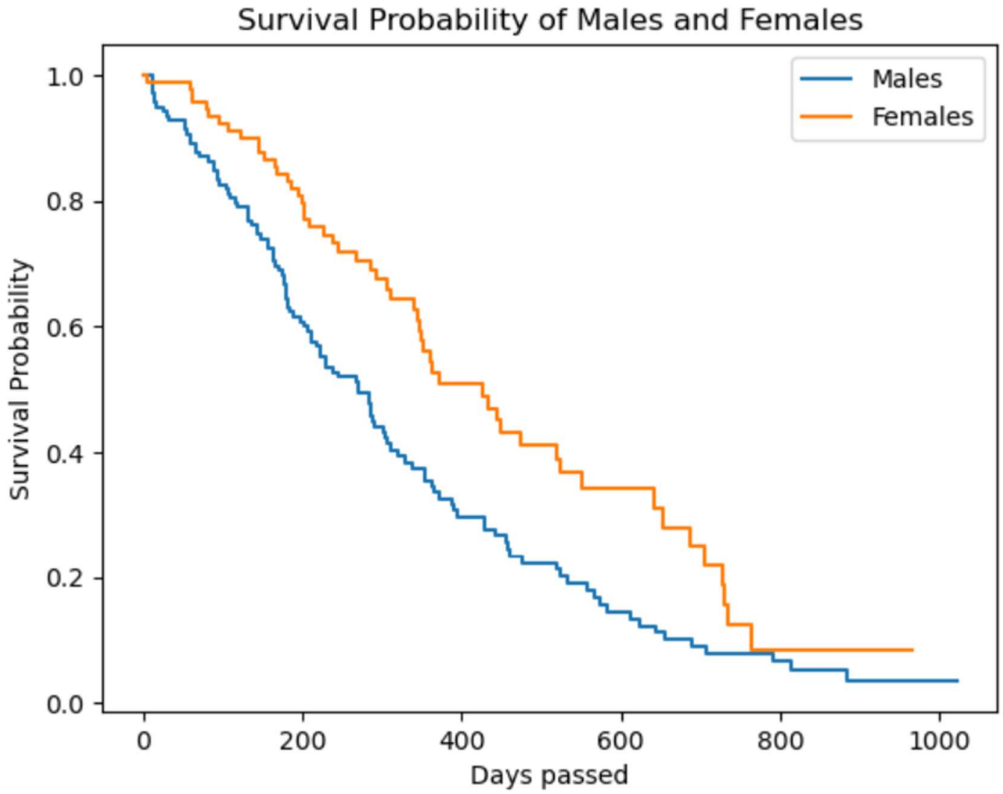


Figure 4: Survival probability of males and females

Patients were stratified by gender (male=1, female=0) and analyzed using Kaplan-Meier curves with the log-rank test (p=0.0013). Results showed males had higher early mortality, while females survived longer. This significant difference highlights gender as an important factor in lung cancer outcomes and is essential for predictive survival models.

3.6.2 Division of two groups by age

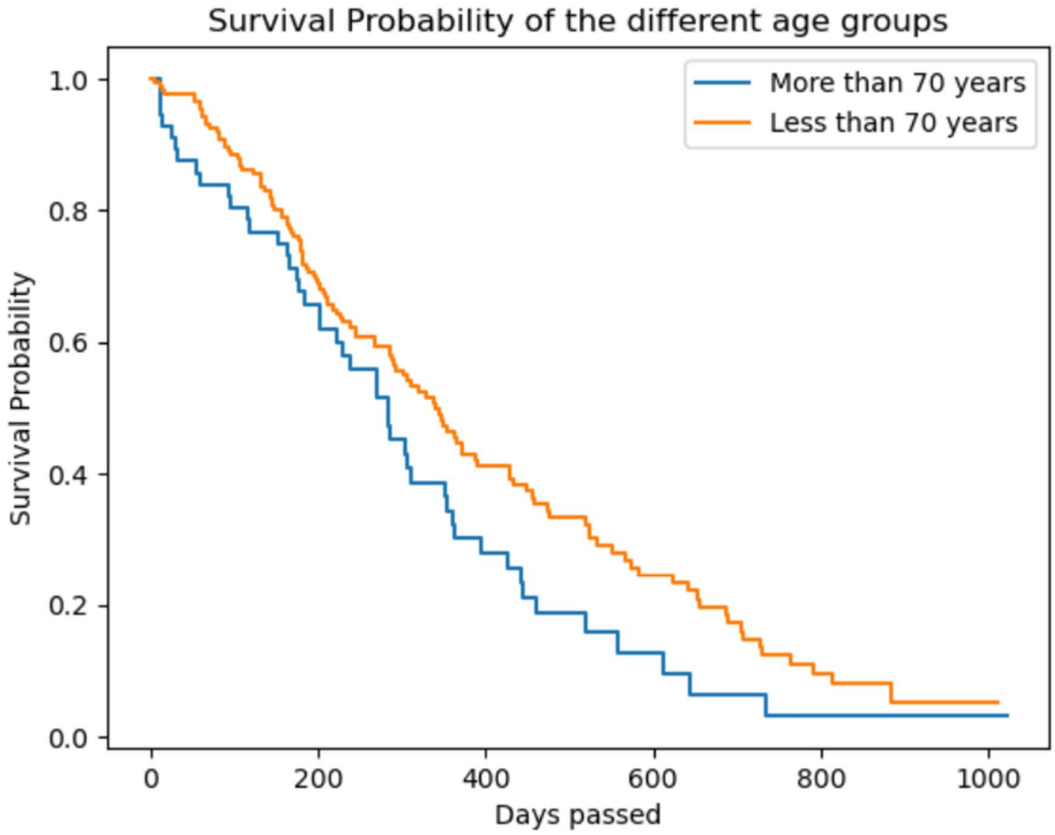


Figure 5: Survival probability of the different age groups

To evaluate the impact of age on survival, divided patients two groups with age ≥70 years (older) and <70 years (younger). Kaplan-Meier analysis showed that the median survival in older patients was significantly shorter than in younger patients. There was an increased hazard of mortality by 28% for older patients (log-rank p=0.0442). Older patients had a steeper decline in survival than younger patients, particularly during the first 200 days. Age ≥70 is a strong negative prognostic factor in lung cancer survival.

3.6.3 Division of two groups by ph.karno scores

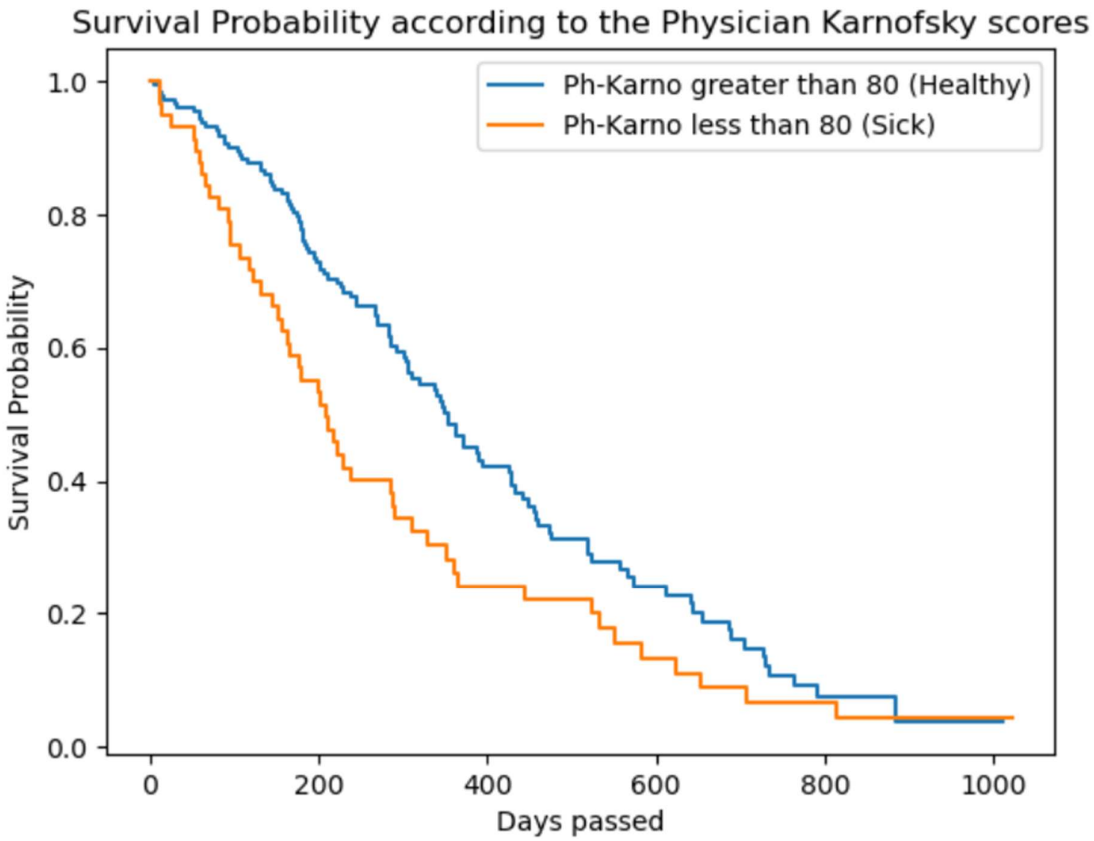


Figure 6: Survival probability according to the physician Karnofsky scores

Stratifying patients by ph.karno score (cut-off 80) showed clear survival differences. Scores >80 were linked to higher survival, with over 70% surviving at 200 days compared to under 50% for scores ≤80. This confirms ph.karno as a strong independent predictor and key covariate in survival models.

3.6.4 Division of two group by ph.ecog scores

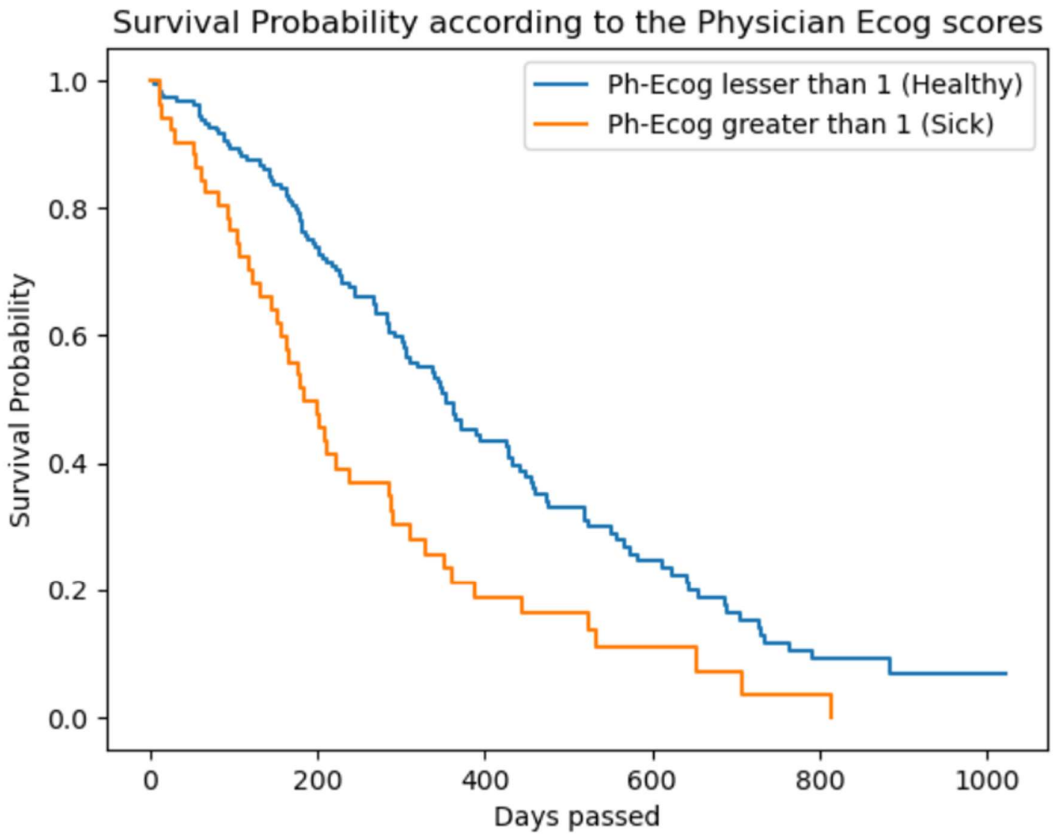


Figure 7: Survival probability according to the Physician Ecog scores

Patients were grouped by ph.ecog scores (≤1 vs. >1). With scores ≤1 group had a greater survival probability that gradually declined with increased risk, while the ph.ecog > 1 had a steep decline in survival probability and higher mortality during the first 300 days of life. The strong separation in survival curves describes the influence ECOG performance status has on survival prediction because it is a good prognostic factor for predicting lung cancer survival.

3.7 Model Preparation

- **Handling Survival Time and Censoring:** We have used prepare_data() function to manipulate and modify a dataset to adhere to base-level survival analysis requirements and specifications. The function will reset the time variable so that the minimum is at least 0.1 (days) to avoid events that happened at zero-time, classify the event status as observed or censored and will drop the ID or other non-predictive columns to avoid overfitting by the model.
- **Strategic Data Partitioning:** Using train_test_split() with a random_state=42, we have split dataset into a training and test subset of 70% and 30%, respectively, and has been controlled to allow for the model types COX-PH, RSF, and AFT to be used for comparison, and the training and testing for prognostic value versus clinical factors to be free of bias.
- **Defining Evaluation Time Horizons:** The _get_evaluation_times() function derives clinically important evaluation times using event time quantiles of the training data

focused on early, mid, and late survival periods. It enables us to cross evaluate survival probabilities in subgroups with time-dependent measurements.

3.8 Model Implementation

3.8.1 The Cox Proportional Hazards survival model

The Cox Proportional Hazards model is a popular choice for survival analysis because it helps predict how different factors affect the risk of death without assuming a specific survival time distribution. We chose it because it's widely used in medical research and can handle censored data, which is common in the lung cancer dataset (63 censored cases). It's also great for testing whether patient self-assessments (karnoPAT) add value compared to physician assessments (karnoPH, ecogPH).

We developed the Cox Proportional Hazards model with the lifelines package on the training data with covariates (age, sex, ph.karno, pat.karno, ph.ecog) to predict estimated survival time and status of death. The key assumptions of the analysis (that hazards are proportional and time-invariant) were well-defined and had violations identified and rectified to prevent bias. The analysis focused on comparing patient-reported (pat.karno) performance scale and physician-reported (ph.karno) performance scale to assess the prognostic information of both scales, including the effect of age (and potentially non-linear effects around 70 years).

3.8.2 Random Survival Forest model

Random Survival Forests (RSF) is a machine learning model that we chose because it can capture complex relationships between features and survival time, unlike simpler models (Ishwaran et al., 2008). It's great for the lung cancer dataset because it handles censoring naturally and doesn't assume linear relationships, which might exist between features like karnoPAT and survival.

The Random Survival Forest (RSF) is based on systematic analytical procedures. Data preparation consists of splitting the datasets into training and testing datasets while maintaining the same distribution of censored data. We trained the RSF model using log-rank splitting and marginalized patient self-reported status (Pat_karno), clinician ratings, and age. The calibration plots at the median survival time indicated good agreement between predicted and observed survival probabilities. Importantly, the analysis directly answered the key research questions, including documenting the significant impact of gender on survival, and also showed that self-assessments by patients (Pat_karno) were the strongest predictors, and improved accuracy of the model compared with just using clinician assessments.

3.8.3 The Weibull Accelerated Failure Time (AFT) model

We applied the Weibull AFT model to estimate, in multiplicative form, the influence of standardized clinical variables (gender, age, ECOG score, patient-rated Karnofsky) on survival time, assuming a Weibull distribution. Fitting the model to 70% of the NCCTG dataset, we used maximum likelihood estimation to estimate time ratios, which made it conceptually easy to interpret the difference in survival time for one unit increase in the covariates. The common statistical significance of each predictor was tested by based Wald tests (α=0.05, =2) with stability confirmed by 95% confidence intervals.

4. Results

4.1 Performance evaluation methods

4.1.1 Concordance Index (C-index)

The concordance index (C-index) is a metric to assess the discriminative ability for survival models in ranking patients by predicted risk. Specifically, it calculates the probability that for two randomly chosen patients, the predicted risk score is higher for the patient who experiences the event (for example, death) first.

The C-index can be calculated as:

$$C - index = \frac{Number\ of\ concordant\ pairs}{Number\ of\ comparable\ pairs}$$

4.1.2 Brier Score

The Brier Score quantifies the calibration accuracy of predicted survival probabilities. It is the mean squared error of observed survival status (1 if alive, 0 dead) and predicted survival probability at a fixed time t. When data are censored, the weights correct for incomplete follow-up:

$$BS(t) = \frac{1}{N}\sum_{i=1}^{N}[S(t|X_i) - \mathrm{I}(T_i > t)]^2 . \omega_i(t)$$

Reason of selection

We chose the Concordance Index (C-index) and Brier score as evaluation metrics because they address censoring in survival data, which is crucial for the NCCTG lung cancer dataset where some patients' survival times exceed follow-up. These metrics give a fuller picture of model performance, unlike accuracy or RMSE, which fail to account for censoring and the time-to-event nature of survival data.

4.2 Results evaluation

4.2.1 Cox Proportional Hazards survival model without pat_karno

	coef	exp (coef)	se (coef)	coef lower 95%	coef up-per 95%	exp(coef) lower 95%	exp(coef) upper 95%	cmp to	z	P	-log2(p)
Age	0.01	1.01	0.01	-0.01	0.03	0.99	1.03	0	1.09	0.27	1.86
Sex	0.61	1.84	0.21	0.2	1.01	1.23	2.76	0	2.94	<0.005	8.25
Ph_ecog	0.54	1.72	0.23	0.09	0.99	1.1	2.69	0	2.37	0.02	5.82
Ph_karno	0.01	1.01	0.01	-0.01	0.03	0.99	1.04	0	0.82	0.41	1.29

Table 5: Cox PH survival model without pat_karno

4.2.2 Cox Proportional Hazards survival model with all features

	coef	exp (coef)	se (coef)	coef lower 95%	coef up-per 95%	exp(coef) lower 95%	exp(coef) upper 95%	cmp to	z	P	-log2(p)
Age	0.01	1.01	0.01	-0.01	0.03	0.99	1.03	0	0.96	0.34	1.56
Sex	0.62	1.86	0.21	0.21	1.03	1.24	2.79	0	2.99	<0.005	8.47
Ph_ecog	0.51	1.66	0.23	0.05	0.96	1.05	2.62	0	2.18	0.03	5.11
Ph_karno	0.01	1.01	0.01	-0.01	0.04	0.99	1.04	0	0.96	0.34	1.57
Pat_karno	-0.01	0.99	0.01	-0.02	0.01	0.98	1.01	0	-0.71	0.48	1.06

Table 6: Cox PH survival model with all features

4.2.3 Weibull AFT model without pat_karno

		coef	exp (coef)	se (coef)	coef lower 95%	coef upper 95%	exp(coef) lower 95%	exp(coef) upper 95%	cmp to	z	P	-log2 (p)
lambda_	Age	-0.01	0.99	0.01	-0.02	0.01	0.98	1.01	0	-1.04	0.3	1.74
	Ph_ecog	-0.36	0.69	0.16	-0.67	-0.06	0.51	0.95	0	-2.31	0.02	5.6
	Ph_karno	-0.01	0.99	0.01	-0.02	0.01	0.98	1.01	0	-0.76	0.45	1.15
	Sex	-0.43	0.65	0.14	-0.71	-0.15	0.49	0.86	0	-2.98	<0.005	8.45
	Intercept	7.61	2027.04	1	5.65	9.58	283.26	14506	0	7.58	<0.005	44.75
rho_	Intercept	0.36	1.44	0.08	0.21	0.51	1.24	1.67	0	4.75	<0.005	18.89

Table 7: Weibull AFT model without pat_karno

4.2.4 Weibull AFT model with all features

		coef	exp (coef)	se (coef)	coef lower 95%	coef upper 95%	exp(coef) lower 95%	exp(coef) upper 95%	cmp to	z	P	-log2(p)
lambda_	Age	-0.01	0.99	0.01	-0.02	0.01	0.98	1.01	0	-0.92	0.36	1.48
	Pat_karno	0	1	0.01	-0.01	0.02	0.99	1.02	0	0.69	0.49	1.02
	Ph_ecog	-0.34	0.71	0.16	-0.65	-0.03	0.52	0.97	0	-2.13	0.03	4.93
	Ph_karno	-0.01	0.99	0.01	-0.03	0.01	0.98	1.01	0	-0.89	0.37	1.42
	Sex	-0.44	0.65	0.14	-0.72	-0.15	0.49	0.86	0	-3.03	<0.005	8.66
	Intercept	7.32	1516.6	1.09	5.19	9.46	179.97	12780.03	0	6.74	<0.005	35.83
rho_	Intercept	0.36	1.44	0.08	0.22	0.51	1.24	1.67	0	4.78	<0.005	19.1

Table 8: Weibull AFT model results with all features

4.2.5 RSF permutation importance results

	age	sex	Ph_karno	Ph_ecog	Pat_karno
Without Pat_karno	0.1417	0.1423	0.0992	0.0817	
With all features	0.1062	0.1214	0.0784	0.0595	0.1056

Table 9: RSF permutation importance results

4.2.6 Model's concordance index comparison

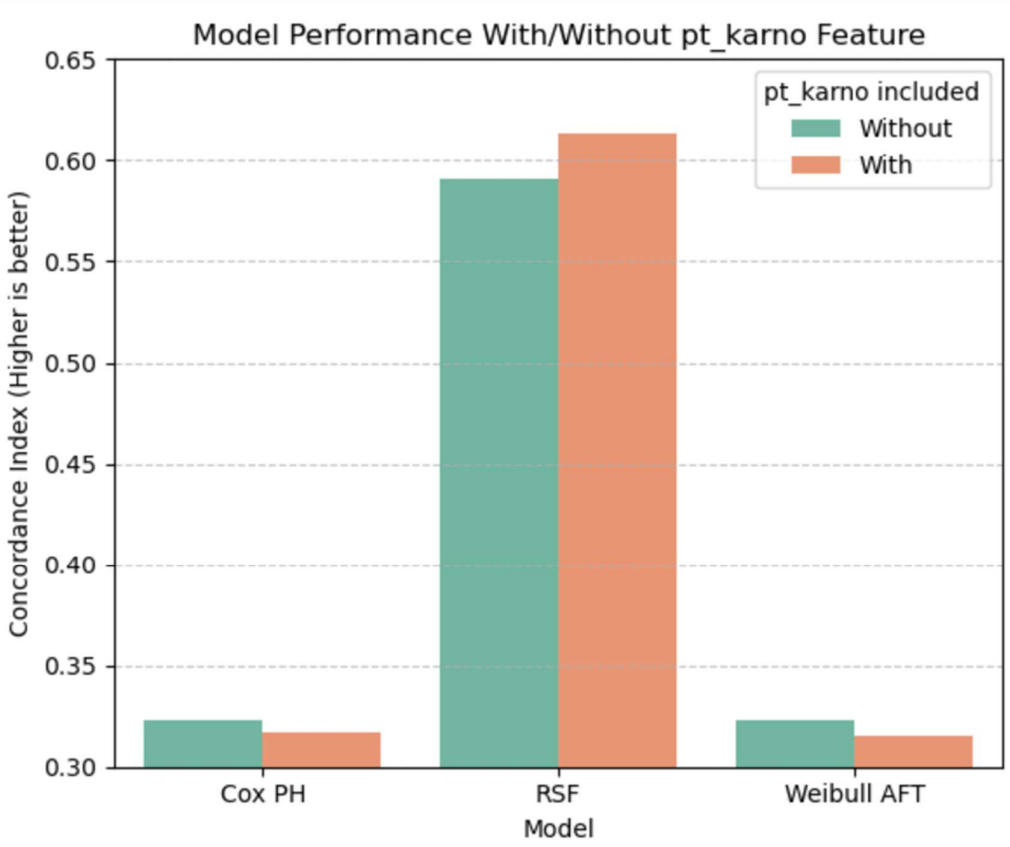


Figure 8: Model's concordance index comparison

4.2.7 Model's overall results

Model	Feature	Evaluation method	Score
Cox Proportional Hazards Model Survival	Without Pat_karno	Concordance Index	0.323
		Brier Score	Time 144.2 days: 0.1484 Time 222.5 days: 0.2040 Time 365.8 days: 0.2065
	With all features	Concordance Index	0.317
		Brier Score	Time 144.2 days: 0.1471 Time 222.5 days: 0.2002 Time 365.8 days: 0.2050
Random Survival Forest Model	Without Pat_karno	Concordance Index	0.591
		Brier Score	Time 144.2 days: 0.1841 Time 222.5 days: 0.2606 Time 365.8 days: 0.2969
	With all features	Concordance Index	0.613
		Brier Score	Time 144.2 days: 0.1697 Time 222.5 days: 0.2370 Time 365.8 days: 0.2586
Weibull Accelerated Failure Model	Without Pat_karno	Concordance Index	0.323
		Brier Score	Time 144.2 days: 0.1467 Time 222.5 days: 0.2036 Time 365.8 days: 0.2068
	With all features	Concordance Index	0.315
		Brier Score	Time 144.2 days: 0.1453 Time 222.5 days: 0.2001 Time 365.8 days: 0.2056

Table 10: Model's overall results

The Cox and Weibull models showed that sex had a consistent and significant effect, with males facing an 86% higher risk of mortality (Cox HR: 1.86, $p<0.005$). ECOG performance status (physician assessed) was also significant, with poorer scores linked to greater risk (HR: 1.72, $p=0.02$). Patient-assessed performance (pat_karno) improved results only in non-linear models: adding it to the Random Survival Forest (RSF) increased the C-index from 0.591 to

0.613 and reduced Brier scores, showing its value in complex models. However, it was not significant in Cox or Weibull (p>0.4), suggesting model-dependent usefulness.

Age and physician-assessed Karnofsky (ph_karno) showed little impact (p>0.05). Among the models, RSF with all features performed best (C-index: 0.613), while Cox (0.317) and Weibull (0.315) showed poor discriminative power, close to chance level. Although Cox and Weibull had better calibration (Brier scores), their weak ranking ability limited clinical value.

These findings confirm the importance of sex and ECOG scores as predictors, while highlighting RSF's advantage in using patient self-assessments to improve predictive accuracy. Nonetheless, the modest RSF C-index indicates some uncertainty, and results should be interpreted with caution.

5. Analysis and Discussion

5.1 Interpretation of Results

The study on the NCCTG dataset provides important perspective on predicting lung cancer survival. Cox PH, RSF, and Weibull AFT models consistently identified sex (female patients demonstrate prolonged survival) and ph_ecog (worse ECOG scores increase the risk of mortality) as important predictors that fit with clinical expectations. However, pat_karno (patient self-appraisal) remained nonsignificant in the parametric models (Cox PH: *p*=0.48; AFT: *p*=0.49) and made limited improvements to RSF. RSF provided the strongest concordance index (0.613) with all features, while parametric models (Cox PH, AFT) therefore provided poor discrimination (C-index for Cox PH and AFT: 0.31–0.32). Brier scores remained lowest in the Cox PH and AFT, respectively, at earlier time points (144.2 days: ~0.15); RSF had a greater score at later periods (365.8 days: 0.2586).

5.2 Best performance model

The Random Survival Forest (RSF) model with all features had better equality than all other models with (C-index: 0.613) because of its approaches to non-linearity, flexibility of our features, and robustness to violations. Moreover, it was able to capture complex interactions, like pat_karno and clinical factors, whereas parametric models (like Cox PH and Weibull AFT) could not capture the complex interactions.

5.3 Comparison to Literature

Performance metrics: Our RSF (C-index: 0.613) underperformed Chen et al. (2022) (C-index: 0.7) and Luo et al. (2025) (C-index: 0.7) potentially due to their sources of richer data (radiomics, SEER database) over our manner of using limited clinical variables. Further Germer et al. (2024) similarly reported RSF performance (C-index: 0.7) indicating our smaller size dataset (n=228) and limited variables collected in NCCTG may have contributed to the lower performance.

Concordance: Features importance of Sex and Ph_ecog aligns with Al Mamlook et al. (2020).
Divergence: Gencer (2025) favored the AFT model but we found RSF as superior than AFT. Germer et al. (2024) reported that the higher RSF performance with (C-Index 0.70), which underscore the NCCTG dataset's limitations.
Innovation: We have uniquely evaluated the pat_karno's feature impact across models, revealing its RSF-specific utility.

5.4 Key evaluations

- The gender's influence is robustly confirmed, with males facing significantly higher mortality risk.
- The Patient self-assessments (Pat_karno) improve predictions in RSF models with validating their complementary role alongside clinical assessments.
- We found features such as sex, ECOG status, and patient Karnofsky as key factors, while age and physician Karnofsky show limited standalone value.
- The RSF model emerges as the superior model, though Cox/Weibull offer better short-term calibration.

5.5 Objectives and Findings

Objective	Results from this study	Comparison with Literature
compare survival probabilities between male and female lung cancer patients using Kaplan-Meier analysis	Males had 86% higher mortality risk (HR = 1.86, $p<0.005$). Kaplan–Meier curves showed significantly shorter survival for males (log-rank $p=0.0013$).	Consistent with May et al. (2023) who reported a persistent survival gap favouring females, and Salmanpour et al. (2025) who found women had 27% reduced mortality risk.
determine whether patients aged ≥70 years have a significantly different survival rate compared to younger patients.	Older patients (≥70) showed 28% higher hazard of mortality (log-rank $p=0.0442$).	Consistent with Corso et al. (2015) who found elderly patients suffer poorer outcomes due to comorbidities and conversely with Sadiq et al. (2025) who found age not statistically significant to predict respondents' outcome.
Evaluate the impact of physician-assessed performance scores (Karnofsky & ECOG) on survival outcomes.	ECOG was a strong predictor (HR = 1.72, $p=0.02$). Physician Karnofsky less significant ($p>0.05$).	Concurs with Al Mamlook et al. (2020) and Sadiq et al. (2025) who also regard ECOG to be more reliable than Karnofsky.
Identify the most influential predictors of survival using Cox Proportional Hazards regression.	Significant predictors: Gender and ECOG. Non-significant: Age, Physician Karnofsky	Supports Sadiq et al. (2025) and Al Mamlook et al. (2020) studies, both found ECOG and gender to be the major predicting factors
Compare Cox PH with RSF and AFT models	RSF outperformed others (C-index = 0.613). Cox PH and Weibull AFT had poor discrimination (C-index ≈ 0.31–0.32).	In contrast, Germer et al. (2024) found Cox PH competitive with RSF (C-index ~0.7). Difference likely due to larger dataset in their study.
Develop survival models such as Cox Proportional Hazards, Random Survival Forests and Accelerated Failure Time .	All three models were implemented successfully. RSF provided a better fit of non-linear effects than Cox/AFT.	Consistent with Chen et al. (2022) who showed RSF performed better than traditional models using radiomics features.
assess the prognostic value of patient self-assessments (Pat_karno) by comparing model performance with and without this feature.	Improved RSF performance (C-index rose from 0.591 → 0.613, reduced Brier score). Not significant in Cox or AFT	Highlights RSF-specific utility. Adds to literature by showing patient self-assessment can complement physician scores.
Evaluate model accuracy using metrics such as the concordance index and Brier score	RSF: best discrimination but higher Brier score at later times. Cox & AFT: better short-term calibration but poor discrimination.	Confirms in Germer et al. (2024) who reported differences in calibration vs discrimination across models

Table 11: Objectives and Findings

Conclusion

This study used the NCCTG lung dataset to develop survival models using time-to-event outcomes. The findings revealed a strong impact of gender on survival, that patient self-assessments improved predictive power of the non-linear models (i.e., discriminative strength of RSF for increased discriminative accuracy and prediction accuracy). ECOG performance status and sex were the most significant clinical predictors for lung cancer patient survival. RSF offered increased overall discrimination compared with parametric approaches but was negatively impacted in short-term prediction accuracy across all models compared with Cox or Weibull model prediction accuracy. These results offer reassurance regarding the feasibility of combining patient-reported outcomes and provide further rationale for using RSF for prognostic purposes specific to lung cancer.

In light of these findings, future work should focus on hyperparameter optimization of RSF and deep survival models in order to improve discriminative power and to include multimodality to overcome dataset limitations.

References

Al Mamlook, R.E., Bzizi, H.F. and Chen, S., 2020, July. Evaluate performance risk score in patients suffering from lung cancer using survival analysis of statistics. In *2020 IEEE International Conference on Electro Information Technology (EIT)* (pp. 145-150). IEEE.

Chaddad, A., Desrosiers, C., Toews, M. and Abdulkarim, B., 2017. Predicting survival time of lung cancer patients using radiomic analysis. *Oncotarget*, *8*(61), p.104393.

Chen, N., Li, R., Jiang, M., Guo, Y., Chen, J., Sun, D., Wang, L. and Yao, X., 2022. Progression-free survival prediction in small cell lung cancer based on radiomics analysis of contrast-enhanced CT. *Frontiers in Medicine*, *9*, p.833283.

Clark, T.G., Bradburn, M.J., Love, S.B. and Altman, D.G., 2003. Survival analysis part I: basic concepts and first analyses. *British journal of cancer*, *89*(2), pp.232-238.

Corso, C.D., Rutter, C.E., Park, H.S., Lester-Coll, N.H., Kim, A.W., Wilson, L.D., Husain, Z.A., Lilenbaum, R.C., Yu, J.B. and Decker, R.H., 2015. Role of chemoradiotherapy in elderly patients with limited-stage small-cell lung cancer. *Journal of Clinical Oncology*, *33*(36), pp.4240-4246.

Extermann, M. and Wedding, U., 2012. Comorbidity and geriatric assessment for older patients with hematologic malignancies: a review of the evidence. *Journal of Geriatric Oncology*, *3*(1), pp.49-57.

Gencer, G., 2025. Lung Cancer Survival Analysis: A Comparative Evaluation of Cox Proportional Hazards and Accelerated Failure Time Models: An Analytical Study. *Türkiye Klinikleri. Tip Bilimleri Dergisi*, *45*(1), pp.8-16.

Germer, S., Rudolph, C., Labohm, L., Katalinic, A., Rath, N., Rausch, K., Holleczek, B., Handels, H. and AI-CARE Working Group, 2024. Survival analysis for lung cancer patients: A comparison of Cox regression and machine learning models. *International Journal of Medical Informatics*, *191*, p.105607.

Gorji, A., Jouzdani, A.F., Sanati, N., Yuan, R., Rahmim, A. and Salmanpour, M.R., 2025. Censor-Aware Semi-Supervised Survival Time Prediction in Lung Cancer Using Clinical and Radiomics Features. *arXiv preprint arXiv:2502.01661*.

He, S., Li, H., Cao, M., Sun, D., Yang, F., Yan, X., Zhang, S., He, Y., Du, L., Sun, X. and Wang, N., 2022. Survival of 7,311 lung cancer patients by pathological stage and histological classification: a multicenter hospital-based study in China. *Translational lung cancer research*, *11*(8), p.1591.

Ishwaran, H., Kogalur, U.B., Blackstone, E.H. and Lauer, M.S., 2008. Random survival forests.

Luo, Q., Zhang, Q., Liu, H., Chen, X., Yang, S. and Xu, Q., 2025. Time-dependent interpretable survival prediction model for second primary NSCLC patients. *International Journal of Medical Informatics*, *195*, p.105771.

Lynch, C.M., Abdollahi, B., Fuqua, J.D., De Carlo, A.R., Bartholomai, J.A., Balgemann, R.N., Van Berkel, V.H. and Frieboes, H.B., 2017. Prediction of lung cancer patient survival via supervised machine learning classification techniques. *International journal of medical informatics*, *108*, pp.1-8.

May, L., Shows, K., Nana-Sinkam, P., Li, H. and Landry, J.W., 2023. Sex differences in lung cancer. *Cancers*, *15*(12), p.3111.

Meng, F.T., Jhuang, J.R., Peng, Y.T., Chiang, C.J., Yang, Y.W., Huang, C.Y., Huang, K.P. and Lee, W.C., 2024. Predicting Lung Cancer Survival to the Future: Population-Based Cancer Survival Modeling Study. *JMIR Public Health and Surveillance*, *10*(1), p.e46737.

Mohanty, S., Devi, Y.S., Sekar, V., Chongthu, J. and Chyrmang, D., 2020. How Does Age Affect Clinicopathology and Survival in Non-Small-Cell Lung Cancer? An Institutional Retrospective Analysis from North-East India. *Journal of the Scientific Society*, *47*(1), pp.17-22.

Mytelka, D.S., Li, L. and Benoit, K., 2018. Post-diagnosis weight loss as a prognostic factor in non-small cell lung cancer. *Journal of cachexia, sarcopenia and muscle*, *9*(1), pp.86-92.

Nakagawa, T., Toyazaki, T., Chiba, N., Ueda, Y. and Gotoh, M., 2016. Prognostic value of body mass index and change in body weight in postoperative outcomes of lung cancer surgery. *Interactive CardioVascular and Thoracic Surgery*, *23*(4), pp.560-566.

Perera, M. and Dwivedi, A.K., 2020. Statistical issues and methods in designing and analyzing survival studies. *Cancer Reports*, *3*(4), p.e1176.

Zhao, Z., Cheng, X., Gao, Y., Tan, F., Xue, Q., Gao, S. and He, J., 2025. Predicting survival in small cell lung cancer patients undergoing various treatments: a machine learning approach. *Translational Lung Cancer Research*, *14*(3), p.736.

Appendix
Import Libraries

```
# Importing required modules
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Survival analysis specific imports
from lifelines import KaplanMeierFitter, CoxPHFitter, WeibullAFTFitter
from lifelines.statistics import logrank_test
from lifelines.utils import concordance_index
from lifelines.calibration import survival_probability_calibration

from pycox.models import CoxPH
import torchtuples as tt

from sksurv.ensemble import RandomSurvivalForest
from sksurv.util import Surv
from sksurv.metrics import concordance_index_censored, brier_score

from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.inspection import permutation_importance
from sklearn.base import BaseEstimator
from sklearn.metrics import brier_score_loss
from sklearn.calibration import calibration_curve
```

Dataset overview

```
df = pd.read_csv('lung_cancer.xls')
df.head()
df.shape
# checking missing values
df.info()
## checking null values
df.isnull().sum()

# Looking at the rows with null values
df[df.isnull().any(axis=1)]

# In here, there are a total of 456 data points. Out of these there 8 attributes which have not any missing values in the dataset which are as
df = df.rename(columns={'Y':'status','TIME':'time','karnoPH': 'ph_karno', 'ecog': 'ph_ecog', 'karnoPAT': 'pat_karno'})
df.head()
# convert cate value into numeric
df['sex'] = df['sex'].map({'M': 1, 'F': 0})
df
```

```
# correlations
corr = df.corr()
sns.heatmap(corr, cmap='coolwarm', annot=True)
plt.title("correlation between features")
plt.show()
```

Conversion of categorical values to numerical

```
df.loc[df['status'] == 0, 'dead'] = 0
df.loc[df['status'] == 1, 'dead'] = 1
df.head()
```

 # finding the number of days a person was alive before they died.

```
durations = df['time']
event_obs = df['dead']

kmf = KaplanMeierFitter()
kmf.fit(durations, event_obs)
```

Creation of new column “dead”

Event Table Overview from KM Estimator
 # looking at the event table

```
event_tab = kmf.event_table

print("Event Table:")

event_tab.head(5)
```

Survival time probability w.r.t. time

```
# survival probability at time t=0:

# theoretical checking of random t values (t = 0, t = 11, t = 13)

# for t = 0 i.e., first row in event table
first_row = event_tab.iloc[0,:]
survival_at_0 = (first_row.at_risk - first_row.observed) / first_row.at_risk
print("Survival at time t = 0 :", survival_at_0)

# for t = 11 i.e., third row in even table
third_row = event_tab.iloc[2,:]
survival_at_11 = (third_row.at_risk - third_row.observed) / third_row.at_risk
print("Survival at time t = 11:", round(survival_at_11, 3))

# for t = 13 i.e., fifth row in even table
fifth_row = event_tab.iloc[4,:]
survival_at_13 = (fifth_row.at_risk - fifth_row.observed) / fifth_row.at_risk
print("Survival at time t = 13:", round(survival_at_13, 3))
```

 # Using kmf's predict function to call out the survival probabilities
kmf.predict(11)
 # looking at all the KM estimates
survival_ls = kmf.survival_function_
survival_ls.head()

Median Survival Period (in days)
# looking at the median number of survival days i.e., the number of days, on an average, 50% of the patients survived.
print("The median survival time:", kmf.median_survival_time_, "days.")

 # Survival Probability with confidence intervals
kmf.confidence_interval_survival_function_

Plotting the graph for survival probability with confidence interval

confidence_surv_func = kmf.confidence_interval_survival_function_

plt.plot(confidence_surv_func["KM_estimate_lower_0.95"], label = "Lower 95%")
plt.plot(confidence_surv_func["KM_estimate_upper_0.95"], label = "Upper 95%")
plt.title("Survival Function with Confidence Interval")
plt.xlabel("Number of days")
plt.ylabel("Survival Probability")
plt.legend();

The median time to an event:

Calculating the conditional median time to the event
kmf.conditional_time_to_event_

Graph for the median time to an event
Conditional median time left for event:

median_time_to_event = kmf.conditional_time_to_event_

plt.plot(median_time_to_event, label = "Median Time Remaining to Event")
plt.title("Median Time Remaining to Event")
plt.xlabel("Total Days")
plt.ylabel("Conditional Median time to Event")
plt.legend();

 # import the Nelson Aalen hazard model
from lifelines import NelsonAalenFitter
# object for NelsonAalenFitter
naf = NelsonAalenFitter()
 # Fitting the data
naf.fit(df['time'], event_observed=df['dead'])

 # finding the cumulative hazard:
naf.cumulative_hazard_

```
# plotting the cumulative hazard graph
naf.plot_cumulative_hazard(ci_show=False)
plt.title("Cumulative Probability  for Event of Interest (Death)")

plt.xlabel("Number of Days")
plt.ylabel("Cumulative Probability of subject's death");
```

Division w.r.t sex

```
df['sex'] = df['sex'].map({'M': 1, 'F': 0})
male = df2[df2['sex'] == 1]
female = df2[df2['sex'] == 0]
results = logrank_test(male['time'], female['time'], male['dead'], female['dead'])
print(f"\nLog-rank test p-value: {results.p_value:.4f}")
if results.p_value < 0.05:
    print("Significant difference in survival between genders (p < 0.05)")
else:
    print("No significant difference in survival between genders")
```

```
# creating 2 objects for the two groups:
kmf_males = KaplanMeierFitter()
kmf_females = KaplanMeierFitter()
# # dividing the data into groups:
# males = df2.query("sex == 1")
# females = df2.query("sex == 0")
males = df2[df2['sex'] == 1]
females = df2[df2['sex'] == 0]

# fitting the data into the models.
kmf_males.fit(durations= males['time'], event_observed= males['dead'], label= "Males")
kmf_females.fit(durations= females['time'], event_observed= females['dead'], label= "Females")

# plotting the graph for the two groups:

kmf_males.plot(ci_show = False)
kmf_females.plot(ci_show = False)

plt.xlabel("Days passed")
plt.ylabel("Survival Probability")
plt.title("Survival Probability of Males and Females");
```

Division w.r.t age

```
# dividing the age into different categories : 1 -> greater than 70 and 0 -> lesser than 70
df_age_cats = df2.copy()

df_age_cats.loc[df_age_cats['age'] >= 70, 'age_cat'] = 1
df_age_cats.loc[df_age_cats['age'] < 70, 'age_cat'] = 0
df_age_cats.head(2)
```

```
# creating kmf objects for the two groups and fitting the categorized data
old   = df2[df2['age'] >= 70].copy()
young = df2[df2['age'] <  70].copy()

old   = old.dropna(subset=['time','dead'])
young = young.dropna(subset=['time','dead'])
kmf_old = KaplanMeierFitter()
kmf_young = KaplanMeierFitter()

kmf_old.fit(durations= old['time'], event_observed= old['dead'], label= "More than 70 years")
kmf_young.fit(durations= young['time'], event_observed= young['dead'], label= "Less than 70
years")
# Log-rank test
results = logrank_test(old['time'], young['time'], old['dead'], young['dead'])
print(f"\nLog-rank test p-value: {results.p_value:.4f}")
if results.p_value < 0.05:
   print("Conclusion: Significant difference in survival between age groups (p < 0.05)")
else:
   print("Conclusion: No significant difference in survival between age groups")

# ECOG score groups
df_clean = df2.copy()
df_clean['ph_ecog_cat'] = df_clean['ph_ecog'].apply(lambda x: '0-1' if x in [0, 1] else '2+')

plt.figure(figsize=(10,6))
kmf_ecog_low = KaplanMeierFitter()
kmf_ecog_high = KaplanMeierFitter()

ecog_low = df_clean[df_clean['ph_ecog_cat'] == '0-1']
ecog_high = df_clean[df_clean['ph_ecog_cat'] == '2+']

kmf_ecog_low.fit(ecog_low['time'], ecog_low['dead'], label='ECOG 0-1 (Good)')
kmf_ecog_high.fit(ecog_high['time'], ecog_high['dead'], label='ECOG 2+ (Poor)')

ax = kmf_ecog_low.plot(ci_show=False)
kmf_ecog_high.plot(ax=ax, ci_show=False)
plt.title('Survival by ECOG Performance Score', fontsize=14)
plt.xlabel('Days Since Diagnosis', fontsize=12)
plt.ylabel('Survival Probability', fontsize=12)
plt.grid(True, alpha=0.3)
plt.tight_layout()
plt.show()

# plotting the graph for the two groups:
kmf_old.plot(ci_show = False)
kmf_young.plot(ci_show = False)

plt.xlabel("Days passed")
plt.ylabel("Survival Probability")
plt.title("Survival Probability of the different age groups")
```

Division w.r.t Ph_karno score

```
df2['ph_karno'].hist();
plt.title("Distribution of Physician Karnofsky Scores");

# dividing the age into 2 categories :
# 1 -> people who have a ph.karno >= 80 -- healthy
# 0 -> people who have a ph.karno < 80 -- sick
df_pat_karno = df2.copy()

df_pat_karno.loc[df_pat_karno['pat_karno'] >= 80, 'pat_karno_cat'] = 1
df_pat_karno.loc[df_pat_karno['pat_karno'] < 80, 'pat_karno_cat'] = 0
# dividing the data into groups:

pat_karno_healthy = df_pat_karno.query("pat_karno_cat == 1")
pat_karno_sick = df_pat_karno.query("pat_karno_cat == 0")
# creating kmf objects for the two groups and fitting the categorized data

kmf_pat_karno_healthy = KaplanMeierFitter()
kmf_pat_karno_sick = KaplanMeierFitter()


kmf_pat_karno_healthy.fit(durations = pat_karno_healthy['time'], event_observed = pat_karno_healthy['dead'], label= "Pat-Karno greater than 80 (Healthy)")
kmf_pat_karno_sick.fit(durations = pat_karno_sick['time'], event_observed = pat_karno_sick['dead'], label= "Pat-Karno less than 80 (Sick)")
# plotting the graph for the two groups:

kmf_pat_karno_healthy.plot(ci_show = False)
kmf_pat_karno_sick.plot(ci_show = False)

plt.xlabel("Days passed")
plt.ylabel("Survival Probability")
plt.title("Survival Probability creating kmf objects for the two groups and fitting the categorized data")

kmf_pat_karno_healthy = KaplanMeierFitter()
kmf_pat_karno_sick = KaplanMeierFitter()
kmf_pat_karno_healthy.fit(durations = pat_karno_healthy['time'], event_observed = pat_karno_healthy['dead'], label= "Pat-Karno greater than 80 (Healthy)")
kmf_pat_karno_sick.fit(durations = pat_karno_sick['time'], event_observed = pat_karno_sick['dead'], label= "Pat-Karno less than 80 (Sick)")
```

Distribution of Physician Karnofsky score

```
df2['ph_karno'].hist();
plt.title("Distribution of Physician Karnofsky Scores");
# dividing the age into 2 categories :
# 1 -> people who have a ph.karno >= 80 -- healthy
# 0 -> people who have a ph.karno < 80 -- sick
df_ph_karno = df2.copy()
```

```
df_ph_karno.loc[df_ph_karno['ph_karno'] >= 80, 'ph_karno_cat'] = 1
df_ph_karno.loc[df_ph_karno['ph_karno'] < 80, 'ph_karno_cat'] = 0
# dividing the data into groups:

ph_karno_healthy = df_ph_karno.query("ph_karno_cat == 1")
ph_karno_sick = df_ph_karno.query("ph_karno_cat == 0")

# creating kmf objects for the two groups and fitting the categorized data

kmf_ph_karno_healthy = KaplanMeierFitter()
kmf_ph_karno_sick = KaplanMeierFitter()

kmf_ph_karno_healthy.fit(durations = ph_karno_healthy['time'], event_observed = ph_karno_healthy['dead'], label= "Ph-Karno greater than 80 (Healthy)")
kmf_ph_karno_sick.fit(durations = ph_karno_sick['time'], event_observed = ph_karno_sick['dead'], label= "Ph-Karno less than 80 (Sick)")

# plotting the graph for the two groups:
kmf_ph_karno_healthy.plot(ci_show = False)
kmf_ph_karno_sick.plot(ci_show = False)

plt.xlabel("Days passed")
plt.ylabel("Survival Probability")
plt.title("Survival Probability according to the Physician Karnofsky scores");
```

Division w.r.t Ph_Ecog score

```
df2['ph_ecog'].hist()
plt.title("Distribution of Physician Ecog Scores");
# dividing the age into 2 categories :
# 1 -> people who have a ph.ecog <= 1 -- healthy
# 0 -> people who have a ph.karno > 1 -- sick
df_ph_ecog = df2.copy()

df_ph_ecog.loc[df_ph_ecog['ph_ecog'] <= 1, 'ph_ecog_cat'] = 1
df_ph_ecog.loc[df_ph_ecog['ph_ecog'] > 1, 'ph_ecog_cat'] = 0

# dividing the data into groups:

ph_ecog_health = df_ph_ecog.query("ph_ecog_cat == 1")
ph_ecog_sick = df_ph_ecog.query("ph_ecog_cat == 0")

# creating kmf objects for the two groups and fitting the categorized data

kmf_ph_ecog_healthy = KaplanMeierFitter()
kmf_ph_ecog_sick = KaplanMeierFitter()

kmf_ph_ecog_healthy.fit(durations = ph_ecog_health['time'], event_observed = ph_ecog_health['dead'], label= "Ph-Ecog lesser than 1 (Healthy)")
```

```
kmf_ph_ecog_sick.fit(durations = ph_ecog_sick['time'], event_observed = ph_ecog_sick['dead'], label= "Ph-Ecog greater than 1 (Sick)")
```

 # plotting the graph for the two groups:

```
kmf_ph_ecog_healthy.plot(ci_show = False)
kmf_ph_ecog_sick.plot(ci_show = False)

plt.xlabel("Days passed")
plt.ylabel("Survival Probability")
plt.title("Survival Probability according to the Physician Ecog scores");
```

Data Preparation

```
# looking at the null values:
df2.isnull().sum()

df2.info()
# running a base analysis first by removing all the null values without any imputation

# temporary copy of the original dataframe
df_cph = df2.copy()

dropper_subset = list(df_cph.columns)
df_cph.dropna(subset=dropper_subset, inplace=True)
df_cph.drop(['status'], axis=1, inplace=True) # dropping the status column also as the dead column is sufficient
df2.head()
df_clean = df2.copy()

def _prepare_data(df):
    """Common data preparation steps for all models."""
    df = df.copy()
    df['time'] = df['time'].clip(lower=0.1)
    if 'ID' in df.columns:
        df = df.drop(columns=['ID'])
    return train_test_split(df, test_size=0.3, random_state=42)

def _get_evaluation_times(train_df):
    """Get evaluation times based on event times."""
    event_times = train_df.loc[train_df['dead'] == 1, 'time']
    return np.quantile(event_times[event_times > 0], [0.25, 0.5, 0.75])


def _calculate_brier_scores(test_df, surv_prob, times, model_name=""):
    """Calculate and print Brier scores at evaluation times."""
    print("\nBrier Scores:")
    brier_scores = []

    for i, t in enumerate(times):
```

```
        preds = 1 - surv_prob.iloc[i].values if hasattr(surv_prob, 'iloc') else 1 - surv_prob[:, i]

        event_occurred = test_df['dead'].astype(bool)
        time_le_t = (test_df['time'] <= t).values
        y_true = (event_occurred & time_le_t).astype(float)

        censored = ~test_df['dead'].astype(bool)
        censored_before_t = censored & (test_df['time'] < t).values
        mask = ~censored_before_t

        if mask.sum() == 0:
            print(f"  Time {t:.1f} days - skipping (no events)")
            continue

        score = brier_score_loss(y_true[mask], preds[mask])
        brier_scores.append(score)
        print(f"  Time {t:.1f} days: {score:.4f}")

    return brier_scores
def _plot_calibration(test_df, pred_probs, time_point, model_name=""):
    """Plot calibration curve at a specific time point."""
    fig, ax = plt.subplots(figsize=(8, 5))
    y_true = (test_df['dead'].astype(bool) & (test_df['time'] <= time_point)).astype(float)

    fraction_of_positives, mean_predicted_value = calibration_curve(
        y_true, pred_probs, n_bins=10, strategy='quantile'
    )

    ax.plot(mean_predicted_value, fraction_of_positives, "s-", label="Calibration Curve")
    ax.plot([0, 1], [0, 1], "k--", label="Perfect Calibration")
    ax.set_title(f"{model_name} Calibration at t ≤ {time_point:.0f} days")
    ax.set_xlabel("Predicted Probability of Mortality")
    ax.set_ylabel("Observed Probability of Mortality")
    ax.legend()
    plt.tight_layout()
    plt.show()
```

Implementation of Cox PH model

```
def run_cox_analysis(df):
    """Run Cox PH analysis with assumption checking, C-index, and Brier score."""
    print("\n" + "="*50)
    print("Cox Proportional Hazards Model")
    print("="*50)

    train_df, test_df = _prepare_data(df)

    # Fit model
    cph = CoxPHFitter()
    cph.fit(train_df, duration_col='time', event_col='dead')
    cph.print_summary()
```

```
    # Check assumptions
    print("\nChecking Proportional Hazards Assumptions:")
    cph.check_assumptions(train_df, p_value_threshold=0.05, show_plots=True)

    # Concordance Index
    risk_scores = cph.predict_partial_hazard(test_df)
    y_test_surv = Surv.from_arrays(event=test_df['dead'].astype(bool), time=test_df['time'])
    c_index = concordance_index_censored(y_test_surv['event'], y_test_surv['time'], -
risk_scores.values)[0]
    print(f"\nCox PH Concordance Index: {c_index:.3f}")

    # Brier Score
    times = _get_evaluation_times(train_df)
    surv_prob = cph.predict_survival_function(test_df, times=times)
    brier_scores = _calculate_brier_scores(test_df, surv_prob, times)

    # Calibration plot
    median_time = np.median(train_df.loc[train_df['dead'] == 1, 'time'])
    if not np.isnan(median_time):
        pred_probs = 1 - cph.predict_survival_function(test_df,
times=[median_time]).iloc[0].values
        _plot_calibration(test_df, pred_probs, median_time, "Cox PH")

    return cph, c_index, brier_scores
```

Implementation of Random Survival Forest Model

```
def run_rsf_analysis(df):
    """Run Random Survival Forest analysis with C-index and Brier score."""
    print("\n" + "="*50)
    print("Random Survival Forest Model")
    print("="*50)

    train_df, test_df = _prepare_data(df)

    # Prepare data
    X_train = train_df.drop(columns=['time', 'dead'])
    y_train = Surv.from_arrays(event=train_df['dead'].astype(bool), time=train_df['time'])
    X_test = test_df.drop(columns=['time', 'dead'])
    y_test = Surv.from_arrays(event=test_df['dead'].astype(bool), time=test_df['time'])

    # Fit model
    rsf = RandomSurvivalForest(n_estimators=100, random_state=42)
    rsf.fit(X_train, y_train)
        # ========== NEW: Statistical Tests ==========
    print("\nStatistical Significance Testing (Univariate Log-Rank):")
    for feature in X_train.columns:
        if len(np.unique(X_train[feature])) > 1:  # Only test if feature has variation
            # Create binary groups for categorical features
```

```
        if len(np.unique(X_train[feature])) <= 5:
            median_val = np.median(X_train[feature])
            group1 = (X_train[feature] > median_val).values
            group2 = (X_train[feature] <= median_val).values

            # Log-rank test
            results = logrank_test(y_train['time'][group1],
                                   y_train['time'][group2],
                                   y_train['event'][group1],
                                   y_train['event'][group2])

            print(f"{feature}: p-value = {results.p_value:.4f} {'*' if results.p_value < 0.05 else
"}")


    # Feature Importance
    try:
        result = permutation_importance(rsf, X_train, y_train, n_repeats=10, random_state=42)
        print("Permutation Importance:")
        for i in result.importances_mean.argsort()[::-1]:
            print(f"  {X_train.columns[i]}: {result.importances_mean[i]:.4f}")
    except Exception as e:
        print(f"Could not calculate feature importance: {str(e)}")

    # Concordance Index
    risk_scores = rsf.predict(X_test)
    c_index = concordance_index_censored(y_test['event'], y_test['time'], risk_scores)[0]
    print(f"\nRSF Concordance Index: {c_index:.3f}")

    # Brier Score
    times = _get_evaluation_times(train_df)
    unique_times = np.unique(y_train["time"][y_train["event"]])
    surv_prob = rsf.predict_survival_function(X_test, return_array=True)

    # Calculate Brier scores for each time point
    brier_scores = []
    print("\nBrier Scores:")
    for t in times:
        time_idx = np.searchsorted(unique_times, t)
        if time_idx >= len(unique_times):
            print(f"  Time {t:.1f} days - skipping (beyond last event)")
            continue

        preds = 1 - surv_prob[:, time_idx]
        event_occurred = test_df['dead'].astype(bool)
        time_le_t = (test_df['time'] <= t).values
        y_true = (event_occurred & time_le_t).astype(float)
        censored = ~test_df['dead'].astype(bool)
        censored_before_t = censored & (test_df['time'] < t).values
        mask = ~censored_before_t
```

```
        if mask.sum() == 0:
            print(f"  Time {t:.1f} days - skipping (no events)")
            continue

        score = brier_score_loss(y_true[mask], preds[mask])
        brier_scores.append(score)
        print(f"  Time {t:.1f} days: {score:.4f}")

    # Calibration plot
    median_time = np.median(train_df.loc[train_df['dead'] == 1, 'time'])
    if not np.isnan(median_time):
        time_idx = np.searchsorted(unique_times, median_time)
        if time_idx < len(unique_times):
            pred_probs = 1 - surv_prob[:, time_idx]
            _plot_calibration(test_df, pred_probs, median_time, "RSF")

    return rsf, c_index, brier_scores
```

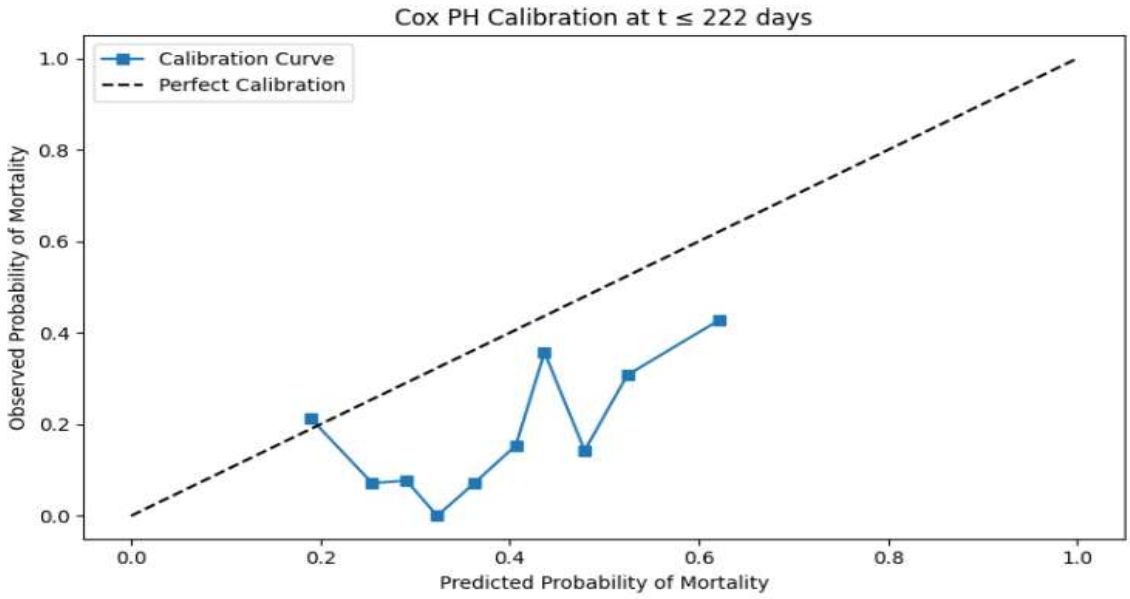
Implementation of Weibull AFT model

```
def run_weibull_aft_analysis(df):
    """Run Weibull AFT model with evaluation metrics and statistical tests."""
    print("\n" + "="*50)
    print("Weibull Accelerated Failure Time Model")
    print("="*50)

    train_df, test_df = _prepare_data(df)

    # Fit model
    aft = WeibullAFTFitter()
    aft.fit(train_df, duration_col='time', event_col='dead')

    # ========== Enhanced Model Summary ==========
    print("\nModel Summary with Statistical Tests:")
    print(aft.print_summary())

    # ========== NEW: Detailed Statistical Output ==========
    print("\nDetailed Statistical Significance:")
    print("---------------------------------")
    print("Null hypothesis: coefficient = 0 (no effect)")
    print("Alternative hypothesis: coefficient ≠ 0\n")

    # Get the full summary dataframe
    summary_df = aft.summary

    # Print each coefficient with p-value and confidence intervals
    for idx, row in summary_df.iterrows():
        if idx != '_intercept':  # Skip intercept
            print(f"Variable: {idx}")
```

```
        print(f"  Coefficient: {row['coef']:.4f}")
        print(f"  exp(coef): {np.exp(row['coef']):.4f} (time ratio)")
        print(f"  p-value: {row['p']:.4f} {'*' if row['p'] < 0.05 else ''}")
        print(f"  95% CI: [{row['coef lower 95%']:.4f}, {row['coef upper 95%']:.4f}]")
        print(f"    Interpretation: A unit increase in {idx} changes survival time by {100*(np.exp(row['coef'])-1):.1f}%")
        print("---------------------------------")

    # Rest of the original function...
    # Concordance Index
    risk_scores = aft.predict_expectation(test_df)
    y_test_surv = Surv.from_arrays(event=test_df['dead'].astype(bool), time=test_df['time'])
    c_index = concordance_index_censored(y_test_surv['event'], y_test_surv['time'], risk_scores.values)[0]
    print(f"\nWeibull AFT Concordance Index: {c_index:.3f}")

    # Brier Score
    times = _get_evaluation_times(train_df)
    surv_prob = aft.predict_survival_function(test_df, times=times)
    brier_scores = _calculate_brier_scores(test_df, surv_prob, times)

    # Calibration plot
    median_time = np.median(train_df.loc[train_df['dead'] == 1, 'time'])
    if not np.isnan(median_time):
        pred_probs = 1 - aft.predict_survival_function(test_df, times=[median_time]).iloc[0].values
        _plot_calibration(test_df, pred_probs, median_time, "Weibull AFT")

    return aft, c_index, brier_scores
```

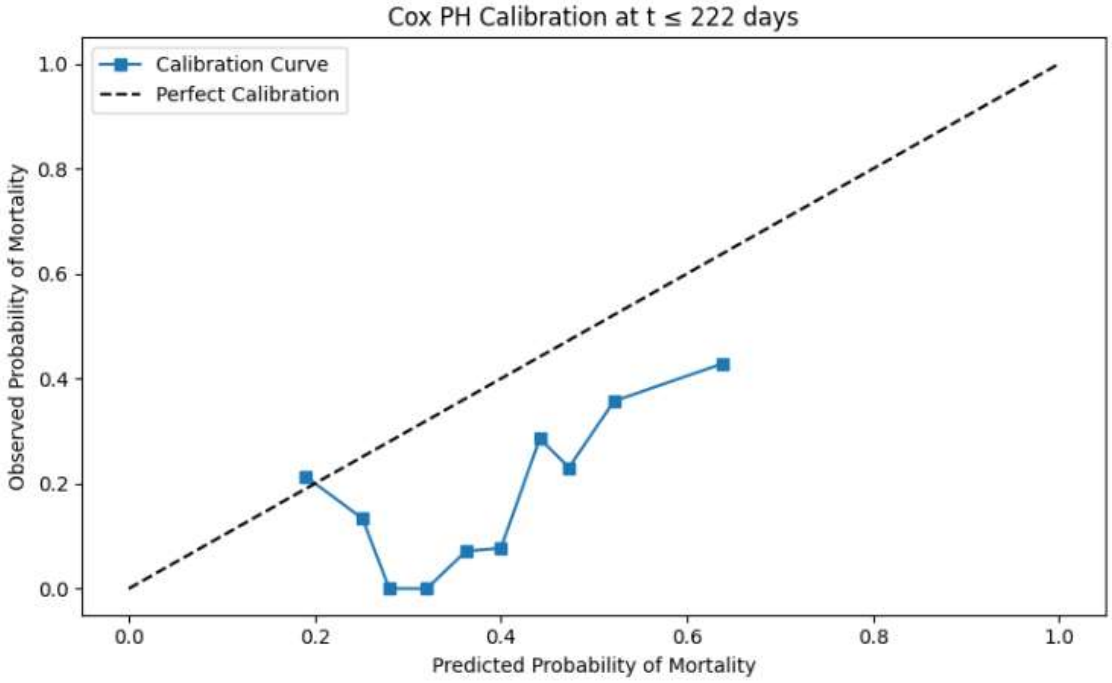
Cox PH Calibration at t < 222 days without Pat_Karno

```
df_no_patkarno  = df_cph.copy().drop(columns=['pat_karno'])
# 1. Cox PH Analysis
cph_model = run_cox_analysis(df_no_patkarno)
```

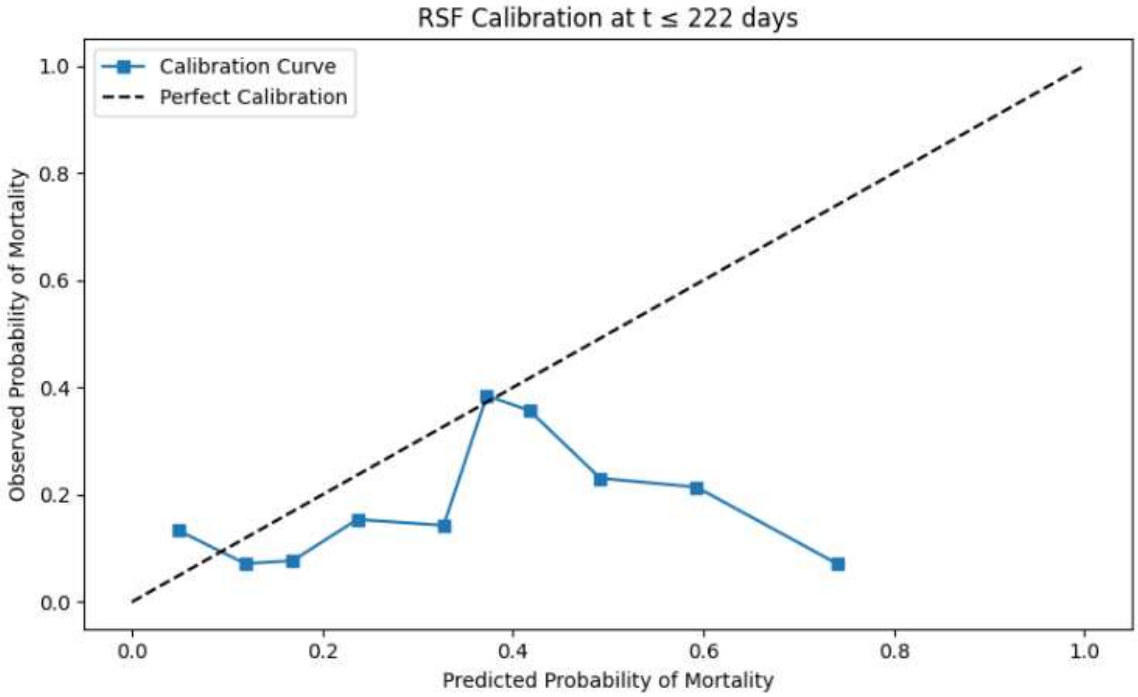


Cox PH Calibration at t < 222 days with all features

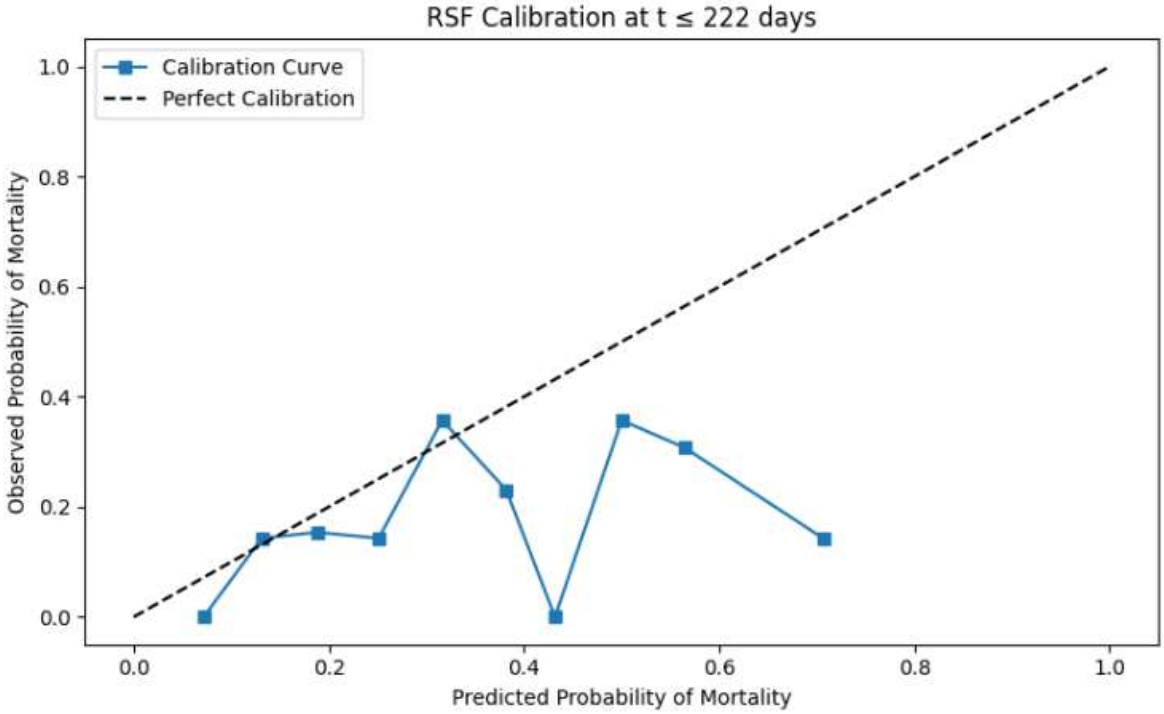
```
df = df_cph.copy()
   # 1. Cox PH Analysis
cph_model = run_cox_analysis(df)
```



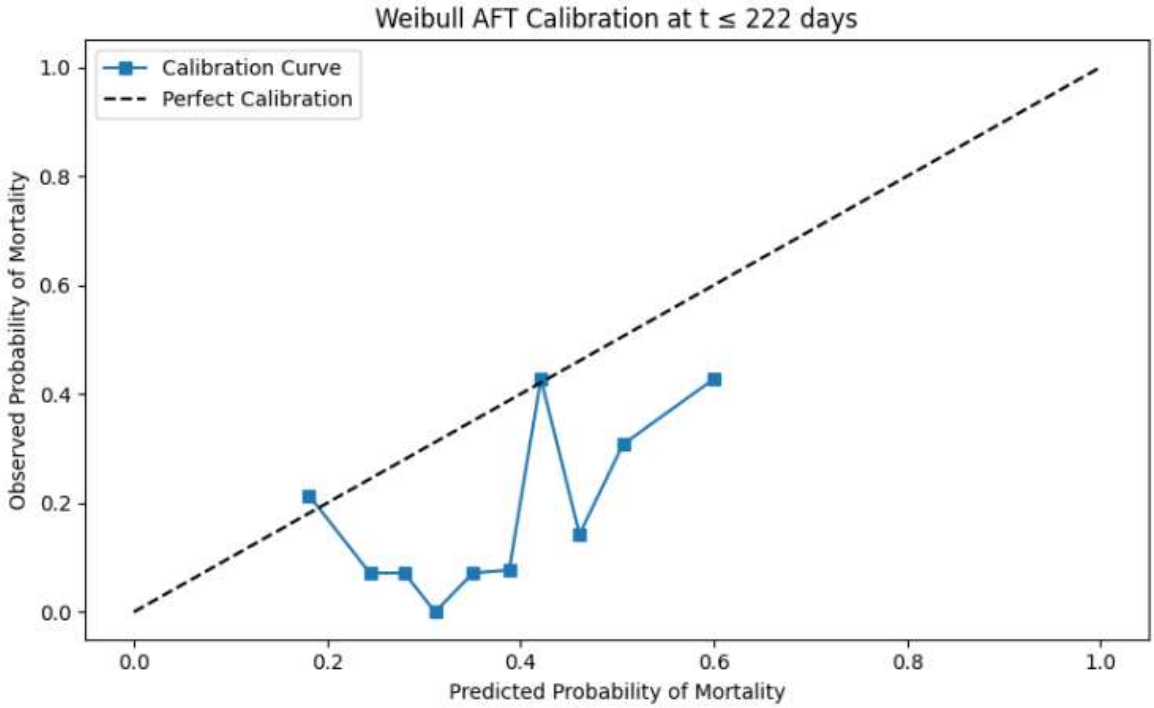
RSF Calibration at t < 222 days without Pat_Karno
rsf_model = run_rsf_analysis(df_no_patkarno)



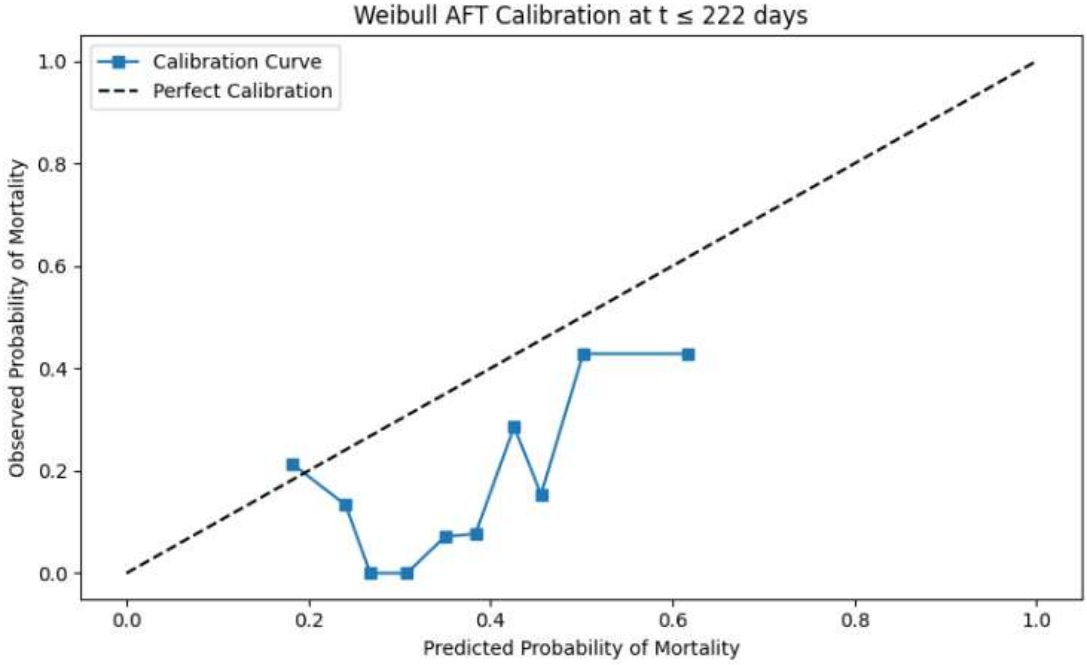
RSF Calibration at t < 222 days with all features
rsf_model = run_rsf_analysis(df)



Weibull AFT Calibration at t < 222 days without Pat_Karno
aft_model_no_patkarno = run_weibull_aft_analysis(df_no_patkarno)



Weibull AFT Calibration at t < 222 days with all features
aft_model = run_weibull_aft_analysis(df)



```
# comparison DataFrame
results = pd.DataFrame({
    'Model': ['Cox PH', 'Cox PH', 'RSF', 'RSF', 'Weibull AFT', 'Weibull AFT'],
    'pt_karno': ['Without', 'With', 'Without', 'With', 'Without', 'With'],
    'Concordance_Index': [0.323,0.317,0.591,0.613,0.323,0.315]
})

# Plot comparison
plt.figure(figsize=(6, 5))
sns.barplot(data=results, x='Model', y='Concordance_Index', hue='pt_karno', palette='Set2')
plt.title('Model Performance With/Without pt_karno Feature')
plt.ylabel('Concordance Index (Higher is better)')
plt.ylim(0.3, 0.65)
plt.legend(title='pt_karno included')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()

# Display the results table
print("\nPerformance Comparison:")
display(results)
```