Kaunas University of Technology

Faculty of Mathematics and Natural Sciences

Mortality Rate Estimation Machine Learning Models of Patients with Prostate Cancer Diagnosis

Master's Final Degree Project

Vytautas Kraujalis

Project author

Doc. Dr. Tomas Ruzgas

Supervisor

Kaunas, 2023

Kaunas University of Technology

Faculty of Mathematics and Natural Sciences

Mortality Rate Estimation Machine Learning Models of Patients with Prostate Cancer Diagnosis

Master's Final Degree Project

Applied Mathematics (6211AX006)

Vytautas Kraujalis

Project author

Doc. Dr. Tomas Ruzgas

Supervisor

Prof. Dr. Robertas Alzbutas

Reviewer

Kaunas, 2023

Kaunas University of Technology

Faculty of Mathematics and Natural Sciences

Vytautas Kraujalis

Mortality Rate Estimation Machine Learning Models of Patients with Prostate Cancer Diagnosis

Declaration of Academic Integrity

I confirm the following:

1. I have prepared the final degree project independently and honestly without any violations of the copyrights or other rights of others, following the provisions of the Law on Copyrights and Related Rights of the Republic of Lithuania, the Regulations on the Management and Transfer of Intellectual Property of Kaunas University of Technology (hereinafter – University) and the ethical requirements stipulated by the Code of Academic Ethics of the University;

2. All the data and research results provided in the final degree project are correct and obtained legally; none of the parts of this project are plagiarised from any printed or electronic sources; all the quotations and references provided in the text of the final degree project are indicated in the list of references;

3. I have not paid anyone any monetary funds for the final degree project or the parts thereof unless required by the law;

4. I understand that in the case of any discovery of the fact of dishonesty or violation of any rights of others, the academic penalties will be imposed on me under the procedure applied at the University; I will be expelled from the University and my final degree project can be submitted to the Office of the Ombudsperson for Academic Ethics and Procedures in the examination of a possible violation of academic ethics.

Vytautas Kraujalis

Confirmed electronically

Kraujalis Vytautas. Mortality rate estimation machine learning models of patients with prostate cancer diagnosis. Master's Final Degree Project supervisor doc. Dr. Tomas Ruzgas; Faculty of Mathematics and Natural Sciences, Kaunas University of Technology.

Study field and area (study field group): Mathematics, Applied mathematics.

Keywords: prostate cancer, men’s mortality risk, survival analysis, discrete-time modelling, logistic regression, random forest, XGBoost, neural network.

Kaunas, 2023. 72.

Summary

Prostate cancer is one of the most common type of men cancer all around the world, including Lithuania. According to World Health Organization 2020 report, prostate cancer was the 4th most common type of cancer and the 2nd between men while by the mortality cases it was the 8th and 5th between men. In Lithuania, the prostate cancer is the most common type of cancer and is the 4th by the mortality incidences and the 2nd between men. This is why it is important for a healthcare professional to distinguish fatal and non-fatal patient’s prostate cancer, this can be done with the help of a machine learning model, which we will try to implement in this work using the data collected in Kaunas Clinics. For mortality risk estimation 4 machine learning models have been created: logistic regression, random forest, XGBoost and neural network. Models were trained for 4 different response variables: cancer specific mortality, death from other causes, biochemical recurrence and metastases. These models have been trained on randomly sampled training set consisting of 1251 observations, models were evaluated on testing set consisting of 313 patients. Dataset have been transformed from continuous time to discrete time data. The hyperparameters of models were found with the use of 5-fold cross validation within training set applying Bayesian optimization method. Optimal models were selected for each response variable. We have obtained that random forest model showed the best AUC value on testing set comparing to other 3 methods on 3 targets. In case of cancer specific mortality, training and testing set average AUC values are respectively 0.951 (SD = 0.037) and 0.928 (SD = 0.045), death from other causes respectively 0.663 (SD = 0.049) and 0.689 (SD = 0.046), biochemical recurrence respectively 0.865 (SD = 0.030) and 0.855 (SD = 0.034). In case of metastases, optimal model was found to be XGBoost, training and testing set average AUC values respectively 0.997 (SD = 0.005) and 0.927 (SD = 0.035).

Kraujalis, Vytautas. Pacientų, kuriems diagnozuotas priešinės liaukos vėžys, mirtingumo rizikos vertinimo mašininio mokymosi modeliai. Magistro baigiamasis projektas / vadovas doc. dr. Tomas Ruzgas; Kauno technologijos universitetas, Matematikos ir gamtos mokslų fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Matematikos mokslai, taikomoji matematika.

Reikšminiai žodžiai: prostatos vėžys, vyrų mirtingumo rizika, išgyvenamumo analizė, diskretinio laiko modeliavimas, logistinė regresija, atsitiktiniai miškai, “XGBoost”, neuroninis tinklas.

Kaunas, 2023. 72 p.

Santrauka

Prostatos vėžys yra viena dažniausių vyrų vėžio formų pasaulyje, įskaitant ir Lietuvą. Pagal Pasaulio Sveikatos Organizacijos 2020 duomenis, prostatos vėžys buvo 4 dažniausią vėžio forma pasaulyje ir 2 tarp vyrų, o pagal mirtingumo skaičius – 8 pasaulyje ir 5 tarp vyrų. Lietuvoje prostatos vėžys buvo dažniausia vėžio forma, 4 pagal bendrus mirtingumo rodiklius ir 2 tarp vyrų. Todėl yra svarbu gydytojui laiku atskirti mirtiną ir nemirtiną paciento prostatos vėžį, tą padaryti galima pasitelkus mašininio mokymosi modeliu, kuriuos šiame darbe ir bandysime realizuoti panaudodami Kauno Klinikose sukauptus duomenis. Mirtingumo rizikos vertinimui sudarėme 4 mašininio mokymosi modelius: logistinės regresijos, atsitiktinių miškų, “XGBoost” ir neuroninių tinkle. Modeliai buvo sudaryti naudojant 4 skirtingus tikslo kintamuosius: mirtingumui nuo prostatos vėžio, mirtingumui nuo kitų priežasčių, biocheminiam pasikartojimui ir metastazei. Šie modeliai buvo apmokyti naudojant atsitiktinai atrinktą apmokymo imtį, sudarytos iš 1251 stebinių, o modeliai įvertinti naudojant validavimo imtį, sudarytą iš 313 stebinių. Duomenys buvo transformuoti iš nuolatinio laiko į diskretinio laiko duomenis. Modelių hyperparametrai rasti naudojant 5 lygių kryžminę validaciją su apmokymo duomenimis taikant Bajeso optimizavimo metodą. Atrinkome optimalius modelius kiekvieno tikslo kintamojo atveju. Gavome, jog atsitiktinių miškų modelis parodė geriausius rezultatus pagal validavimo imties vidutines AUC reikšmes lyginant su kitais 3 metodais trijų tikslo kintamųjų atveju. Mirtingumo nuo prostatos vėžio atveju, apmokymo ir validavimo imties vidutinė AUC atitinkamai 0.951 (SD = 0.037) ir 0.928 (SD = 0.045), mirtingumo nuo kitų priežasčių atitinkamai 0.663 (SD = 0.049) ir 0.689 (SD = 0.046), biocheminio pasikartojimo atitinkamai 0.865 (SD = 0.030) ir 0.855 (SD = 0.034). Metastazės atveju, optimalus modelis buvo „XGBoost“ metodas, apmokymo ir validavimo imties vidutinės AUC atitinkamai 0.997 (SD = 0.005) ir 0.927 (SD = 0.035).

Table of contents

List of figures

List of tables

List of abbreviations and terms

Abbreviations:

ADA – adaptive boosting

AI – artificial intelligence

ANN – artificial neural network

AUC – Area under the ROC curve

BCR – biochemical recurrence

BMI – body mass index

CI – confidence interval

CSM – cancer specific mortality

Cox – semi-parametric Cox proportional hazard regression model

DL – deep learning

DNN – deep neural network

DT – decision tree

Fine-Gray – Fine-Gray competing risk model

HRPC – high-risk prostate cancer

IQR – interquartile range

KN – K-neighbours

LDA – linear discriminant analysis

LF – linear classification

ML – machine learning

MLP – multi-layer perceptron

MLPC – multi-layer perceptron classifier

MTS – metastases

NB – Naïve Bayes

NPV – negative predictive value

OM – overall mortality

PPV – positive predictive value

PSA – prostate specific antigen

QD – quadratic discriminant analysis

RF – random forest

ROC – receiver operating characteristic

RP – radical prostatectomy

SD – standard deviation

SVM – support vector machine

XGBoost – extreme gradient boosting

Introduction

The research problem and the relevance of the work. According to World Health Organization 2020 report, prostate cancer was the 4th most common type of cancer and the 2nd between men while by the mortality cases it was the 8th and 5th between men [1]. In Lithuania, the problem is even worse, as there the prostate cancer is the most common type of cancer and is the 4th by the mortality incidences and the 2nd between men. Most commonly, prostate cancer develops slowly, but there can be cases when an aggressive type of cancer develops and heavily disturbs men's health [3]. In the Western world, a man has 40% lifetime risk of getting diagnosed with prostate cancer, yet there is only 10% risk of becoming symptomatic [4]. Establishing a treatment plan for a patient in time is difficult as it is unknown where the diagnosed cancer will develop to a fatal stage or to an indolent one that does not have a fatal risk.

Work objective: To develop mortality risk estimation machine learning models for patients diagnosed with prostate cancer.
Work tasks:

- Get acquainted with already performed research in the field of prostate cancer.
- Perform descriptive analysis of the data sample and split the data to train/test datasets.
- Transform data from continuous time to discrete time data.
- Find optimal hyperparameters for each machine learning model.
- Create linear regression, random forest, XGBoost and neural network mortality risk estimation machine learning models.
- Compare the created models.

Research methods: Scientific literature analysis, data analysis and visualisation using *python* programming language, Bayesian hyperparameter optimization using *hyperopt* module, linear regression, random forest and neural network models using *sklearn* module, XGBoost model using *xgboost* module.

Research novelty: Usually, when examining scientific publications, classical method such as either Cox or Fine-Gray model are found being used but there is no comparison with other machine learning models. Also, the results are presented for one issue – cancer specific mortality, biochemical recurrence, etc., but are not compared with each other.

Work structure: The project consists of three main sections: literature review, research methods and research results. The first chapter presents a review of the literature in scientific publications about prostate cancer, mortality from it and applied research methods. The second chapter describes theoretical information about survival analysis, discrete time modelling, some information about used machine learning models, evaluation methods as well as the structure of the research. Finally, the third chapter provides information about the research results, examples and interpretations of the results, and the comparison of the models.

Approval of results: The work was presented at the student scientific conference "New Generation of Scientists" organized by the Lithuanian Science Council on 2023-04-18 as well as at DAMSS 2021: 12th conference on data analysis methods for software systems on 2021-12-02. The work was also published in the Baltic journal of modern computing by the University of Latvia on 2022-06-16.

1. Literature review

1.1. Prostate cancer

According to World Health Organization 2020 report, prostate cancer was the 4th most common type of cancer and the 2nd between men while by the mortality cases it was the 8th and the 5th between men [1]. In Lithuania, prostate cancer is the most common type of cancer by the number of cases, the 4th by the mortality incidences and the 2nd among men. Research and analysis has been done on the risk of prostate cancer [3], and results show that while most commonly the prostate cancer develops slowly, there can be cases when an aggressive cancer develops and heavily disturbs men's health. In the Western world, a man has 40% lifetime risk of being diagnosed with prostate cancer, yet there is only a 10% risk of becoming symptomatic [4].

There are many factors which contributes to prostate cancer risk (genetic, environmental, stochastic effects) but they are also largely unknown [5]. Some risk factors [6]:

- **Family history**: associated significantly with prostate cancer risk. But could be influenced by the detection bias.
- **Hormones:** elevated concentrations of testosterone, metabolite, dihydrotestosterone may increase the risk over a long period time. But the results are inconsistent.
- **Race:** the highest incidence rates for prostate cancer are among African-American men. But this could be caused by differences in access to care, decision-making of whether to seek medical attention, differences in dietary, genetic differences and etc.
- **Aging and oxidative stress:** risk could be increased due to increase in oxidative stress. Supportive evidence on this is also limited.
- **Diet:** fat consumption (polyunsaturated fat) shows a strong and positive correlation with increased risk, while vitamin D deficiency increases risk. Intake of vitamin E decreases the risk.

Several studies also discuss about the importance or significance of stratifying patients into risk groups [7] but there are and can be several techniques in stratification logic [8].

1.2. Predictive models

Having patient's data, in most cases the predictive models give us the probability of an event occurring withing a specific time. Sometimes this is not a primary focus of a study and instead, the prediction of whether the event will occur or no is more important. An example can also be thought of in urologist workday: a decision must be made to either operate a patient with clinically localized prostate cancer or no, for such decision, the probability of cancer recurrence is a very important decision factor [9].

1.2.1. Classical statistical methods

To research and determine the impact of prostate cancer, researchers usually perform what is called *survival analysis* [3]. To perform such analysis, researchers take advantage of the parametric methods which helped to understand better about the patients and how prostate cancer affects men. Using such parametric analysis, a study even debunked the idea that the risk of developing prostate cancer was higher in African American men than the other races [3]. The cumulative incidence function, multivariate logistic regression model can be used to measure the biochemical recurrence, metastasis,

cancer specific mortality or overall mortality [7]. Moreover, to predict various outcomes related to prostate cancer, nomograms, look-up tables, classification and regression-tree analysis, propensity scores or risk-group stratification models are also used in practice [10]. Semi-parametric Cox proportional hazards regression is a frequent method in survival analysis between researchers [11-19] as well as Fine-Gray competing risk regression [12; 14; 20].

1.2.2. Artificial intelligence methods

Modern life has been already changed by the machine learning (ML) and there is no doubt that artificial intelligence (AI) is revolutionizing healthcare where together with ML methodologies, complex insights can be gathered about the patients [21-23].

A machine learning approach differs from traditional approaches by a key distinction that a ML model learns from real examples whereas tradition approach is based on set of predefined rules [24]. Previously we have mentioned classical statistical methods, it is worth mentioning that there is no simple distinction between a machine learning method and a classical statistical method. While we cannot simply distinguish those 2 mentioned approaches, there is a more sophisticated version of machine learning methodology – deep learning (DL), which leverages the use of artificial neural networks that are able to capture extremely complex relationships and provide accurate predictions [21; 24]. Both ML and DL are able to provide useful insights to a researcher and on some occasions – outperform classical statistical methods or human physicians but as a result, there is a limitation on the usage of such methodologies, ML and especially DL can require an enormous amount of data to be trained on in order to give accurate and reliable predictions (**Fig. 1**) [21; 24]. Having a complex model which can find various relationships in data to predict accurately is undeniably important but on the flipside, those models (especially DL models) are often uninterpretable and are so called "black boxes" (**Fig. 1**) [21].

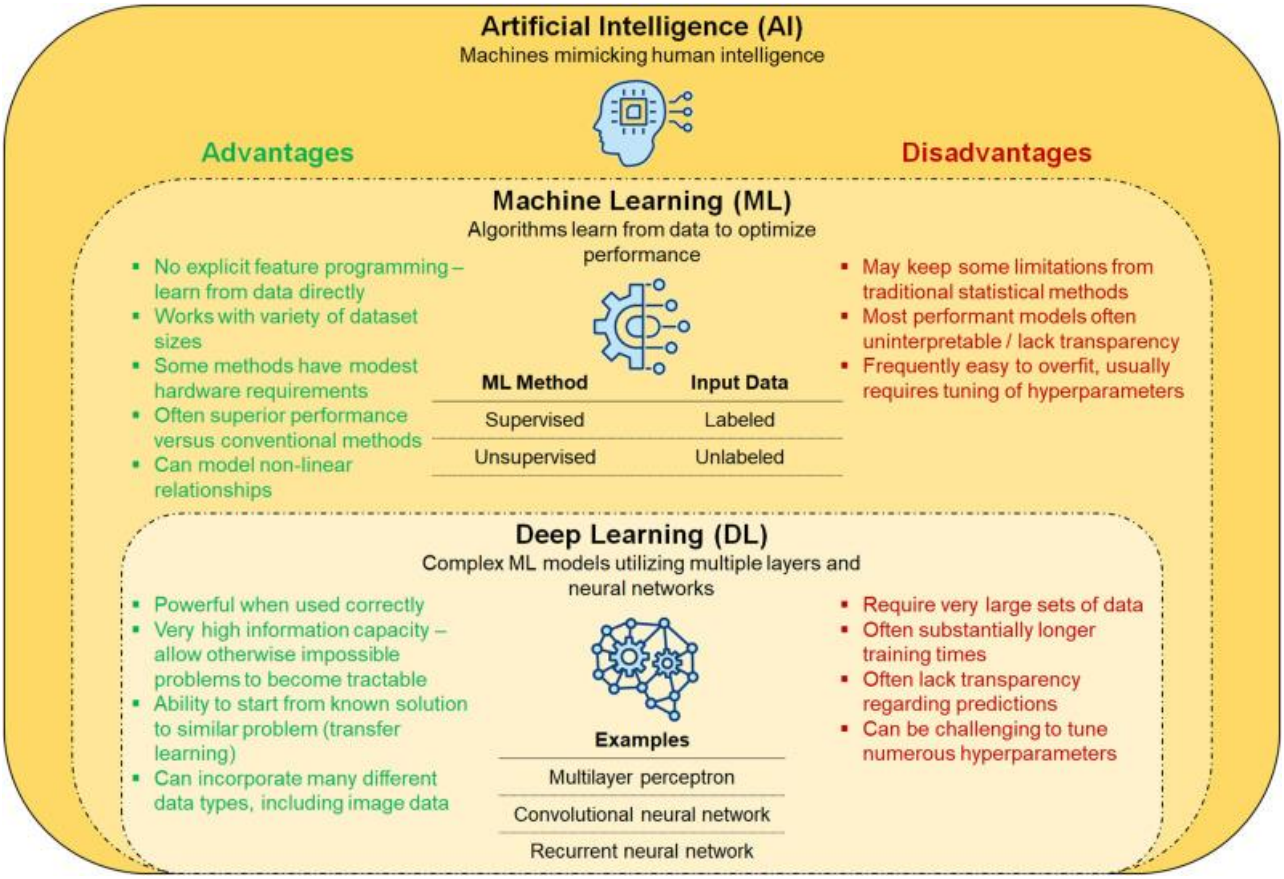

Fig. 1. Schematic diagram describing advantages and disadvantages of ML methodologies [23]

We have learned previously that prostate cancer is not such an easy topic and there are many unknown factors leading to it. Traditional approaches to identify prostate cancer risk can be limited as they will not be able to capture complex relationships between various factors. As a result, AI methods are used in prostate cancer outcome prediction [10; 22]. Such prognostic algorithms developed using AI methods can help urologists in diagnosis of an aggressive prostate cancer quicker and without fewer unnecessary biopsies [21].

1.3. The significance of PSA persistence in prostate cancer risk groups

A study on similar data as ours was done by Milonas D., Venclovas Z., Sasnauskas G. and Ruzgas T. on assessing the significance of prostate specific antigen persistence in prostate cancer risk groups on long-term oncological outcomes [7]. Authors defined persistent prostate specific antigen (PSA) as ≥0.1 ng/mL at 4-8 weeks after radical prostatectomy (RP). The patients were also stratified into 3 risk groups: low, intermediate and high which was made using preoperative PSA, pathological stage, grade group and lymph node status.

The study reported 10-year cumulative incidences of biochemical recurrence (BCR), metastases (MTS), cancer specific mortality (CSM) and overall mortality (OM) using the cumulative incidence function with persistent vs. undetectable PSA in different groups (**Fig. 2**). This chi-square and Mann-Whitney U test was used to assess the difference in groups. The Kaplan-Meier survival curves were used to estimate survival. Multivariate logistic regression model was used to measure the relationship between covariates and the incidences by providing hazard ratios. A total of 1225 patients were

analysed with median follow-up 103 months and 246 (20.1%) patients had persistent PSA. 226 (18.4%) overall deaths were recorded where 45 (3.8%) of were cancer related deaths. 383 (31.3%) experienced biochemical recurrence and 87 (7.1%) – metastasis. The research reports 10-year cumulative incidence of BCR, MTS, CSM and OM – 39.61% (95% CI: 35.95-43.64), 9.70% (95% CI: 7.67-12.27), 4.81% (95% CI: 3.49-6.61) and 18.15% (95% CI: 16.04-20.53). Incidence was significantly higher with PSA persistence for BCR, MTS, CSM and OM: 27.21% (95% CI: 23.68-31.28) undetectable PSA vs 88.46% (95% CI: 83.48-93.74) persistent PSA, 4.40% (95% CI: 3.05-6.35) vs 29.57% (95% CI: 22.41-39.04), 1.92% (95% CI: 1.92-3.05) vs 17.61% (95% CI: 11.70-26.49) and 15.21% (95% CI: 13.16-17.59) vs 30.92% (95% CI: 25.23-37.89). The PSA persistence was detected as significant predictor in multivariate regression analysis for each BCR, MTS, CSM and OM with hazard ratios: 4.2 (95% CI: 3.06-5.76, p<0.001), 2.7 (95% CI: 1.44-5.09, p=0.002), 5.5 (95% CI: 2.08-14.49, p=0.006) and 1.8 (95% CI: 1.13-2.76, p=0.01).

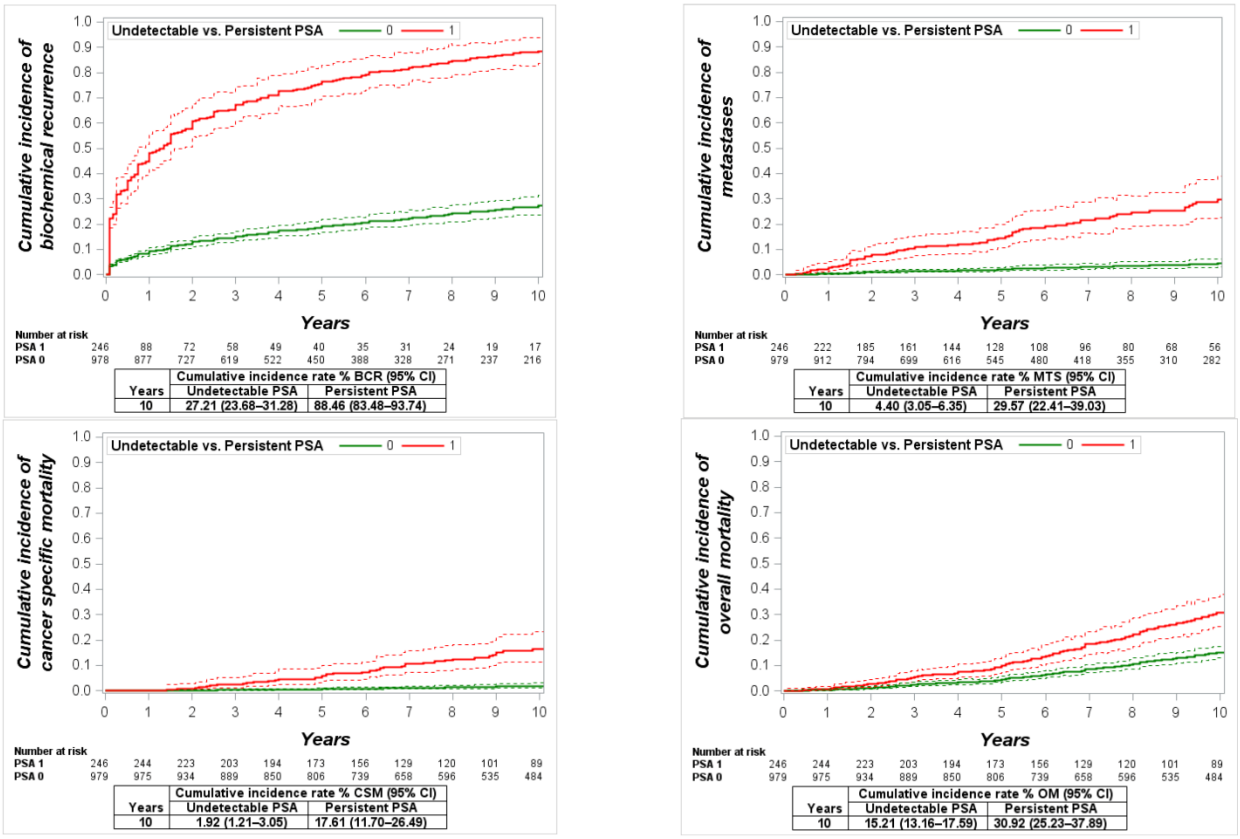

Fig. 2. 10-year cumulative incidences of BCR, MTS, CSM and OM [7]

Authors conclude that PSA persistence is a strong predictor of BCR, MTS, CSM and OM; this significance is also observed in different risk groups. But the significance is marginal in low-risk group, while it has the biggest impact in the high-risk group.

1.4. Artificial neural network for prostate cancer risk prediction

David A. Roffman et al. developed and validated a multi-parameterized artificial neural network (ANN) on the basis of personal health information for prostate cancer risk prediction and stratification [5].

The authors did a 70/30% split for training and validation where 1171 patients with prostate cancer and 70023 without cancer were selected for training; 501 patients with prostate cancer and 30010 without – for validation, the split was done randomly. NHIS data set was used to train ANN; features included: age, body mass index (BMI), smoking status, emphysema, asthma, diabetes status, history of stroke, hypertension, heart disease, race, ethnicity, vigorous exercise habits. Model structure consisted of two hidden layers with 12 neurons in each layer (**Fig. 3**). The inputs were normalized to the 0-1 range. The response of such model was individual's risk for receiving a diagnosis of prostate cancer.

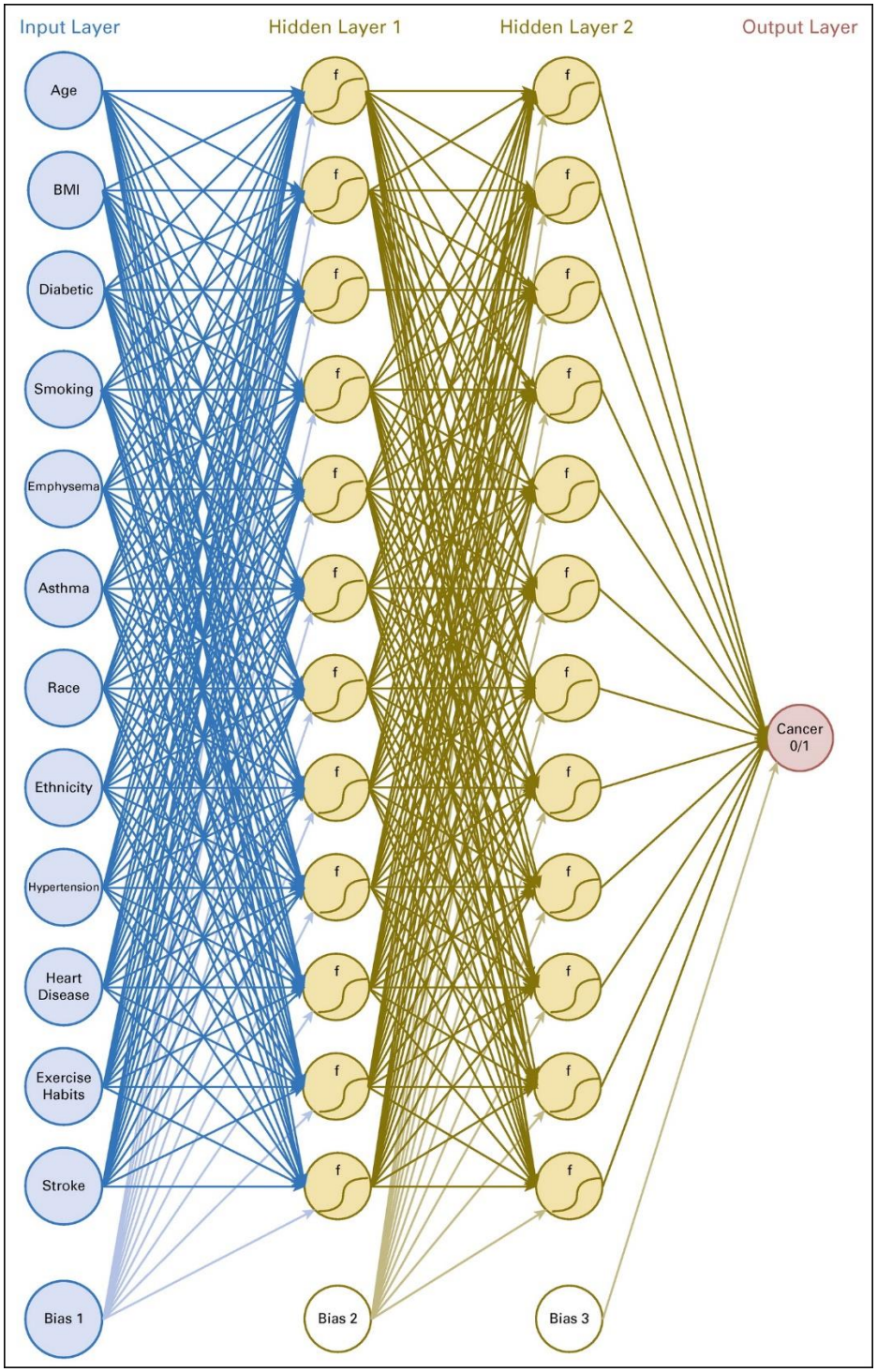

Fig. 3. A schematic diagram of ANN [5]

A priori risk stratification scenarios were also created for high, medium and low risk (**Fig. 4**). There, blue shading shows high-risk category, yellow – medium-risk and red – high-risk. In such risk stratification, high-risk patients could be recommended to undergo screening for prostate cancer, whereas medium-risk patients might be recommended based on their personal preference and low-risk patients would be encouraged not to be screened. The authors selected the thresholds for the groups so that only 1% of the noncancer population would be classified as high risk and only 1% of the cancer population would be classified as low risk.

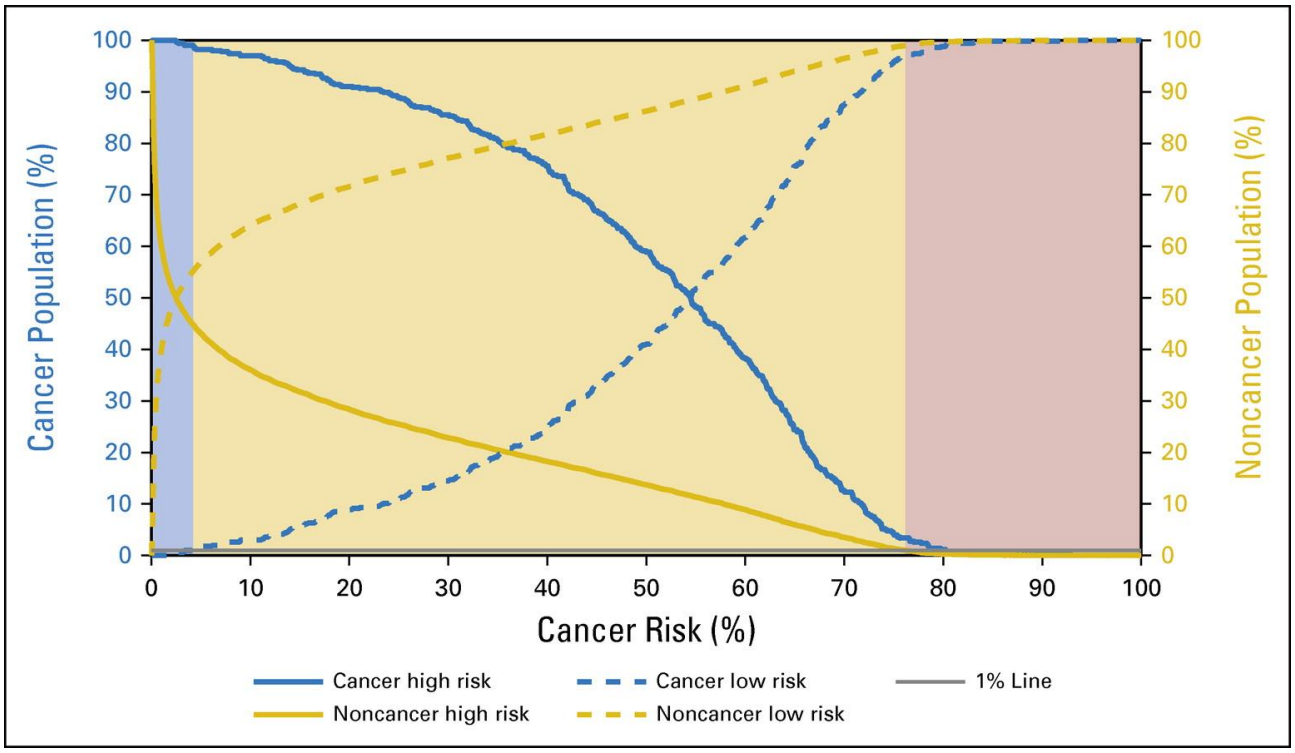

Fig. 4. Cumulative distribution function for the cancer and noncancer population in the validation set [5]

Sensitivity for prostate cancer prediction was 21.5% (95% CI: 19.2 – 23.9), specificity 91% (95% CI: 90.8 – 91.2) on training set and for validation set: sensitivity 23.2% (95% CI: 19.5 – 26.9) and specificity 89.4% (95% CI: 89 – 89.7). Area under the ROC curve (AUC) values were also reported for both training and validation set, respectively: 0.73 (95% CI: 0.71 – 0.75) and 0.72 (95% CI: 0.70 – 0.75).

1.5. Development and validation of an interpretable artificial intelligence model to predict 10-year prostate cancer mortality

Jean-Emmanuel Bibault et al. presented a gradient-boosted model which predicts 10-year prostate cancer mortality with high accuracy [25].

Article uses data from the Prostate, Lung, Colorectal, and Ovarian (PLCO) Cancer Screening Trial with selected prostate cancer diagnosed patients. A dataset consists of 8776 patients and it was randomly split (80/20 ratio) into a training set of 7021 men and a testing set of 1755. In total, 685 (6.2%) patients have died from prostate cancer during follow-up. These features were used in the analysis:

- Prostate cancer: PSA, T, N, M stage, Gleason score and initial treatment (if performed)

- Medical history: Age, height, weight, current smoking status, smoking pack-years, daily alcohol consumption, history of prostatitis, nocturia, arthritis, bronchitis, diabetes, emphysema, heart attack, hypertension, liver disease, osteoporosis, stroke, elevated cholesterol.
- Physical activity: Activity at least once a month during the last year, physical activity at work
- Socio-economic status: Family income, education
- Hormonal status: Hair pattern at age 45, weight gain pattern

An XGBoost was used to predict prostate cancer 10-year mortality; hyperparameters were selected using the training dataset, nested cross-validation and Bayesian optimization technique. Inside XGBoost, the class imbalance was compensated with scale_pos_weight hyperparameter (controls the balance of positive and negative weights [31]).

Authors achieved an excellent model accuracy of 0.98 (±0.01) on the test dataset:

Table 1. Performances of the survival model evaluated with the bootstrap method on the test dataset.

Metric	Result
Accuracy	0.98 (±0.01)
Precision	0.80 (±0.1)
Recall	0.60 (±0.08)
F1-score	0.66 (±0.07)
auROC	0.80 (±0.04)
prAUC	0.54 (±0.07)

In this paper, Shapley values have been used to interpret the predictions, where a high Shapley value would indicate a greater risk of prostate cancer mortality and vice versa for lower values – decreased risk of dying. Feature importance was also analysed, the article reports the five features that contributed most to model performance: Gleason score, PSA at diagnosis, age, type of initial treatment and T stage. Medical features such as alcohol consumption, hormonal status and physical activity also found to be significant in making predictions (**Fig. 5**). In this graph, a colour represents the feature's value, red – high feature value (e.g. age - older), blue – low feature value (e.g. age - younger). Higher SHAP value (x-axis) indicates a greater risk of prostate cancer mortality.

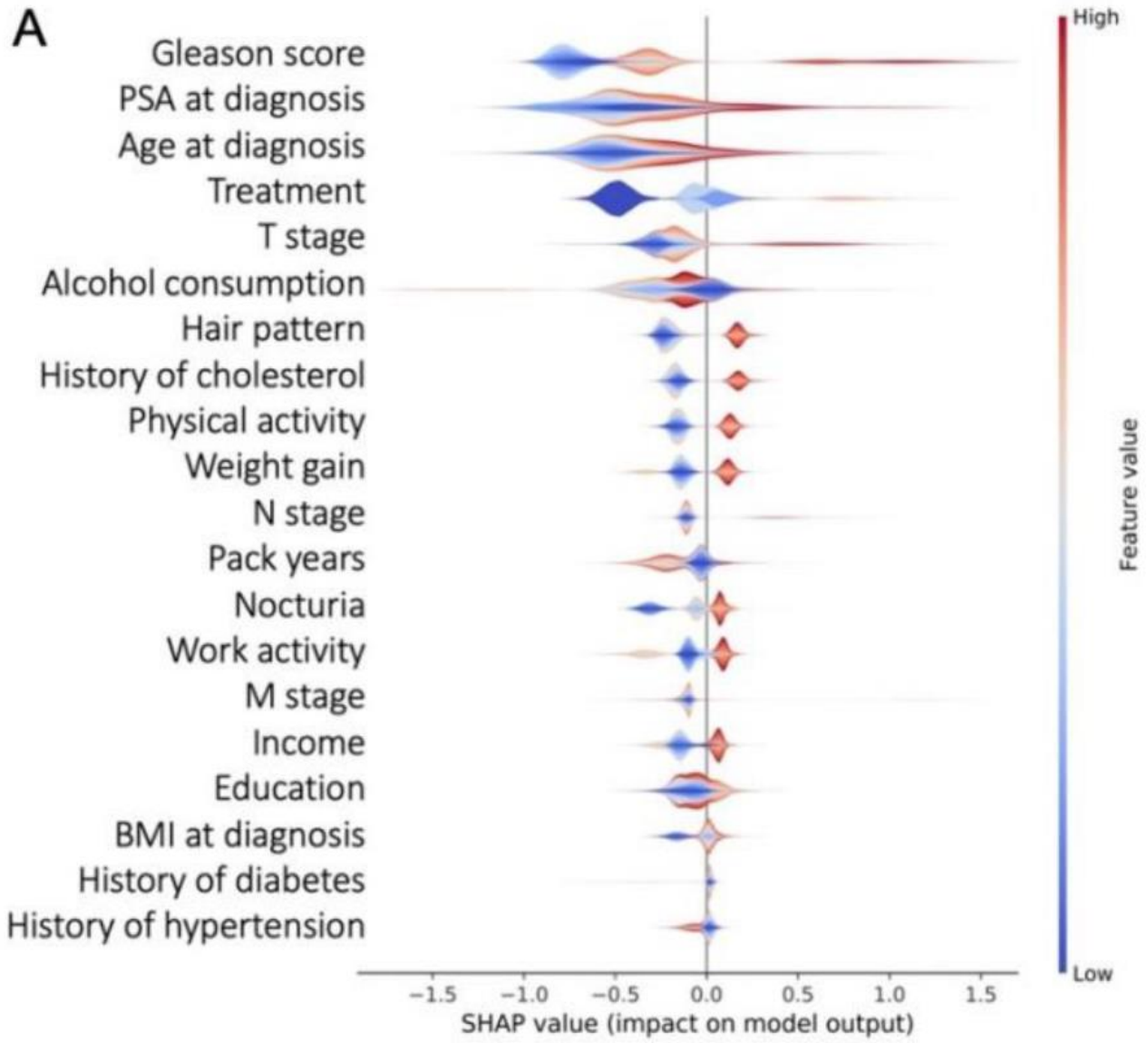

Fig. 5. The 20 most important features [25]

Higher Gleason score, PSA levels and age at diagnosis have a higher Shapley value, which means it has a greater risk of prostate cancer mortality (**Fig. 6 B-D**).

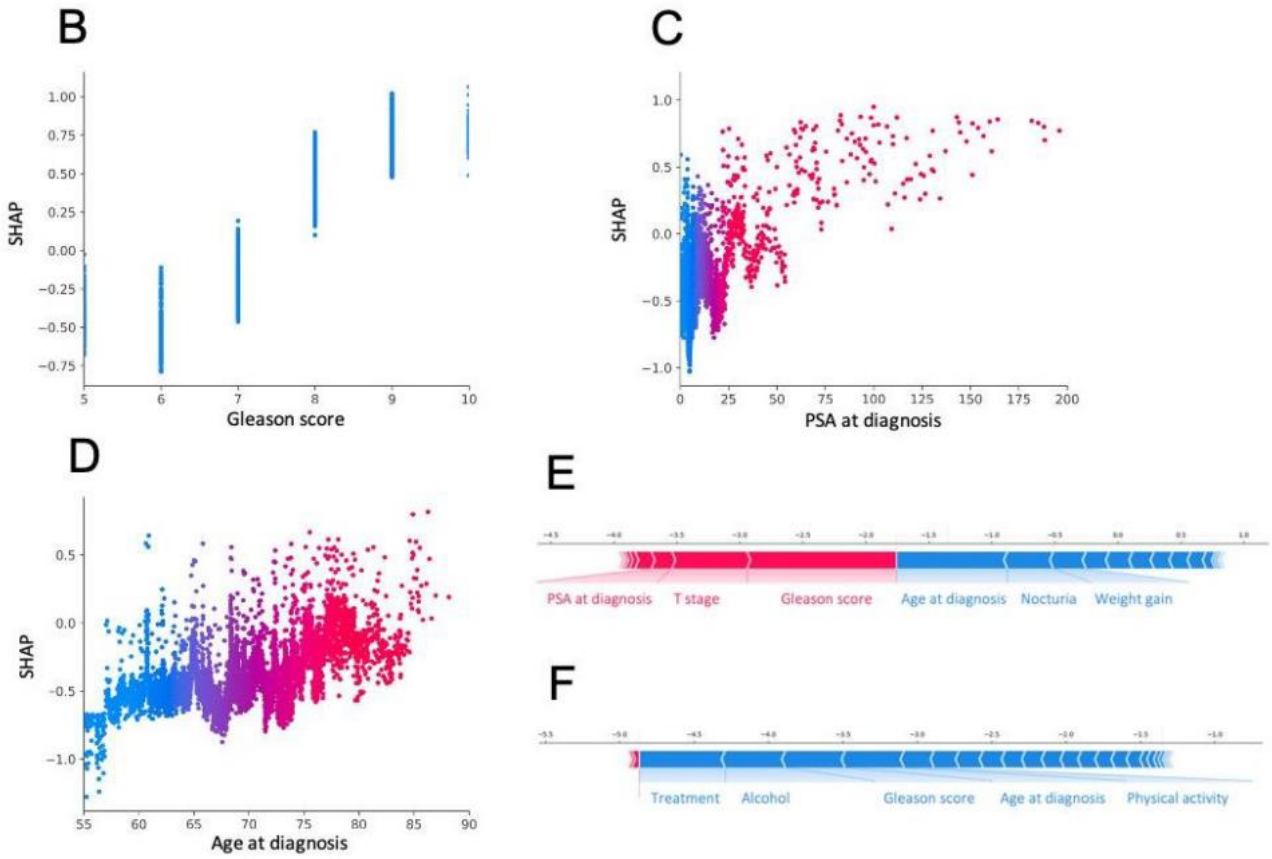

Fig. 6. Population (**B-D**) and individual (**E,F**) level interpretability [25]

1.6. Predicting high-risk prostate cancer using machine learning methods

Henry Barlow et al. (2019) aimed to test the machine learning methods for prostate cancer screening using various clinical measurements [26]. This research also determined the effect on the model's accuracy for variables such as BMI, race, rate of change and age. The total size of the dataset was 21171 patients with 1130 prostate cancer diagnosed individuals of whom 190 have been classified as high-risk prostate cancer (HRPC) patients. High-risk cancers are cancers where the cancer cells spread fast, which leads to a higher possibility of mortality. The definition of such cancers was: Gleason score >7, PSA ≥ 20 ng/mL.

The authors developed two machine learning models: one for testing the presence of cancer and another one to test the presence of high-risk cancer. For the first one, the dataset was labelled in such way: patients without prostate cancer were labelled negative, while patients with low or high-risk prostate cancer were labelled positive. For the second model, patients without prostate cancer or with low-risk cancer were labelled negative, while patients with high-risk cancer were labelled positive.

Data pre-processing techniques have been used. The overall rate of change (ROC) and recent ROC were calculated for PSA. Also, the handling of missing values was done, imbalance correction and scaling methods were also performed using various different methods. K-neighbours (KN), support vector machine (SVM), decision tree (DT), random forest (RF), multi-layer perceptron classifier (MLPC), adaptive boosting (ADA) and quadratic discriminant analysis (QD) models have been implemented. To evaluate the classifiers, training (75%) and test (25%) datasets have been used with holdout and 10-fold cross-validation. Metrics were such: accuracy, AUC, confusion matrices,

sensitivity, specificity, positive predictive value (PPV), negative predictive value (NPV) and F1 score.

After authors reported average accuracy and average AUC using different data imbalance methods, SVMSMOTE was chosen as the sampling method. Nine scaling methods were also tested, and standard scaling methods were chosen as their AUC and accuracy were acceptable by the authors.

ADABoost algorithm performed the best for this dataset (**Table 2**), given that it had the best AUC in holdout set and is only 0.002 off the best in cross-validation. Its accuracy in both was also no more than 0.076 from the top accuracy.

Table 2. Average accuracy and AUC scores for each machine learning algorithm on PoPC

	KN	**SVM**	**QD**	**DT**	**RF**	**MLPC**	**ADA**
Holdout accuracy	0.886	0.899	0.916	0.831	0.831	0.850	0.846
Holdout auc-score	0.683	0.653	0.577	0.777	0.772	0.791	0.777
10-fold cross validation accuracy	0.876 (±0.009)	0.894 (±0.013)	0.916 (±0.009)	0.838 (±0.023)	0.835 (±0.024)	0.845 (±0.015)	0.843 (±0.014)
10-fold cross validation auc	0.674 (±0.038)	0.662 (±0.049)	0.575 (±0.049)	0.778 (±0.030)	0.771 (±0.024)	0.771 (±0.037)	0.776 (±0.028)

To predict high-risk prostate cancer, ADABoost algorithm showed the highest AUC according to cross-validation and the third highest according to the holdout set (**Table 3**).

Table 3. Average accuracy and AUC score for each machine learning algorithm on PoHRPC

	KN	**SVM**	**QD**	**DT**	**RF**	**MLPC**	**ADA**
Holdout accuracy	0.979	0.926	0.930	0.906	0.930	0.905	0.929
Holdout auc-score	0.551	0.674	0.630	0.687	0.653	0.618	0.664
10-fold cross validation accuracy	0.979 (±0.007)	0.925 (±0.011)	0.941 (±0.011)	0.927 (±0.016)	0.915 (±0.028)	0.909 (±0.030)	0.894 (±0.013)
10-fold cross validation auc	0.576 (±0.082)	0.686 (±0.108)	0.617 (±0.098)	0.669 (±0.086)	0.696 (±0.115)	0.675 (±0.114)	0.711 (±0.120)

1.7. Machine learning approach vs. prostate-specific antigen density and prostate-specific antigen velocity

Satoshi Nitta et al. compared several machine learning methods in predicting prostate cancer versus the traditional prostate-specific antigen based screening for prostate cancer [27]. The authors start with a statement that those traditional approaches are widely performed but the accuracy is unsatisfactory. They discuss that artificial neural networks show high AUC values compared with PSA alone, ranging from 0.67 to 0.87 depending on the selected variables and the examined population.

A total of 512 patients were analysed who had prostate biopsy, out of those, 193 (37.7%) were diagnosed with prostate cancer. The mean PSA level, PSA distribution, PSA density and PSA velocity were not significant between patients with and without prostate cancer. The mean prostate volume was found to be significant as it was higher in patients with prostate cancer than those without one. To predict prostate cancer, age, PSA level (max, min, median, mean and variance), prostate volume, white blood cell count and result of biopsy have been used.

Three machine learning methods have been constructed by the authors: artificial neural network, support vector machine and random forest. Performance of these models were evaluated using ROC curve and AUC.

In predicting prostate cancer on the first biopsy, all three machine learning methods are above the PSA level, PSA density and PSA velocity in ROC curve comparison (**Fig. 7**). Artificial neural network had 0.69 AUC value and was superior to the random forest and support vector machine as those had 0.64 and 0.63 AUC, respectively. The AUC values of the PSA level, PSA density and PSA velocity were 0.53, 0.41 and 0.55, respectively.

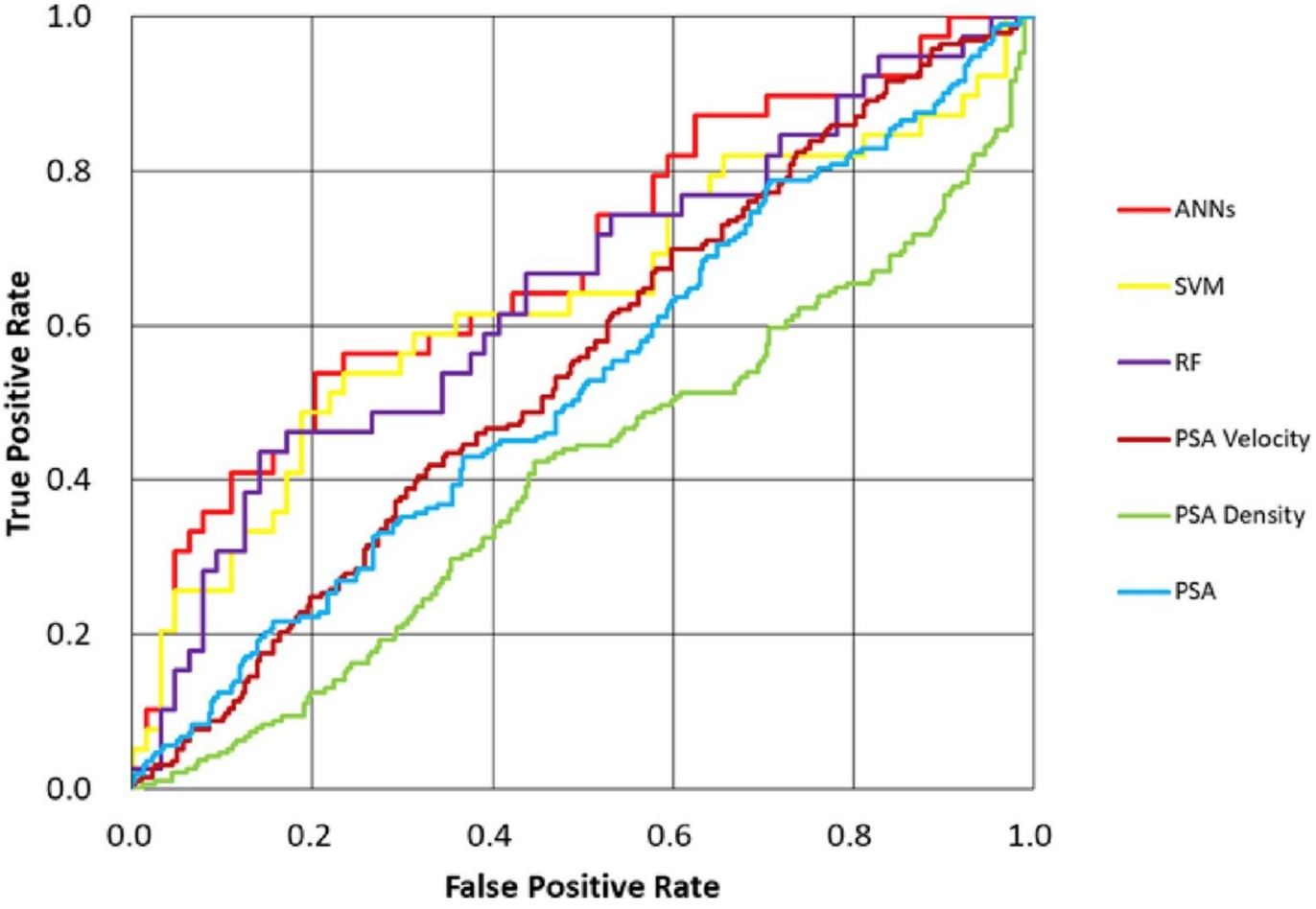

Fig. 7. ROC curve for prediction of prostate cancer on the first therapy [27]

Prediction of the results of second or more biopsies have also been done. Similar results are again received, where all three machine learning methods outperform PSA-based approaches (**Fig. 8**). AUC values of the three machine learning methods are such: ANN – 0.70, RF – 0.68 and SVM – 0.71.

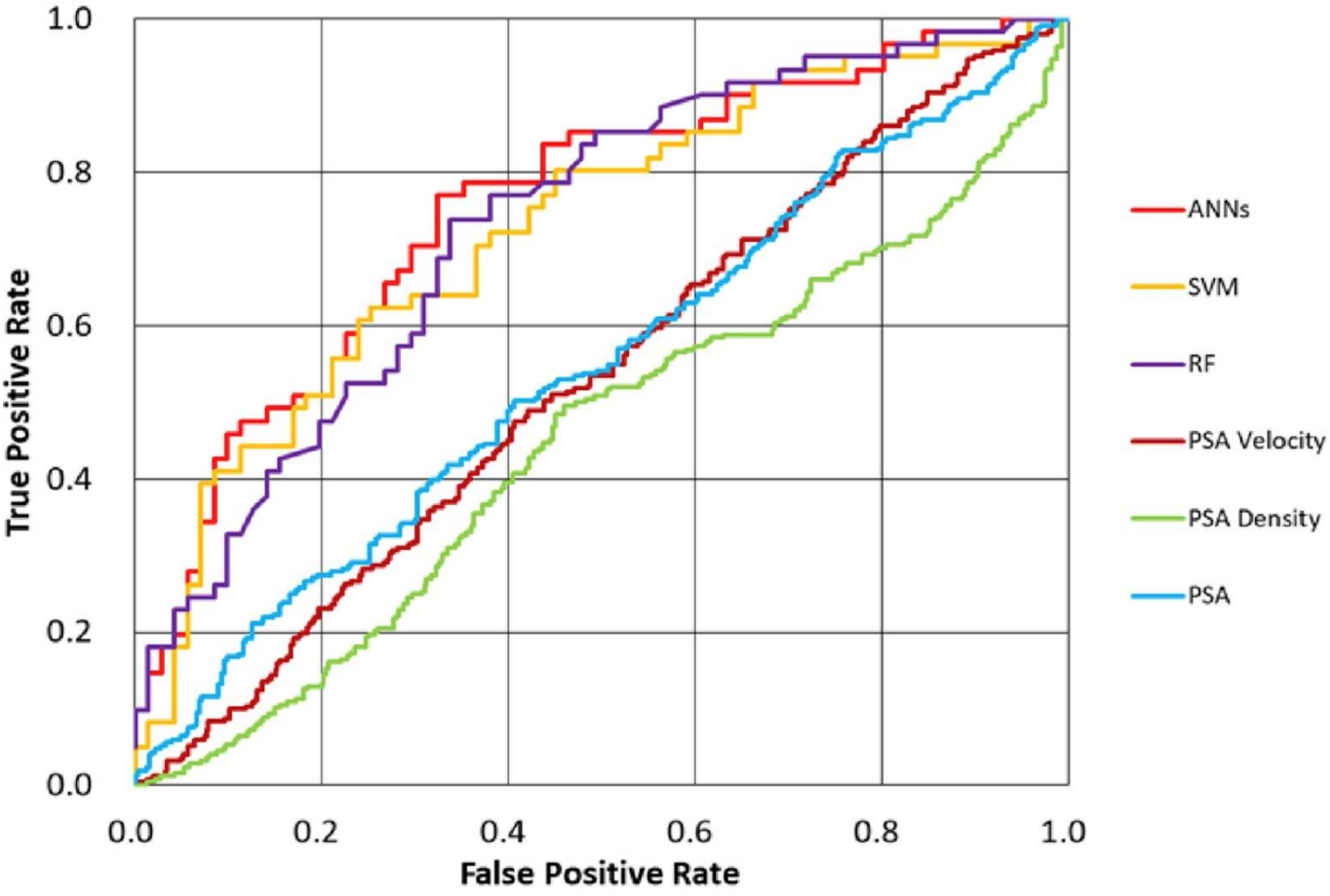

Fig. 8. ROC curve for prediction of prostate cancer on the first and second therapy [27]

1.8. Incorporating artificial intelligence in urology: supervised machine learning algorithms demonstrate comparative advantage over nomograms in predicting biochemical recurrence after prostatectomy

Yu Guang Tan et al. published an article with the objective being building a machine learning algorithm to predict biochemical recurrence (BCR) after radical prostatectomy (RP) and compare with more classical methods [28].

1130 patients who underwent RP were analysed; a split of 70/30 ratio was done. Authors chose 3 ML algorithms for prediction: Naïve Bayes (NB) classifier, random forest (RF) and support vector machine (SVM). Model evaluation metrics have been chosen as accuracy and area under the curve (AUC). BCR has been evaluated at 3 periods: short-term of 1 year, intermediate-term of 3 year and long-term of 5 year. Next to that, the authors compared these 3 ML models with logistic regression and classical nomograms.

Various patient characteristics have been used in models, those are: age, race (Chinese, Malay, Indian, Eurasian, others), BMI, PSA at diagnosis, percentage of cores positive, prostate biopsy grade group, tumour locality, prostate volume, largest tumour diameter, extraprostatic extension, seminal vesicle invasion, surgical margins, nodal disease, perineural invasion, variant ductal histology, pathological grade group.

After applying the 3 ML models to the training and validation sets, robust prediction score of $AUC > 0.83$ and accuracies > 0.82 have been found across all three models (**Table 4**, **Table 5**, **Table 6**). ML models showed a good consistency of the AUC and retained good prediction scores at 5 years on validation set (NB: 0.894, RF: 0.888, SVM: 0.855). The authors conclude that prediction scores across all models were homogenous, which represented inter-model validity and the reliability of incorporating ML as a tool in general.

Table 4. 1-year BCR (n = 1130)

	Training set (790)		Validation set (340)	
Model	**Accuracy**	**AUC (95% CI)**	**Accuracy**	**AUC (95% CI)**
Naïve Bayes	0.894	0.898 (0.850-0.945)	0.894	0.881 (0.814-0.947)
Random Forest	0.958	0.904 (0.847-0.962)	0.950	0.846 (0.751-0.941)
SVM	0.946	0.838 (0.728-0949)	0.932	0.835 (0.742-0.927)
Logistic regression	0.942	0.843 (0.772-0914)	0.926	0.797 (0.724-0.870)
JHH				0.820 (0.766-0.875)
CAPSURE				0.706 (0.637-0.775)
KATTAN				0.815 (0.771-0.859)

Table 5. 3-year BCR (n = 895)

	Training set (625)		Validation set (270)	
Model	**Accuracy**	**AUC (95% CI)**	**Accuracy**	**AUC (95% CI)**
Naïve Bayes	0.826	0.863 (0.814-0.911)	0.851	0.876 (0.815-0.936)
Random Forest	0.869	0.883 (0.849-0.917)	0.888	0.875 (0.825-0.926)
SVM	0.815	0.833 (0.778-0.889)	0.859	0.850 (0.788-0.912)
Logistic regression	0.834	0.837 (0.777-0.896)	0.862	0.848 (0.788-0.907)
JHH				0.757 (0.714-0.800)
CAPSURE				0.720 (0.676-0.765)
KATTAN				0.798 (0.765-0.830)

Table 6. 5-year BCR (n = 698)

	Training set (488)		Validation set (210)	
Model	**Accuracy**	**AUC (95% CI)**	**Accuracy**	**AUC (95% CI)**
Naïve Bayes	0.823	0.860 (0.825-0.895)	0.829	0.894 (0.849-0.940)
Random Forest	0.852	0.884 (0.856-0.912)	0.838	0.888 (0.835-0.941)
SVM	0.817	0.846 (0.806-0.886)	0.810	0.855 (0.800-0.917)
Logistic regression	0.830	0.847 (0.801-0.893)	0.757	0.862 (0.804-0.919)
JHH				0.750 (0.706-0.793)
CAPSURE				0.749 (0.706-0.792)
KATTAN				0.799 (0.765-0.834)

1.9. A comparison of various supervised machine learning techniques for prostate cancer prediction

Research on many machine learning method applications to predict prostate cancer was done by Ebru Erdem and Ferhat Bozkurt [29]. The aim of their study was to compare various supervised machine learning algorithms such as support vector machines (SVM), random forest (RF), k-nearest neighbour (kNN), logistic regression (LR), linear regression (LR), naïve Bayes (NB), linear discriminant analysis (LDA), linear classification (LF), multi-layer perceptron (MLP) and deep neural network (DNN) to predict prostate cancer. The complete research flow is shown in **Fig. 9**.

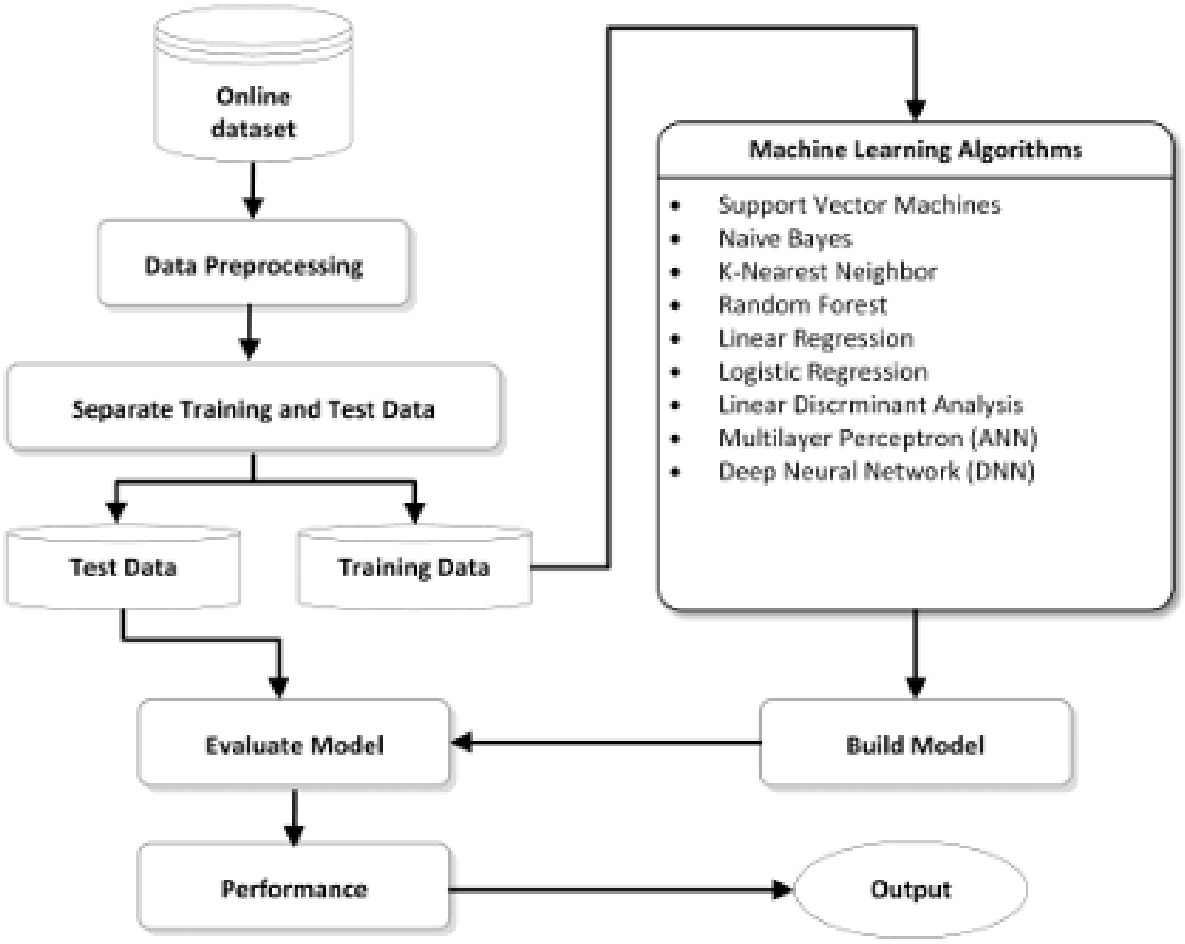

Fig. 9. Workflow of the study [29]

The dataset with 100 patients has been used with 8 independent variables: radius, texture, area, perimeter, compactness, smoothness, fractal dimension and symmetry. The authors split the dataset into training and test sets; samples were randomly selected for training 70% and test 30% data.
From the results shown in **Table 7**, it is clear that MLP shows the best performance for this dataset, having 95.8% AUC, 97% accuracy and 97% F1-score.

Table 7. Performance comparison of the observed machine learning algorithms on prostate cancer dataset

Name of the algorithm	Precision (%)	Recall (%)	AUC (%)	Accuracy (%)	F1-Score (%)
kNN	88	83	83.3	83	86
SVM	94	89	90.3	90	91
Logistic Regression	73	92	84.7	83	81
NB	83	94	86.2	87	88
RF	95	91	88.5	90	93
Linear Regression	89	84	87.6	83	86
LDA	100	81	90.5	87	89
MLP	95	100	95.8	97	97
MLP-Regressor	90	90	90.3	90	90
DNN	89	94	88.9	90	92

1.10. An overview of conducted research

After reviewing the literature, it is easy to understand the relevance of our research topic – prostate cancer, which causes a substantial amount of mortalities and incidences both in Lithuania and around the globe. During the review, we have noticed there are many factors, variables or ways how researchers analyse and predict mortality from prostate cancer. Not only that, but also the response interest can vary from paper to paper, we have found papers analysing biochemical recurrence, metastasis, cancer specific mortality, overall mortality and high-risk prostate cancer within various time frames. Due to this, we have found that authors struggle to compare their results with other works 1-to-1, which is why our work comparison with other papers can also be problematic.

Similar data has been already analysed by us with classical methods [12]. In that research we used a similar amount of data from the same source but with lower number of features, also biochemical recurrence and metastasis have not been explored at the time. We used classical models – semi-parametric Cox proportional hazards regression and Fine-Gray competing risk regression to predict cancer specific mortality and death from other causes. The Cox model had 0.771 and 0.675 AUC for training and testing datasets respectively, while Fine-Gray model had 0.767 and 0.658 AUC values.

As we have already seen in our previous work, most of the research on prostate cancer prediction is done using either Cox or Fine-Gray model, while other traditional machine learning techniques are left out. Implementing traditional machine learning algorithms requires more work as to get comparable results to classical methods for survival analysis, researchers must do additional steps, such as discrete-time modelling. Some of the reviewed literature already highlights the predictive power of machine learning algorithms on this topic.

2. Methods

2.1. Survival Analysis

Survival analysis lets us analyse time-to-event data where we have data until a specific event occurs or we have a duration of time between events. This technique helps estimating survival function and hazard function of a population. The main challenge of such data is the presence of unobservable data after some time or observations where the event has not yet been experienced [30]. One of the important aspects of survival analysis data, which helps with the challenge mentioned previously, is censoring [3; 30]. This is crucial as censoring helps define incomplete information when the event has not occurred at the time of analysis or the follow-up information was lost. There are three main groups of censoring: right – survival time is less than or equal to the true survival time, left – survival time is greater than or equal to the true survival time and interval – it is known that an event occurred within a specific time interval [30]. In our research, a right censoring has been applied.

The survival function itself gives a probability that an individual (or a group) will survive past a certain point in time. The hazard function measures the risk of an event occurrence at time *t*. These functions are estimated from the observed data, and popular methods are used for that such as semi-parametric Cox proportional hazard regression [11-19], Fine-Gray competing risk regression [12; 14; 20].

The survival function is defined as:

$$S(t) = \Pr(T > t) = 1 - F(t) = \int_t^{\infty} f(x)dx \qquad (1)$$

Where, T – continuous random variable with probability density function *f(t)* and cumulative distribution function $F(t) = \Pr(T \leq t)$.

The survival function monotonically decreases with *t* and the initial value is 1 at $t = 0$, which means that at the start of the study, all 100% observations are alive and no events occurred. Such survival function will give a probability that an individual (or group) is alive at *t*. An example of $S(t), f(t)$ and $F(t)$ is presented in **Fig. 10**.

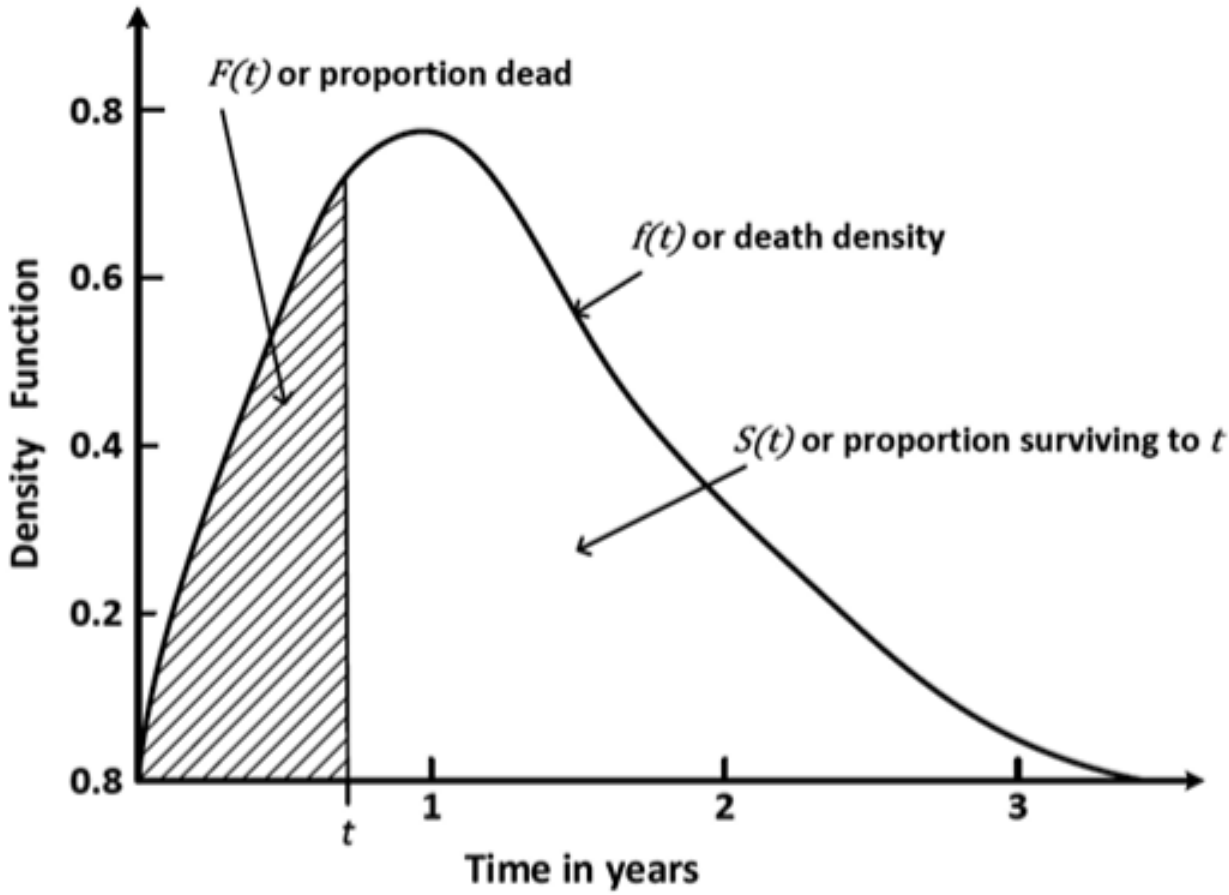

Fig. 10. Example of survival function, probability density function and cumulative distribution function

Another commonly used function is the hazard function which represents the likelihood of the event occurring at *t* time knowing that no event has yet occurred up to *t*. The hazard function is defined as:

$$\lambda(t) = \lim_{dt \to 0} \frac{\Pr\,(t < T \leq t + dt | T > t)}{dt} = \frac{f(t)}{S(t)} = -\frac{d}{dt} \log S(t). \qquad (2)$$

2.2. Discrete-time modelling

Survival prediction models are built on top of data, which is time-to-event data. This data either has a time indicator of event occurrence or the last follow-up. In survival analysis, a discrete-time modelling is a technique used for data analysis when event (e.g. death, illness, failure) occurs at different discrete time periods instead of continuously during a period [2]. This type of modelling is especially useful when the time between events is unequal or when it is a challenge to determine exact time of event occurrence. When building a survival prediction model, specific models have to be used which were built for time-to-event data and have certain assumptions inside them. This is not the case for discrete-time modelling where any ML classifier can be applied. One of the main discrete-time modelling advantages are time dependant variable inclusion, those variables can change over time and can be associated with the target variable. This violates continuous-time models where the assumption is made that the covariate effect is constant during a time period.

Discrete-time data can be easily obtained from continuous-time data. For each person in a dataset (in the case of patient data), follow-up time can be split into a set of pre-defined time intervals and for

such discrete time interval, an event indicator is set. Having such data, any ML classifier can be applied as then the binary outcome is experiencing an event within a specific time interval.

Suppose we have a continuous-time survival data, where one record holds one individual's time to an event (or latest follow up time), we divide the time into *J* intervals $(t_0, t_1], (t_1, t_2], \dots, (t_{J-1}, t_J]$, where $t_0 = 0$. Patient's hazard in interval $A_j = (t_{j-1}, t_j]$, having covariates X_i, can be expressed as the conditional probability:

$$\lambda_{ij}(X_i) = \Pr\left(T_i \in A_j \middle| T_i > t_{j-1}, X_i\right) = \Pr\left(t_{j-1} < T_i \leq t_j | T_i > t_{j-1}, X_i\right) \quad (3)$$

And the discrete probability function:

$$f_{ij} = \Pr\left(T_i \in A_j \middle| X_i\right) = S\left(t_{j-1} \middle| X_i\right) - S(t_j | X_i) \quad (4)$$

Survival probability in discrete-time can be obtained in a similar manner as in continuous-time. The probability to survive past time *t* can be obtained as the product of the conditional survival probabilities for all time intervals up to and including $(t_{J-1}, t_J]$, such that $t_j \leq t$:

$$S_i(t|X_i) = \Pr(T_i > t | X_i) = \prod_{j:t_j \leq t} (1 - \lambda_{ij}(X_i)) \quad (5)$$

2.3. Prediction models

2.3.1. Logistic regression

Logistic regression is one of the many statistical methods used for classification task. It is used to predict binary outcome probability, e.g. a probability that an email is spam or no, a probability that patient is sick from a certain illness or not.

In logistic regression, a relationship between a dependant variable (outcome variable) and one or several independent variables is modelled with the use of the logistic function. It's a type of sigmoid function which accepts any input and outputs a value between 0 and 1, and interpretable as a probability.

Logistic regression goal is to find optimal independent variable coefficients (weights) so that the predicted outcome probability is close to the observed results. It is done by optimizing the loss function, e.g. *log* loss, which measures difference between predicted probabilities and observed results.

The logistic function is defined as:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (6)$$

Where *x* is the function input.

For example, if *x* is a very large value (either positive or negative), logistic function output will be respectively close to 1 or 0. If *x* is 0, the output will be 0.5.

Logistic regression model is defined as follows:

$$P(y|x) = f(w_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n) \quad (7)$$

Where *P(y|x)* is a probability that observation with covariates $x_1, x_2, \ldots, x_n$ belongs to class *y*, $w_0, w_1, \ldots, w_n$ are weight (or coefficients), *f(x)* is a logistic function.

2.3.2. Random forest

Random forest is a type of supervised machine learning methods which can be used in binary outcome classification prediction. Random forest is composed of many decision trees. An individual decision tree is similar to a scheme of tree structure, and this decision tree is used to make predictions based on a set of covariates. Each tree's internal node means a solution based on one of the attribute values, each leaf note means a prediction. To predict an observation, the tree goes from root node to the leaf node, following the rules of each internal node.

In a random forest, each decision tree prediction is combined to obtain the final outcome. This combination can be made in several ways, one of the simplest ways – using the majority rule of trees. For example, if a random forest is used in binary classification and 60% of trees predict that an observation falls into class 0 while 40% of trees predict class 1, random forest prediction will be that observation belongs to class 0.

The random forest training process covers many decision tree creations using various training data and function subsets. It is done repeating such process:

1. Take a random subset of training data.
2. Train a decision tree using sampled training data.
3. Repeat this process to obtain many decision trees.

The idea behind such flow is that each decision tree will make slightly different predictions due to the randomness of data and attributes it was trained on, and the combination of all tree predictions will be more accurate than any individual tree.

The following formula shows a general random forest model structure for binary classification:

$$\hat{y} = \arg\max_{k \in 0,1} \frac{1}{T} \sum_{t=1}^{T} [\hat{y_t} = k] \quad (8)$$

Where $\hat{y}$ is the final random forest prediction, T is the number of decision trees in a forest, $\hat{y_t}$ – prediction of *t*-th decision tree, $k \in 0,1$ are class values.

The term $[\hat{y_t} = k]$ is equal to 1 if $\hat{y_t} = k$ and 0 otherwise. Formula calculates how many times, on average, each class is predicted by different trees, and the final predicted class is selected as the one having the highest average.

2.3.3. XGBooost

XGBoost (eXtreme Gradient Boosting) is a type of gradient boosting algorithm which can be used in classification tasks [31]. Gradient boosting is a supervised learning method covering training a set of

models to make a prediction when each model is being trained to fix earlier model's mistakes. XGBoost individual models are decision trees.

XGBoost takes advantage of boosting strategy, where trees are built sequentially in such a way that a new tree aims to reduce the errors of the previous one. It can be explained in a few simple steps:

- Initial model F_0 is defined to predict response y.
- A new model h_1 is fit for the residuals $y - F_0$.
- The boosted version of the F_0 model is then created - F_1, this model is now a combination of the initial model F_0 and h_1. Since h_1 is fit on the residuals of F_0, the error of F_1 will be lower than F_0.

$$F_1(x) = F_0(x) + h_1(x) \quad (9)$$

This process can go on for *m* iterations to improve the model and minimize the residuals as much as possible:

$$F_m(x) = F_{m-1}(x) + h_m(x) \quad (10)$$

Here *x* is our covariates.

XGBoost model training process is done by repeating such steps:

1. Train a decision tree using current training data.
2. Measure loss function gradient with respect to decision tree prediction.
3. Update training data with the addition of a negative gradient value to the decision tree predictions. This is done to increase the weight of observations which have been falsely predicted by the decision tree.
4. Repeat this process to obtain many trained decision trees.

2.3.4. Neural Networks

Neural networks or multi-layer perceptron is a supervised learning algorithm with the goal of learning such functions from training data:

$$f(\cdot): R^n \rightarrow R^o \quad (11)$$

Where *n* – input dimension count, *o* – output dimension count.

Having covariates $X = x_1, x_2, \dots, x_n$ and target *y*, neural networks can learn non-linear function approximator for classification or regression task. This is different from logistic regression in such a way that there can be one or more non-linear layers between the input and output layer, called hidden layers. **Fig. 11** shows single layer perceptron and **Fig. 12** gives an example of a neural network with one hidden layer.

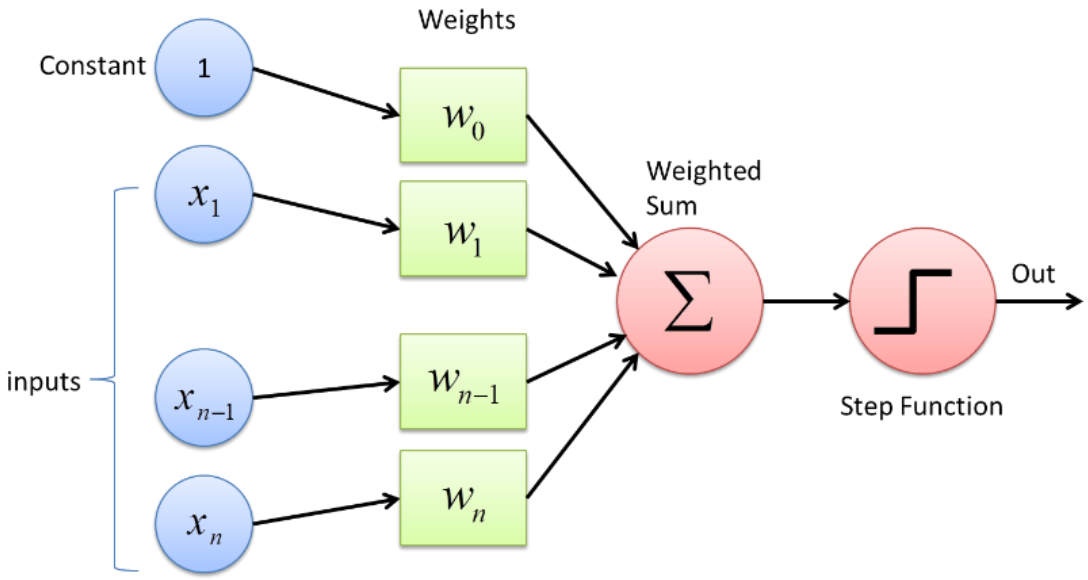

Fig. 11. Perceptron

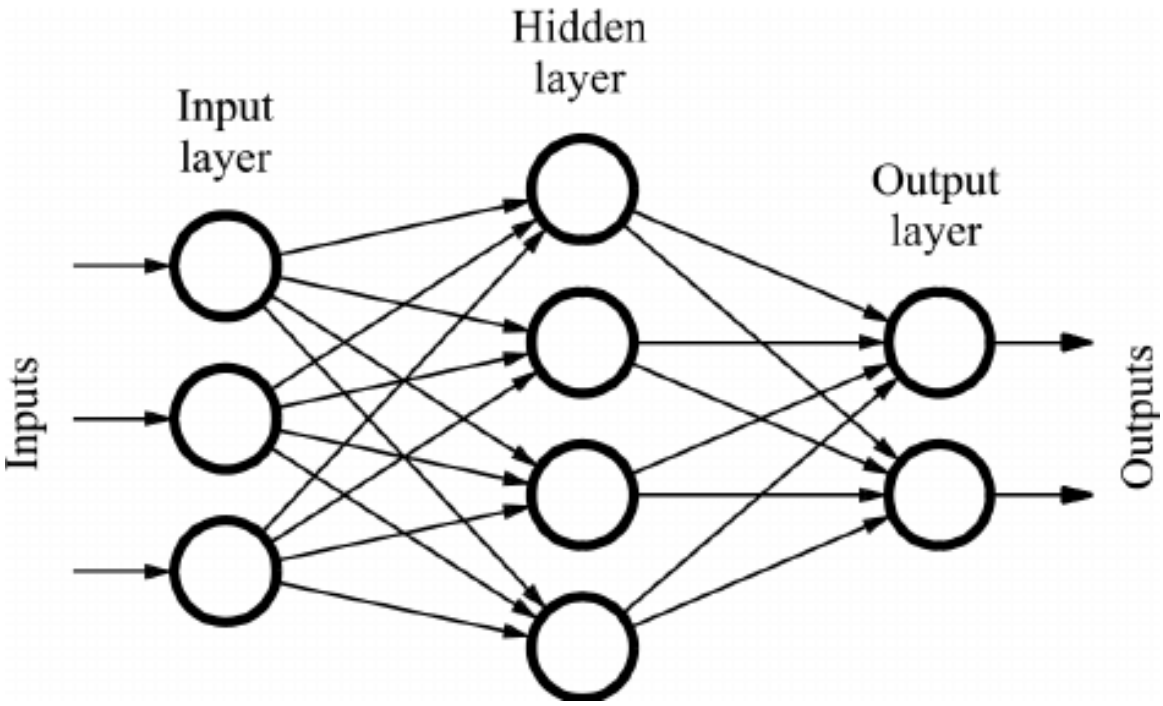

Fig. 12. Neural network example with one hidden layer

In these figures, the most left layer is an input layer consisting of neurons $\{x_1, x_2, \dots, x_n\}$ (covariate values). Each neuron in the hidden layer transforms values from the previous layer with linear summation: $w_1x_1 + w_2x_2 + \cdots + w_nx_n$, after which non-linear activation function $g(\cdot): R \to R$ is used, for example, a hyperbolic tangent function. The last layer is the output layer which transforms values into outputs (predictions).

2.4. Evaluation method

2.4.1. Receiver operating characteristic curve

An receiver operating characteristic (ROC) curve is a type of graph which shows the performance of the classifier at all classification thresholds [32]. It is commonly used both in medical decision making and machine learning research [32]. This graph measures true positive rate (TPR) and false positive rate (FPR) which are defined as:

$$TPR = \frac{TP}{TP + FN} \quad (12)$$

$$FPR = \frac{FP}{FP + TN} \quad (13)$$

Where TP – true positives (correctly predicted the positive class), FN – false negatives (incorrectly predicted the negative class), FP – false positives (incorrectly predicted the positive class), TN – true negatives (correctly predicted the negative class).

Lowering the classification threshold will result in more classified observations as positives, increasing both FP and TP. A typical ROC curve graph looks like this:

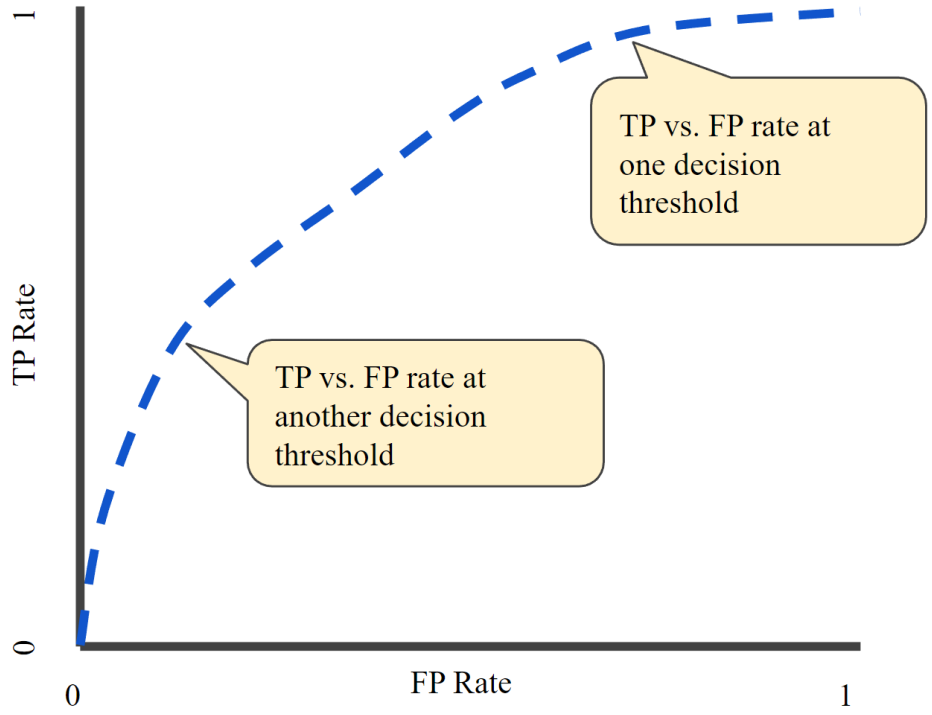

Fig. 13. Example of ROC curve

There are several points in this graph that represent a very specific type of model's performance. The point (0, 0) represents a model which never gives a positive classification, this means it outputs no false positive errors but also no true positives. The opposite is (1, 1) point. The point (0, 1) shows a perfect classification while $y = x$ diagonal line shows a model as good as a random guess [32].

2.4.2. Area under the ROC curve

Area under the ROC curve (AUC) measures the entire two-dimensional area underneath the entire ROC curve, which is also sometimes called as c-statistic or concordance index [2]. This reduces the ROC performance to a single scalar value. AUC lets us measure the performance of the model across all possible classifier thresholds. AUC ranges from 0 to 1, with 0 representing a model whose predictions are 100% false, while 1 represents a model whose predictions are 100%. Any real model should have AUC above the 0.5, as 0.5 represents a model no better than a random guess. The AUC of a classifier is equals to a probability that the classification model will rank a random positive observation higher than random negative observation [32]. This metric is scale-free as it measures how the predictions are ranked and not their absolute values. It is also classification-threshold-free as it measures the quality of the model's predictions not only with a single threshold.

2.5. Structure of the research

2.5.1. Research Flow

Research begins with data preparation: relevant column selection, removing rows with missing values, creating "death_from_other_causes" and "patient_id" column. A random 80/20% stratified split based on overall mortality of patients is made to obtain training and testing data. Testing data is not used throughout any step of the model training, evaluation or hyperparameter optimization, this subset of data is only used to evaluate the final model (see **Fig. 14**).

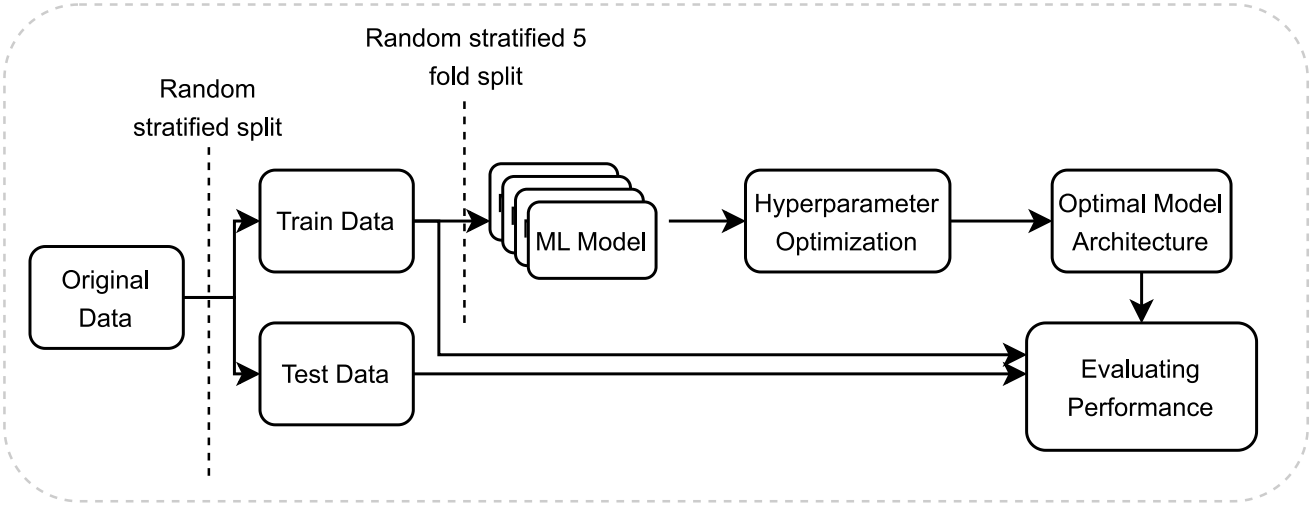

Fig. 14. Complete research flow

2.5.2. Hyperparameter optimization

For each machine learning model (logistic regression, random forest, XGBoost, neural network) and each response variable (cancer specific mortality, death from other causes, biochemical recurrence, metastasis) a hyperparameter optimization based on Bayesian optimization was performed, which uses the results from the previous iteration for hyperparameter value sampling improvement [3; 33]. A complete flow of this approach is presented in **Fig. 15**.

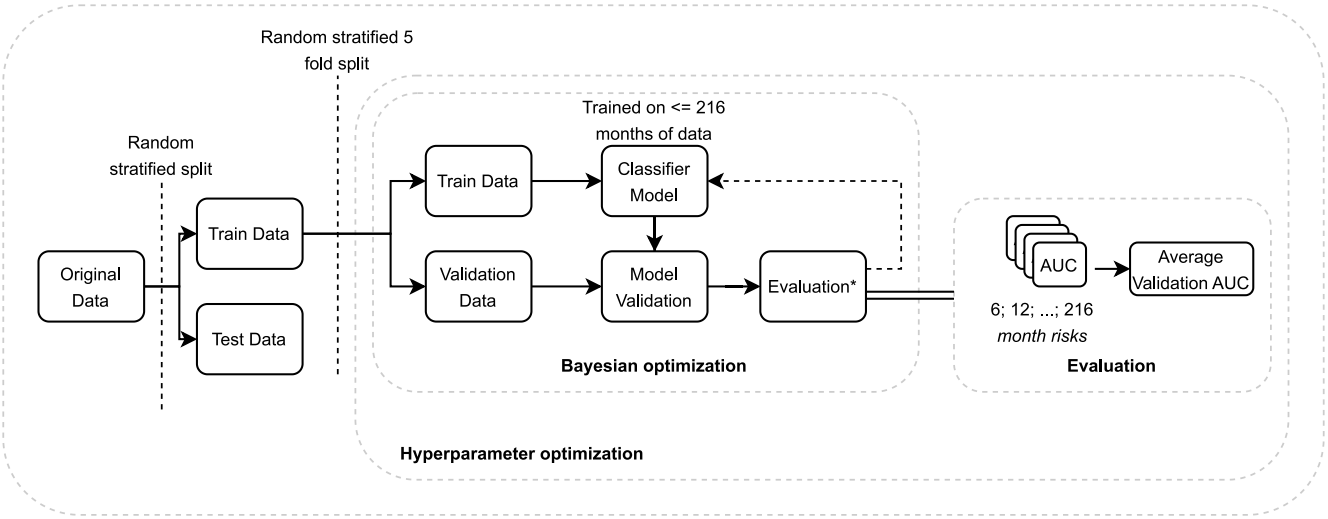

Fig. 15. Hyperparameter optimization

One important part to mention is the evaluation process inside the Bayesian optimization. Unlike most built-in packages for machine learning, which evaluates the model performance on discrete-time data by default, we developed an evaluation strategy to measure the performance of the model on continues-time data. This way, the optimization process minimised the loss function, which was a negative of average validation AUC.

2.5.3. Evaluation of the models

Each developed model is evaluated the same way for every response variable. Once the appropriate model's hyperparameters are selected, training of the model is performed with different amount of data. Models are trained using a subset of continuous-time training data having $\leq 24, \leq 36, \ldots, \leq 216$ observed survival data, this lets us obtain models for short, medium and long-term prognosis. Each model is then validated on predicted 6, 12, 18, 24 … month risks using subsets of training and testing data. All of the AUC values obtained throughout the evaluation are then averaged to obtain single metric, which will be compared across different models and architectures. This flow is shown in **Fig. 16**.

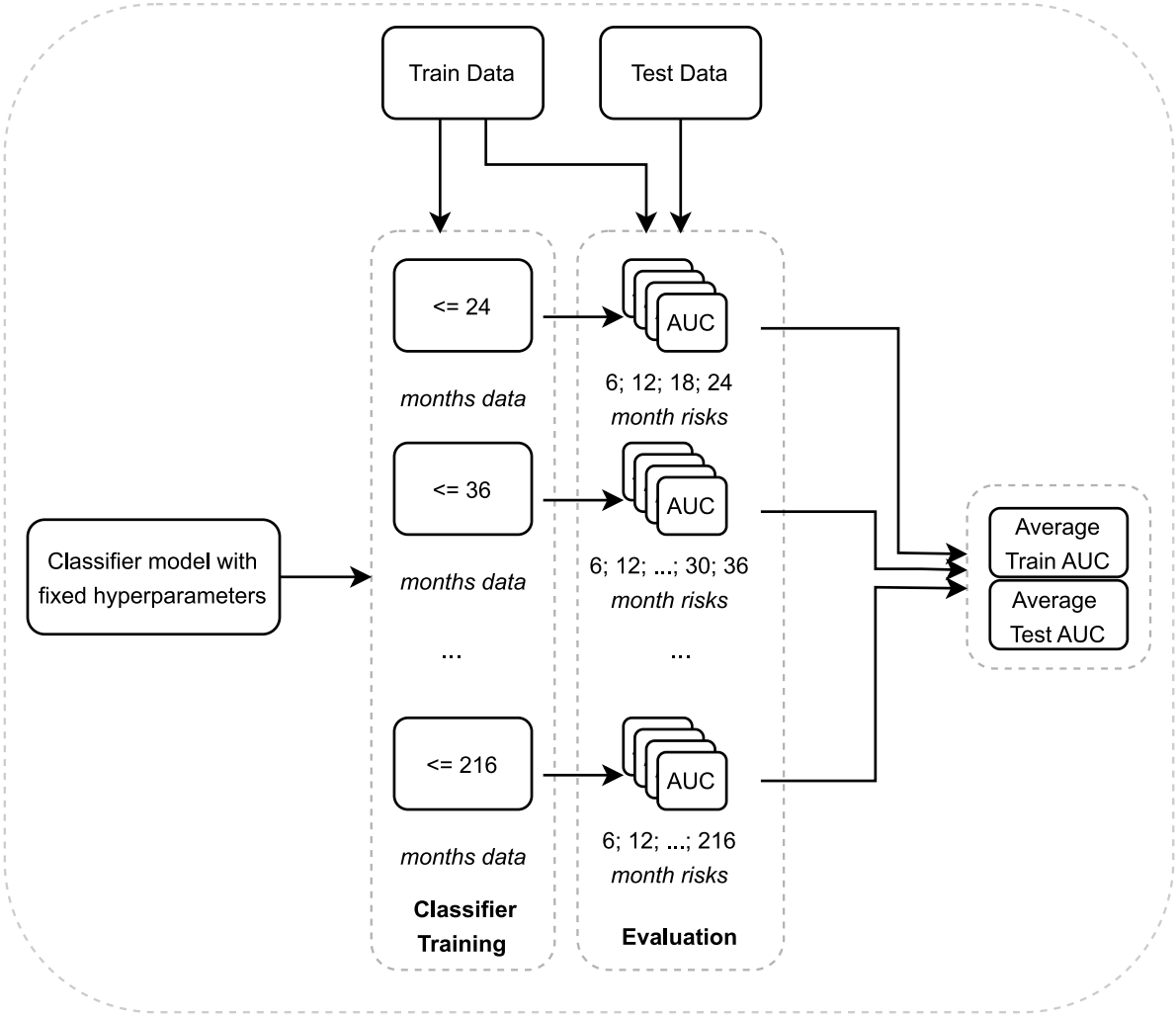

Fig. 16. Flow of model evaluation

2.6. Software

This work makes use of many various statistical methods, machine learning models and pre-processing techniques. For this reason, a *python* (version: 3.10.4) programming language is used within *Visual Studio Code* (version: 1.76.2) integrated development environment. The project used version control using *Git* and *Github*, whole project can be accessed on https://github.com/vytautas9/prostate-cancer-mortality.

3. Results

3.1. Data

The descriptive characteristics of patients are presented in **Table 17**. The median age of patients was 64 years with minimum of 40 and maximum of 87 years. Median PSA level is 6.4 (IQR 4.8 – 9.7). Out of all 1564 prostate cancer diagnosed patients, 264 (16.9%) patients have died, out of those, 50 (3.2%) died from prostate cancer and 214 (13.7%) from other causes. The median follow-up after RP was 104 months (IQR 65 – 159). 460 (29.4%) of men also experienced biochemical recurrence and 99 (6.3%) experienced metastases.

3.2. Data split

A random 80/20% stratified split of the data was made to obtain train and test datasets. Stratification is performed on overall mortality feature to have a similar proportion of mortalities across datasets. The testing dataset was only used to evaluate the final machine learning models as seen in **Fig. 14**. Descriptive characteristics of patients split across train and test datasets are presented in **Table 18**. The split is stratified on overall mortality feature, that is what we see as well in the characteristics, where the training dataset consist of 211 (16.9%) mortalities and test dataset has 53 (16.9%) mortalities. Worth mentioning that the training set has 42 (3.4%) cancer specific mortalities with a median follow-up 104 months (IQR 64 – 159), while the testing set – has 8 (2.6%) mortalities with a median follow-up of 106 months (IQR 70 – 163).

3.3. Hyperparameter optimization

As mentioned in Hyperparameter optimization subsection, hyperparameter optimization was done for each response variable and machine learning method, based on Bayesian optimization. A sample of experiments is presented in **Table 20** with some experiments removed due to very low accuracy or similar accuracy to other experiments.

Various hyperparameters have been considered in optimization process, such as:

- Random forest: criterion (entropy, gini), the maximum depth of the tree (uniformly between 5 and 20), the number of features to consider for split (root square, log2, number of features), the minimum number of samples required for a leaf node (uniformly between 0 and 0.5), the minimum number of samples require to split (uniformly between 0 and 1), the number of trees (uniformly between 100 and 500)
- XGBoost: the subsample ratio of columns for tree (uniformly between 0.5 and 1), gamma (uniformly between 0 and 1), learning rate (uniformly between 0 and 0.2), the maximum depth of the tree (uniformly between 2 and 20), the minimum sum of instance weight needed for child (uniformly between 1 and 10), the number of trees (uniformly between 100 and 1000), subsample ratio of the training instance (uniformly between 0.5 and 1), L1 regularization term (uniformly between 0 and 0.1), L2 regularization term (uniformly between 0 and 10), balance of positive and negative weights (uniformly between 0 and 1)
- Neural network: hidden layer size (with uniformly between 25 and 100, 8, 16, 32, 64, 128, (8, 8), (16, 16), (32, 16), (32, 32), (16, 8), (128, 64, 32, 16), (32, 16, 16)), alpha (logarithmic normally distributed with $mu = log(10^{-4})$ and $sigma = 1$), activation function (logistic sigmoid, hyperbolic tan, rectified linear unit function (relu)), learning rate schedule for weight updates (constant, adaptive), solver for weight optimization

(stochastic gradient descent, adam, limited-memory Broyden-Fletcher-Goldfarb-Shanno (lbfgs))

Other hyperparameters are set as default by the used method library. Optimal hyperparameters for each ML model and response variable:

- Biochemical recurrence
    - Random Forest: criterion (gini), the maximum depth of the tree (19), the number of features to consider for split (number of features), the minimum number of samples required for a leaf node (0.000055), the minimum number of samples require to split (0.32), the number of trees (240)
    - XGBoost: the subsample ratio of columns for tree (0.2), gamma (0.1), learning rate (0.1), the maximum depth of the tree (4), the minimum sum of instance weight needed for child (3), the number of trees (500), subsample ratio of the training instance (0.2), L1 regularization term (0.001), L2 regularization term (1), balance of positive and negative weights (0.7)
    - Neural network: hidden layer size (34), alpha (0.0058), activation function (logistic sigmoid), solver for weight optimization (adam)
- Cancer specific mortality
    - Random Forest: criterion (entropy), the maximum depth of the tree (3), the number of features to consider for split (0.1), the minimum number of samples required for a leaf node (1), the minimum number of samples require to split (2), the number of trees (400)
    - XGBoost: the subsample ratio of columns for tree (0.2), gamma (0.1), learning rate (0.1), the maximum depth of the tree (4), the minimum sum of instance weight needed for child (3), the number of trees (500), subsample ratio of the training instance (0.2), L1 regularization term (0.001), L2 regularization term (1), balance of positive and negative weights (0.7)
    - Neural network: hidden layer size (32), alpha (0.0002), activation function (hyperbolic tan), solver for weight optimization (adam)
- Death from other causes
    - Random Forest: criterion (entropy), the maximum depth of the tree (16), the number of features to consider for split (number of features), the minimum number of samples required for a leaf node (0.05), the minimum number of samples require to split (0.59), the number of trees (378)
    - XGBoost: the subsample ratio of columns for tree (0.2), gamma (0.1), learning rate (0.1), the maximum depth of the tree (4), the minimum sum of instance weight needed for child (3), the number of trees (500), subsample ratio of the training instance (0.2), L1 regularization term (0.001), L2 regularization term (1), balance of positive and negative weights (0.7)
    - Neural network: hidden layer size (70), alpha (0.00026), activation function (relu), solver for weight optimization (adam)
- Metastasis
    - Random Forest: criterion (entropy), the maximum depth of the tree (3), the number of features to consider for split (0.1), the minimum number of samples required for a leaf node (1), the minimum number of samples require to split (2), the number of trees (400)

- XGBoost: the subsample ratio of columns for tree (0.55), gamma (0.05), learning rate (0.15), the maximum depth of the tree (13), the minimum sum of instance weight needed for child (2), the number of trees (535), subsample ratio of the training instance (0.65), L1 regularization term (0.01), L2 regularization term (0.76), balance of positive and negative weights (0.91)
- Neural network: hidden layer size (8, 8), alpha (0.026), activation function (relu), solver for weight optimization (lbfgs)

3.4. Discrete time modelling example

3.4.1. Dataset transformation

We can visualise and describe discrete time modelling with our data to better understand the workflow. As from **Table 18**, we know that we have 1251 patients in a training set, if we would train a model on ≤ 200 months data, we would firstly need to transform the data into discrete-time data. As mentioned in the Discrete-time modelling section, we divide the survival time into *J* intervals $(t_0, t_1], (t_1, t_2], \ldots, (t_{J-1}, t_J]$, where $t_0 = 0$. Doing such transformation, it explodes our dataset from 1251 rows (1 row per patient) to 134666 rows (multiple rows per patient) and also we add 2 features to the dataset: discrete survival time indicator and discrete event indicator.

We can take the example of a patient who has not died from prostate cancer. Such an example can be a patient 66 years old with the latest follow-up time of 132 months. For such patient, a dataset will be generated:

Table 8. Discrete-time non-event patient data example

Survival months	Cancer specific mortality	Discrete survival time	Discrete cancer specific mortality indicator
132	No	1	No
132	No	2	No
…	…	…	…
132	No	131	No
132	No	132	No

We are training a model on ≤ 200 months of data, and because this patient has a follow-up time of 132 months, there will be 132 rows created for each discrete month timestamp. At each discrete time, an indicator for cancer specific mortality is also created. A different example of a non-event patient with follow-up time longer than 200 months. A patient is 62 years old with follow-up time of 211 months. For such patient, a dataset will be generated:

Table 9. Discrete time non-event patient data with longer follow-up time example

Survival months	Cancer specific mortality	Discrete survival time	Discrete cancer specific mortality indicator
211	No	1	No
211	No	2	No
…	…	…	…
211	No	199	No
211	No	199	No

Because our training range is lower than this patient's follow-up time, the data will be clipped, and in such a dataset, only 200 rows (for each discrete month) will be generated.

The two above examples are of non-event patients, meaning those patients have not been reported dead due to prostate cancer. We can take the example of patients who have died from prostate cancer. For example, a patient 63 years old who died from prostate cancer and has a follow-up time of 25 months. For such patient, a dataset will be generated:

Table 10. Discrete time event patient data example

Survival months	Cancer specific mortality	Discrete survival time	Discrete cancer specific mortality indicator
25	Yes	1	No
25	Yes	2	No
…	…	…	…
25	Yes	24	No
25	Yes	25	Yes

In this case, we can see that the last row is the time when patient death has been reported, an event indicator for this discrete timestamp will be made. Another example would be with a patient who has died due to prostate cancer, but the follow-up time is above our training range:

Table 11. Discrete time event patient data with longer follow-up time example

Survival months	Cancer specific mortality	Discrete survival time	Discrete cancer specific mortality indicator
203	Yes	1	No
203	Yes	2	No
…	…	…	…
203	Yes	199	No
203	Yes	200	No

Even though we know that this person died from prostate cancer, when building a model with ≤ 200 months data, this person would be considered as censored patient who have not yet experienced death (up to 200^{th} month).

3.4.2. Evaluating model

After training a model on discrete-time data and before evaluating the model, the data must be transformed in a slightly different way. As shown in **Table 8** and **Table 10**, when follow-up time is shorter than the model's training range, we will only have rows up to patient's follow-up time. To evaluate the model, we need to extend this data. For example, a patient 68 years old who died from prostate cancer and has a follow-up time of 39 months. To evaluate a model on such patient, the individual's data will transformed in such a way:

Table 12. Discrete time event patient data with cumulative indicator

Survival months	Cancer specific mortality	Discrete survival time	Discrete cancer specific mortality indicator	Cumulative cancer specific mortality indicator
39	Yes	1	No	No
…	…	…	…	…
39	Yes	37	No	No
39	Yes	38	No	No
39	Yes	39	Yes	Yes
39	Yes	40	-	Yes
39	Yes	41	-	Yes
…	…	…	…	…
39	Yes	200	-	Yes

Because this person has a follow-up of 39 months and a cancer specific death, the discrete indicator cannot be used for evaluation and a cumulative indicator is created, which marks the event time and onwards as event times. For the same patient, we can use our model to predict the mortality probability:

Table 13. Discrete time event patient data with discrete mortality probabilities

Survival months	Cancer specific mortality	Discrete survival time	Discrete cancer specific mortality indicator	Cumulative cancer specific mortality indicator	Discrete mortality probability
39	Yes	1	No	No	0.000594
…	…	…	…	…	…
39	Yes	37	No	No	0.000886
39	Yes	38	No	No	0.000896
39	Yes	39	Yes	Yes	0.000906
39	Yes	40	-	Yes	0.000916
39	Yes	41	-	Yes	0.000926
…	…	…	…	…	…
39	Yes	200	-	Yes	0.005388

What the ML model predicted is the probability of a patient to experience cancer specific mortality at the specific month after diagnosis. Out of this probability, a cumulative hazard can be calculated:

Table 14. Discrete time event patient data with cumulative hazard

Discrete survival time	Discrete cancer specific mortality indicator	Cumulative cancer specific mortality indicator	Discrete mortality probability	Negative log probability	Cumulative hazard
1	No	No	0.000594	-0.000595	0.000594
…	…	…	…	…	…
37	No	No	0.000886	-0.000887	0.026691
38	No	No	0.000896	-0.000896	0.027563
39	Yes	Yes	0.000906	-0.000907	0.028444
40	-	Yes	0.000916	-0.000917	0.029335
41	-	Yes	0.000926	-0.000927	0.030234
…	…	…	…	…	…
200	-	Yes	0.005388	-0.005403	0.353876

where $negative\ log\ probability\ = \log_e(1 - discrete\ mortality\ probability)$ and $cumulative\ hazard = 1 - e^{cumulative\ sum\ of\ patient's\ negative\ \log probability}$

3.4.3. Visualising patient's mortality

Once a dataset has been transformed into discrete-time data and cumulative hazards have been calculated, we can visualise the mortality for each patient. A model has been trained on ≤ 216 months data using a random forest algorithm for cancer specific mortality with optimal hyperparameters mentioned in Hyperparameter optimization section.

The patient's cumulative hazard is shown in **Fig. 17**, the x-axis shows the number of months after diagnosis (shown in years as well at the top of the chart), the y-axis shows cumulative probability of cancer specific mortality, and the hazard is shown for the whole 216 months period, the red line shows the moment a patient experienced cancer specific mortality. The hazard always starts as 0 at t_0 and, in this individual's case, reaches 0.21 probability on the follow-up month (20^{th}). If a person would have survived past 20^{th} months, the model predicts a steeper increase in probability after a follow-up timestamp, increasing the probability to 0.73 at 5^{th} year, 0.95 at 10^{th} year and 0.99 at 18^{th} year after diagnosis.

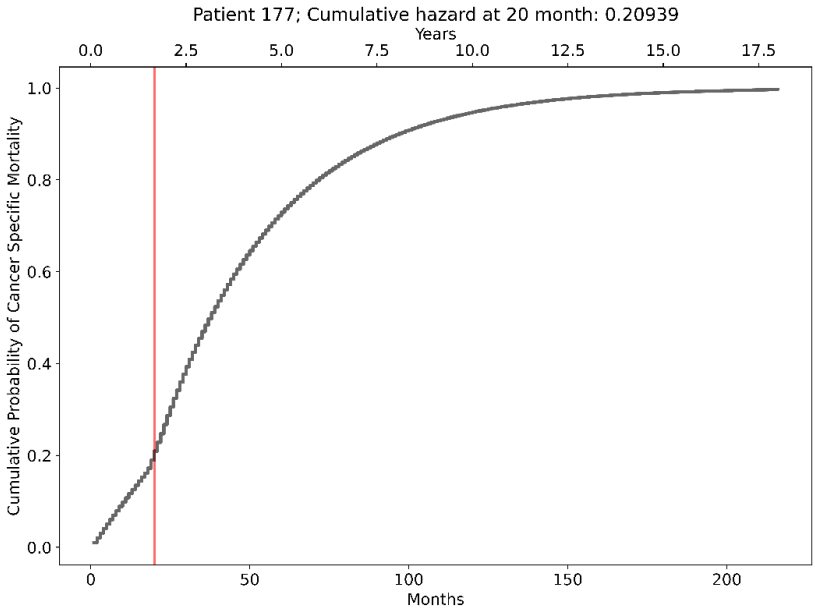

Fig. 17. Patient‘s 177 cumulative hazard from training set

The change in steeper cumulative hazard slope can be seen in discrete mortality probability figure shown in **Fig. 18**. This graph shows the same data, except the y-axis shows the discrete mortality probability instead of cumulative one. At t_1 the mortality probability is 0.01 and increases to 0.024 at t_{20}, after the follow-up, the increased instant mortality probability either stays or increases, reaching 0.033 at t_{216}.

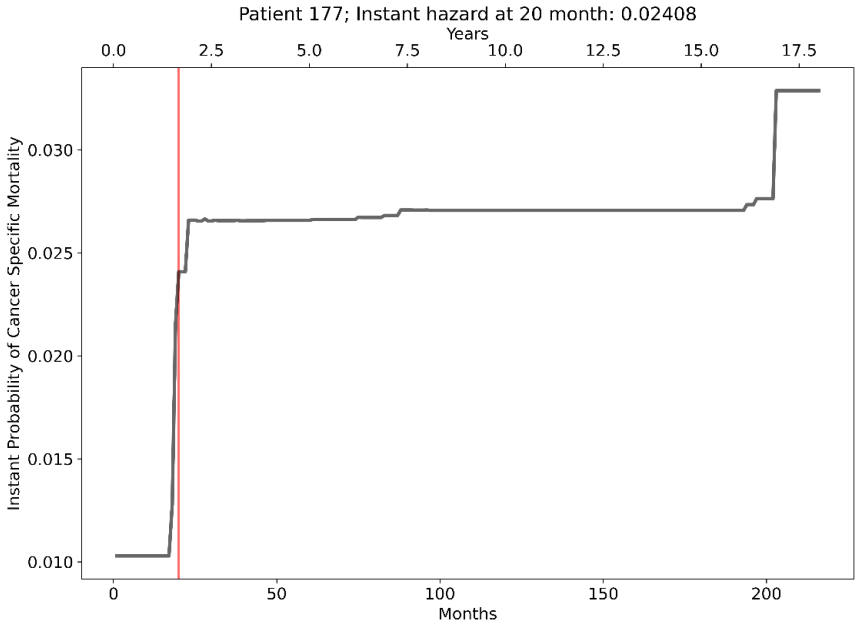

Fig. 18. Patient‘s 177 discrete mortality probability from training set

A different trend is modelled for another person in **Fig. 19**, where at follow-up time t_{96} the risk reached 0.033 (at this time, the individual died due to prostate cancer), increased to 0.065 at 15th year and 0.099 at 18th year. The instant mortality probability can be seen in **Fig. 27**.

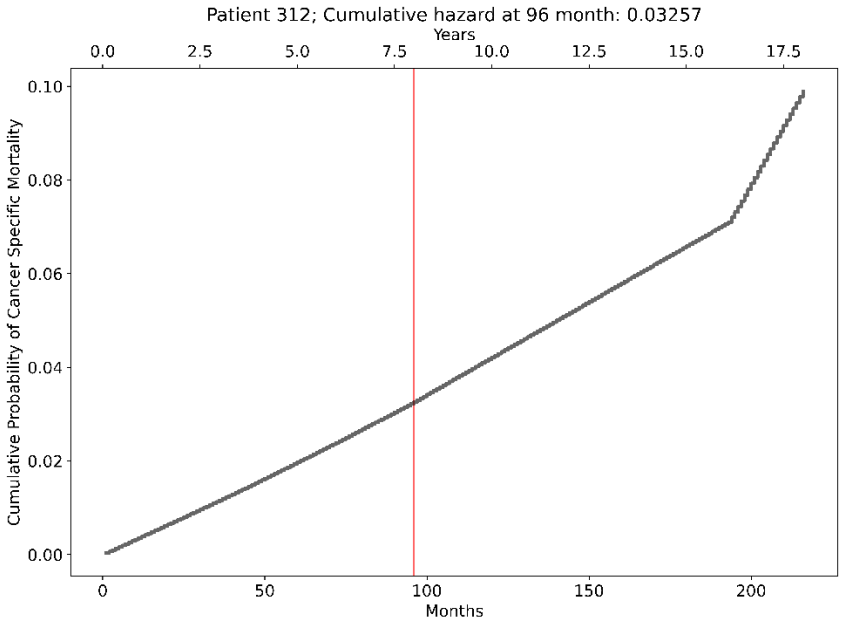

Fig. 19. Patient‘s 312 cumulative hazard from training set

There can also be cases where a patient has not yet died from prostate cancer, but the model predicts a high probability. Such case can be seen in **Fig. 20**, where the cumulative hazard is 0.33 at follow-up time t_{31} and increases in a similar manner as the 177th patient seen in **Fig. 17**. The mortality probability increases to 0.67 in the 5th year, 0.92 in the 15th year and 0.99 in the 18th year. Discrete mortality probabilities can be seen in **Fig. 28**.

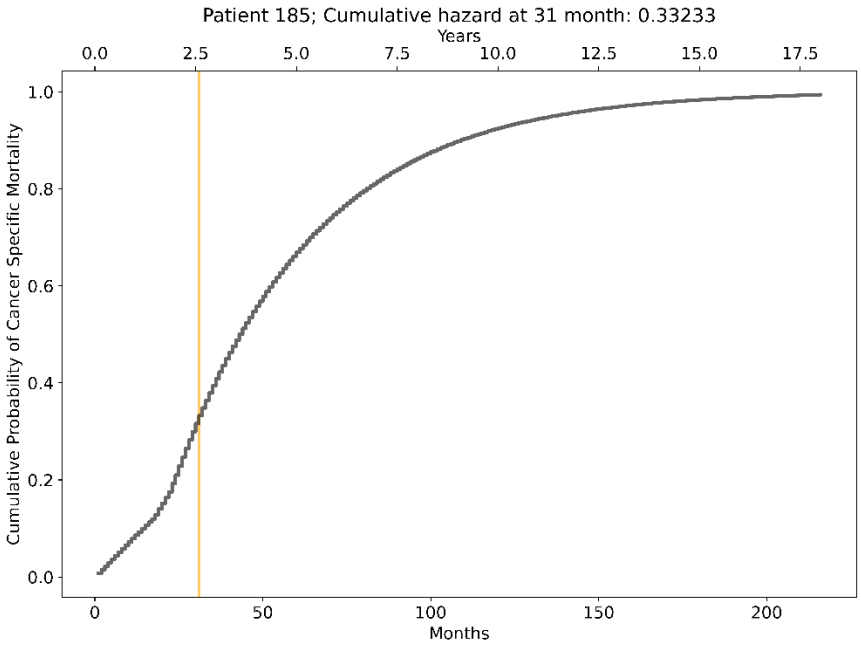

Fig. 20. Patient‘s 185 cumulative hazard from testing set

3.5. Comparison of models

Using optimal hyperparameters for each ML algorithm and response variable, average training and testing set AUC values have been reported in **Table 15** alongside with average standard deviations.

In the case of cancer specific mortality, the optimal model was found to be a random forest having 0.951 (SD = 0.037) training AUC and 0.928 (SD = 0.045) testing AUC as well as having the lowest standard deviation. While logistic regression showed 2nd best results on training set with 0.935 AUC (SD = 0.038), it was the worst out of 4 on the testing set, having 0.890 AUC (SD = 0.058). Predicting death from other causes seemed much harder, where the training AUC ranged from 0.663 to 0.722 and testing AUC from 0.637 to 0.689. While the XGBoost model was found to be optimal on training set, the optimal model is random forest as it performed better on the testing set. All 4 models performed quite the same predicting biochemical recurrence, where the training AUC ranged from 0.859 to 0.897 and testing AUC from 0.840 to 0.855. The optimal model is random forest having training AUC 0.865 (SD = 0.030) and testing AUC 0.855 (SD = 0.034) followed very closely by neural network with training AUC 0.865 (SD = 0.034) and testing AUC 0.850 (SD = 0.040). The XGBoost showed almost perfect AUC for the training set in predicting metastasis, having 0.997 AUC (SD = 0.005), yet it shows overfitting as testing set AUC decreased down to 0.927 (SD = 0.035) while still having the highest testing AUC out of 4 models but having the highest standard deviation as well. The other model which performed well is random forest once again with training set AUC 0.933 (SD = 0.038) and testing AUC 0.921 (SD = 0.028).

Table 15. Optimal models and their AUC metrics

Model	Avg. train AUC	Avg. train AUC SD	Avg. test AUC	Avg. test AUC SD
Cancer specific mortality				
Logistic regression	0.935	0.038	0.890	0.058
Random Forest	0.951	0.037	0.928	0.045
XGBoost	0.919	0.051	0.896	0.053
Neural network	0.903	0.079	0.902	0.092
Death from other causes				
Logistic regression	0.699	0.046	0.637	0.087
Random Forest	0.663	0.049	0.689	0.046
XGBoost	0.722	0.070	0.659	0.066
Neural network	0.670	0.060	0.654	0.078
Biochemical recurrence				
Logistic regression	0.859	0.031	0.840	0.038
Random Forest	0.865	0.030	0.855	0.034
XGBoost	0.897	0.025	0.841	0.039
Neural network	0.865	0.034	0.850	0.040
Metastasis				
Logistic regression	0.926	0.041	0.907	0.027
Random Forest	0.933	0.038	0.921	0.028
XGBoost	0.997	0.005	0.927	0.035
Neural network	0.920	0.042	0.903	0.030

Comparison across models can also be made considering different prediction periods. This can be seen for training set and cancer specific mortality in **Fig. 21**, here, x-axis shows a prediction horizon in months (lower graph side) and years (upper), y-axis shows average AUC. As in **Table 15**, random forest model is optimal, having 0.996 AUC for 2 year prediction horizon, 0.988 for 5 years, 0.909 for

10 years and 0.885 for 18 years. It is also clear from the graph that neural network model struggles more predicting short-term risk than other 3 models, having 0.922 AUC for 2 year prediction.

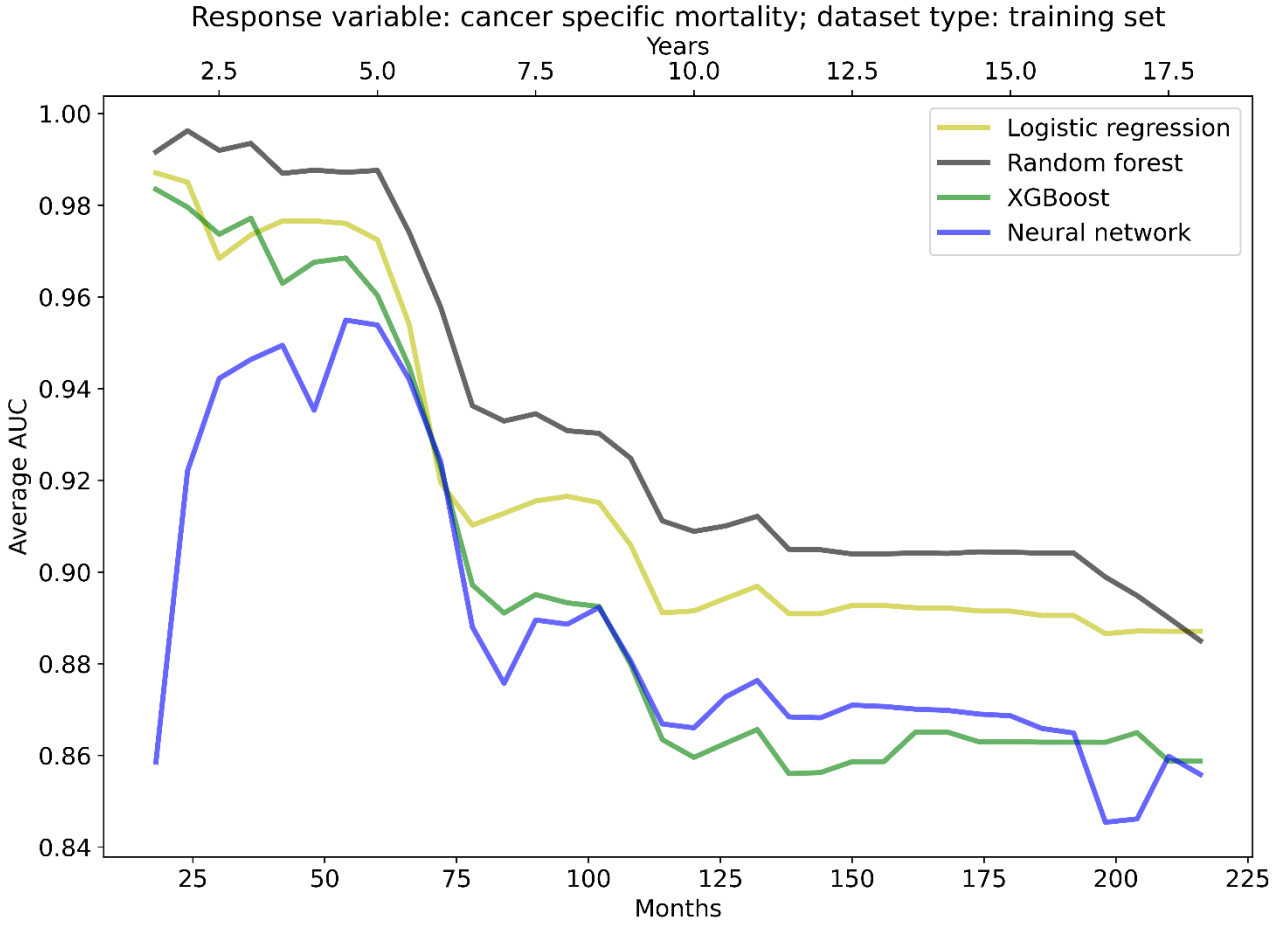

Fig. 21. Average training AUC across different prediction periods for cancer specific mortality

Using testing set, random forest was both optimal when AUCs are average across and as well as looking into different prediction horizons (**Table 15**, **Fig. 22**). Random forest had 0.924 AUC for 2.5 years prediction, 0.971 for 5 years, 0.943 for 10 years and 0.883 for 18 years. It is worth mentioning that prediction horizon only start from 2.5 years as testing set only has first cancer specific mortality at 2.5 years mark. In testing set, the last cancer specific mortality was reported at 11.5 years mark and up to 18 years there is no mortalities reported (**Table 19**).

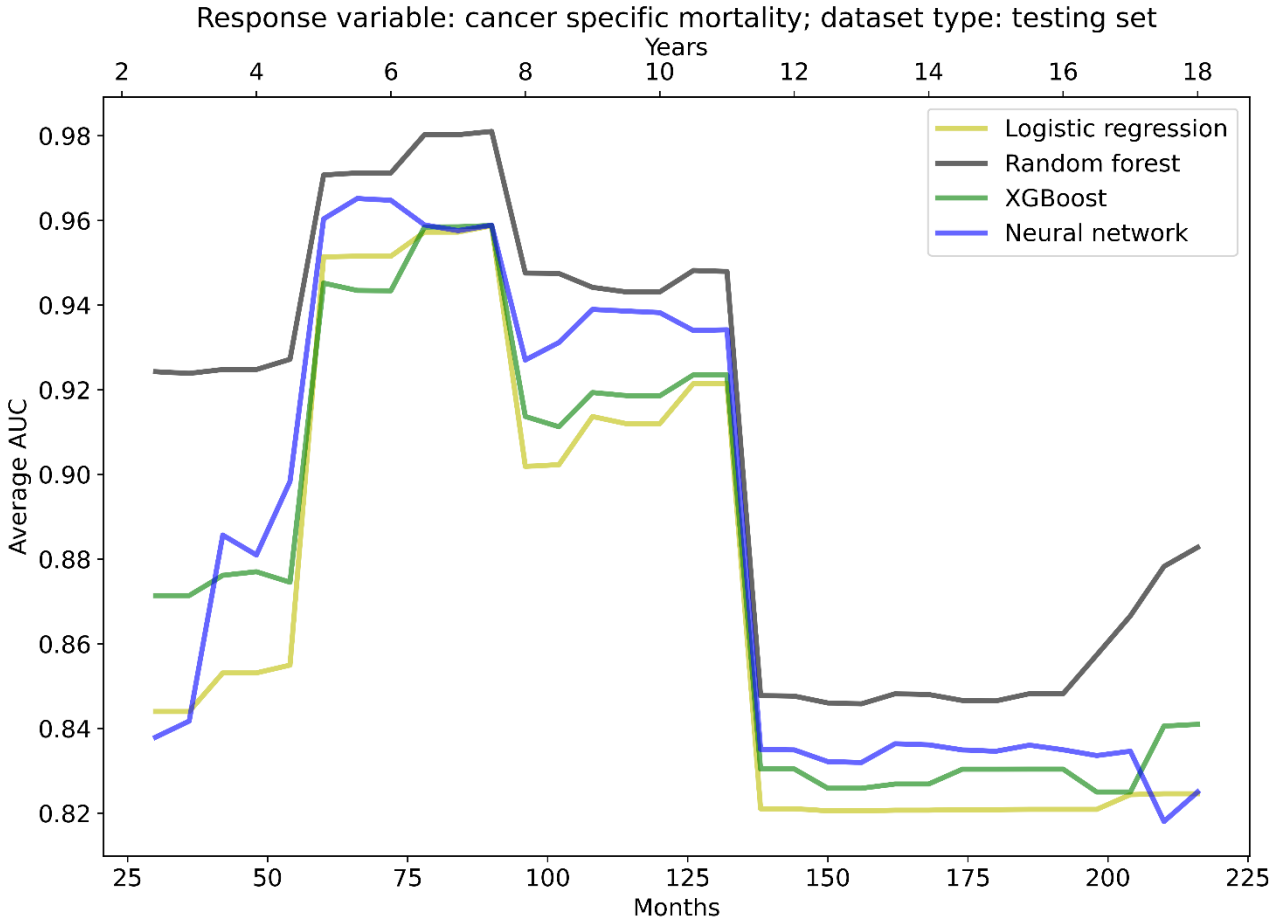

Fig. 22. Average testing AUC across different prediction periods for cancer specific mortality

ML models can also be compared when predicting single patient's mortality. We used optimal models for cancer specific mortality and an example for that is presented in **Fig. 23** with 67 years old patient who has died from prostate cancer at 77th month mark. As we saw above, all 4 models predict cancer specific mortality well, in terms of AUC, but there is a difference in what risk ML models do predict. From the graph it is clear that for this patient, neural network predicts much lower risk than other 3 models. At 5 year mark, random forest predicts 0.20 risk, XGBoost and linear regression – both 0.09 while neural network – only 0.02. At 10 year mark, random forest predicted risk increases to 0.38, XGBoost to 0.24, linear regression – 0.25 and neural network 0.05. While all 3 models predict a different risk, all those risks can be used for prediction depending on different thresholds, the similar AUC values indicates that.

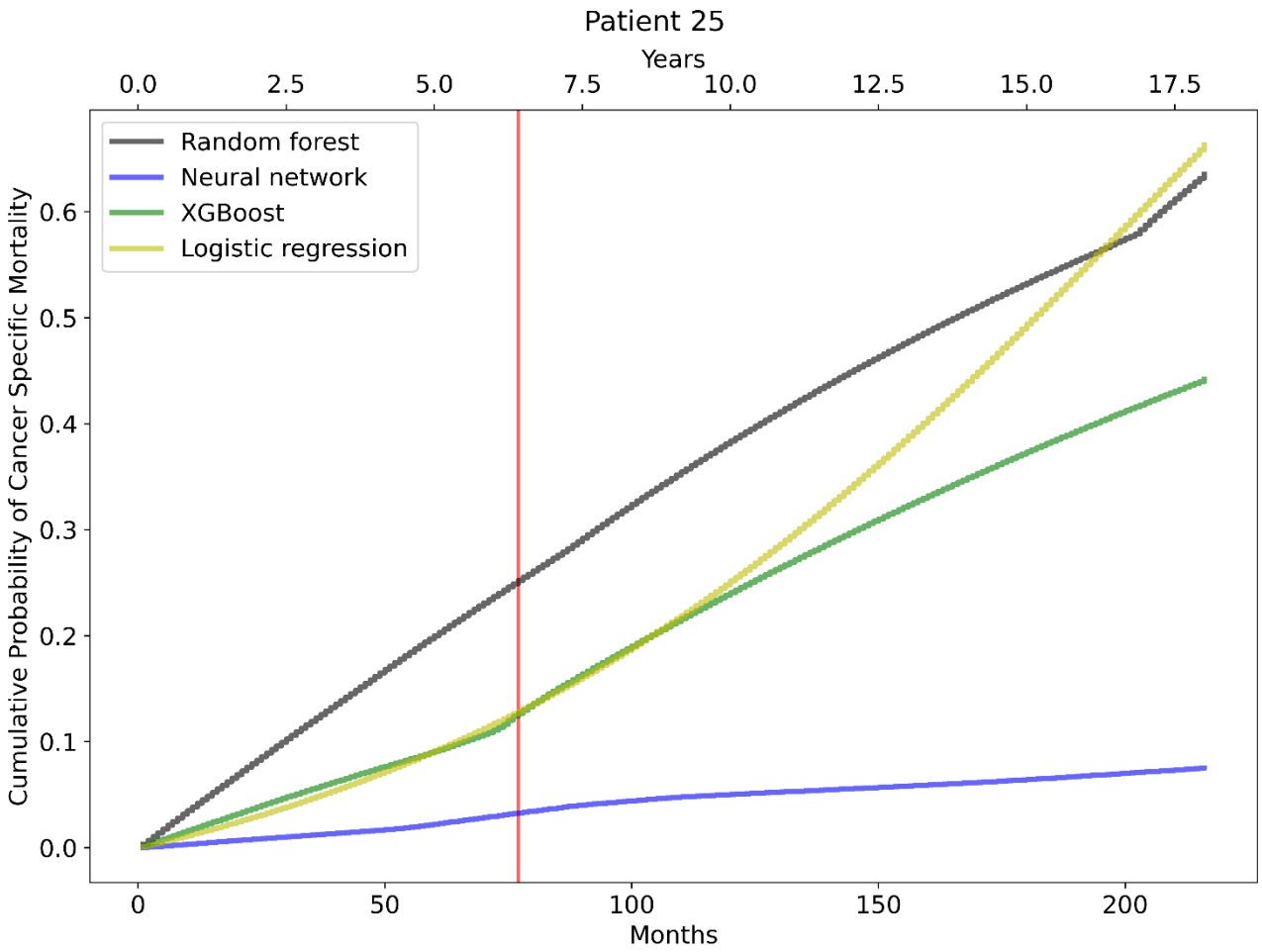

Fig. 23. Patient‘s 25 cumulative hazard from testing set across different models

Looking into discrete mortality probability graph presented in **Fig. 24**, it is visible, that logistic regression stands out from the 4 models. While random forest, neural network and XGBoost methods do estimate a non-monotonic function, which can either increase or decrease, logistic regression estimates monotonically increasing discrete mortality probability function. Logistic regression estimates start with 0.001 discrete probability and monotonically increases up to 0.014. It is also noticeable that XGBoost and neural network estimates slightly increased mortality risk during and around the follow-up month and after that it slightly decreases. Random forest on the other hand does not show any risk increase during the follow-up month.

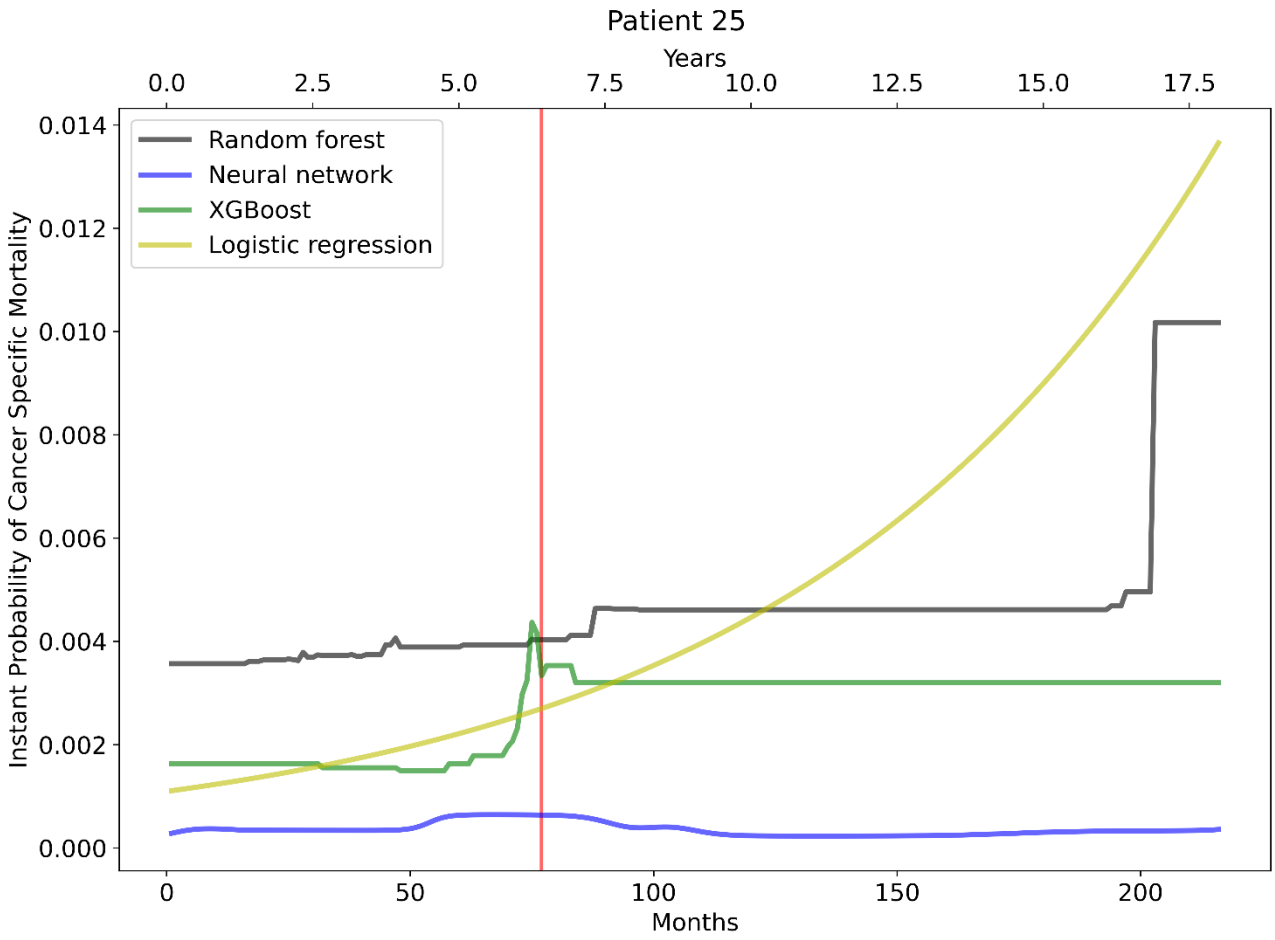

Fig. 24. Patient‘s 25 discrete mortality probability from testing set across different models

Another example of the same patient analysed in **Fig. 20** is presented in **Fig. 25**. In this case, the patient did not yet die from prostate cancer but as we can see, the modelled risks are high. At follow-up mark (31st month) random forest predicts 0.33 risk, linear regression – 0.14 while XGBoost and neural network – 0.04 and 0.02 respectively.

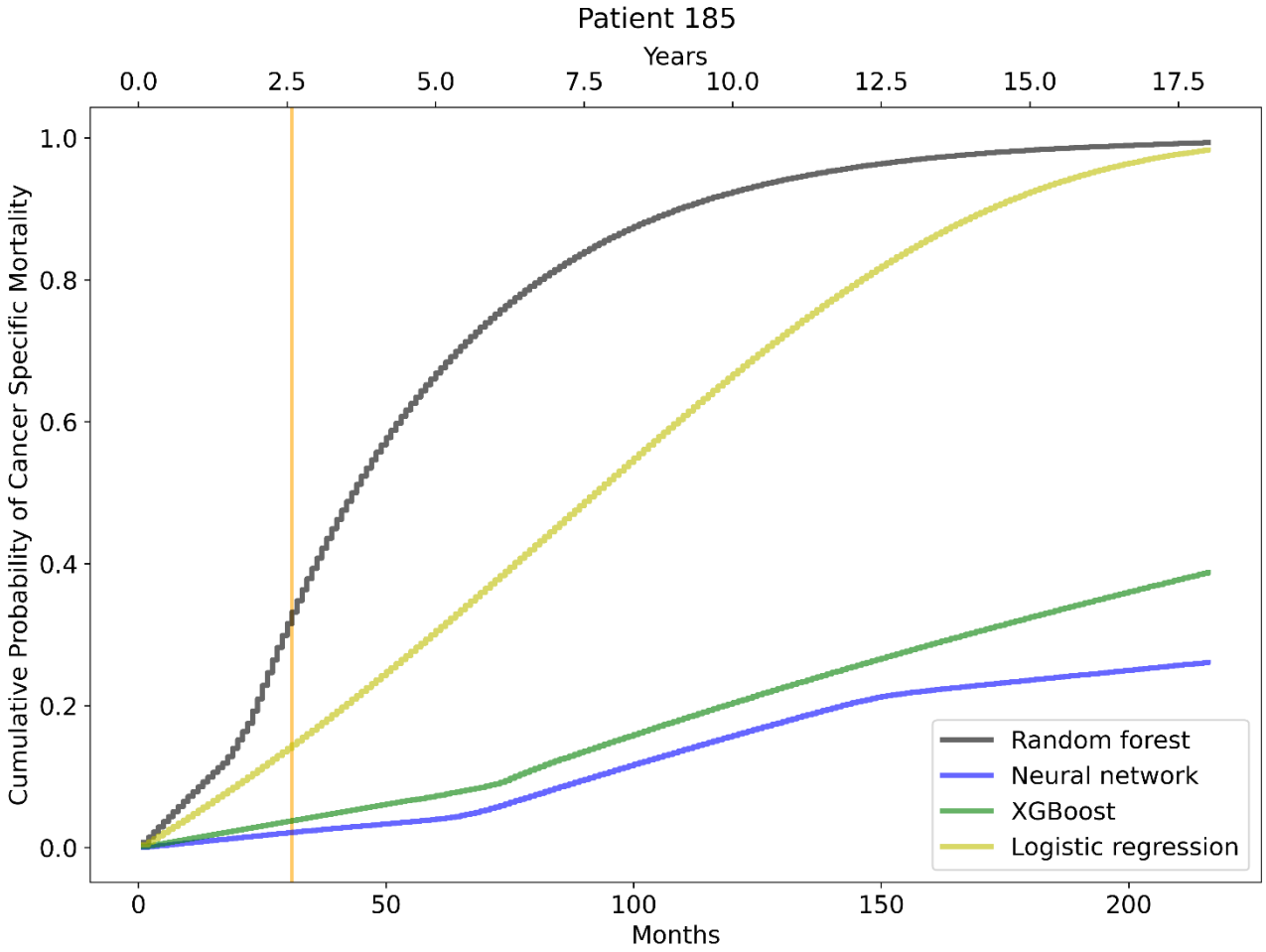

Fig. 25. Patient‘s 185 cumulative hazard from testing set across different models

The discrete mortality probability graph presented in **Fig. 26**, shows a similar trend for logistic regression as in **Fig. 24**. Logistic regression estimates start with 0.004 discrete probability and monotonically increases up to 0.05.

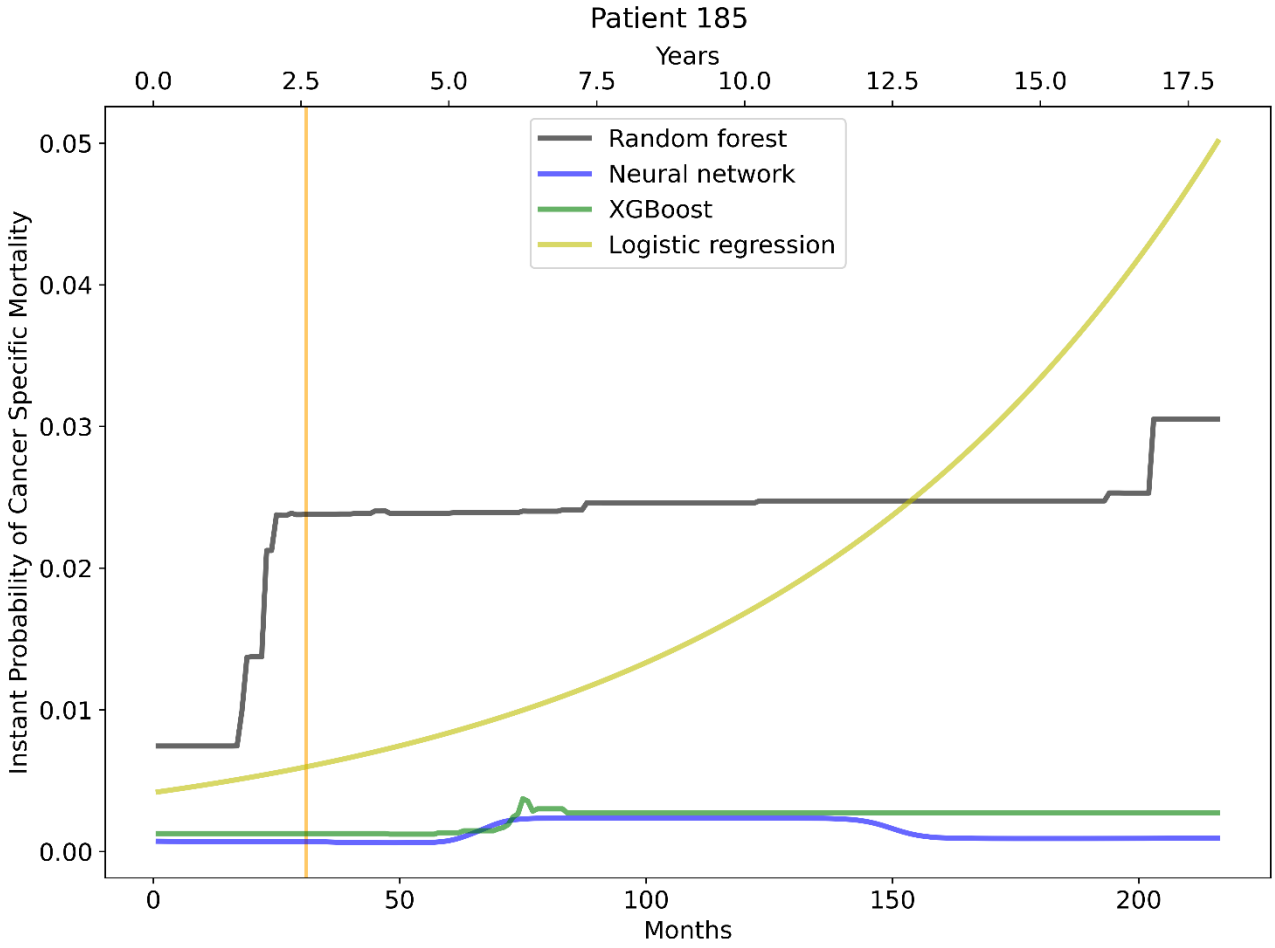

Fig. 26. Patient‘s 185 discrete mortality probability from testing set across different models

3.6. Comparison with classical models

As mentioned in the An overview of conducted research section, a similar analysis has been already done on similar data using classical models – semi-parametric Cox proportional hazards regression and Fine-Gray competing risk regression [12]. Two analysis and the models cannot be directly compared, only the high-level overview can be done. The mentioned paper had a dataset of 2410 and we had 1564 patients, the train/test split was performed differently, and the number of features were also different. In our current research, all features from the previous paper were also used (this time we did not had patient with unknown surgical margin status) with some additional ones, such as PLND01, persistent PSA, clinical stage, biopsy and pathologic Gleason scores, biopsy Gleason grade group and risk groups. The AUC values of these models for cancer specific mortality are shown in **Table 16**.

Table 16. AUC values of classical models

Model	Train AUC	Test AUC
Cox	0.771	0.675
Fine-Gray	0.767	0.658

The classical models achieved AUC testing values of 0.675 and 0.658, respectively with Cox and Fine-Gray models. In our current research, we have been able to achieve 0.928 testing AUC with random forest model.

Conclusions

1. After reviewing prostate cancer incidence and mortality rates across the globe, we have learned that it is a serious issue as based on 2020 report, prostate cancer was the 4th most common type of cancer across the world and 2nd between men. In Lithuania it is the most common type of cancer, 4th by mortality incidences and 2nd between men.
2. After hyperparameter optimization we have been able to find optimal hyperparameters for each machine learning method and for each response variable. The parameters vary depending on the response variable we use.
3. We have successfully implemented discrete time modelling and custom loss function.
4. After calculating cumulative hazards and discrete mortality probabilities on individual level, we have been able to measure the patient's risk not only during the follow-up but also the short-, mid-, and long-term risk after diagnosis. Discrete mortality probabilities gave us insights at what point the discrete risks are increasing.
5. The optimal models have been trained for each response variable. Random forest algorithm returned the highest testing AUC values when modelling cancer specific mortality, death from other causes and biochemical recurrence, respectively the AUC values are: 0.928, 0.689 and 0.855. XGBoost method modelled metastasis with the highest testing AUC – 0.927 but showed overfitting symptoms.
6. We have been able to compare the machine learning models on a single individual level and saw that the predicted risks differ between the models.
7. Doing a high-level comparison of our research to the research we did previously with classical models – we saw that our models predict risks much more accurately, the optimal AUC testing value from classical methods was 0.675 with Cox model while our optimal AUC was 0.928 with random forest.

List of references

1. FERLAY J., LAM F., ERVIK M., COLOMBET M., MERY L., PIÑEROS M., ZNAOR A., SOERJOMATARAM I., and BRAY F., 2020. Global Cancer Observatory: Cancer Today. [interactive]. Access: https://gco.iarc.fr/today
2. SURESH, K., SEVERN, C. and GHOSH, D. Survival prediction models: an introduction to discrete-time modeling. BMC Med Res Methodol 22, 207 (2022). https://doi.org/10.1186/s12874-022-01679-6
3. CHAN, Yiu Ming, "Statistical Analysis and Modeling of Prostate Cancer" (2013). USF Tampa Graduate Theses and Dissertations. https://digitalcommons.usf.edu/etd/4806
4. BOTT SRJ, BIRTLE AJ, TAYLOR CJ, et al. Prostate cancer management: (1) an update on localised disease. Postgraduate Medical Journal 2003; 79:575-580.
5. ROFFMAN, D.A. et al. Development and Validation of a Multiparameterized Artificial Neural Network for Prostate Cancer Risk Prediction and Stratification. In JCO Clinical Cancer Informatics. 2018. no. 2, p. 1–10.
6. BOSTWICK, D.G., BURKE, H.B., DJAKIEW, D., EULING, S., Ho, S.-m., LANDOLPH, J., MORRISON, H., SONAWANE, B., SHIFFLETT, T., WATERS, D.J. and TIMMS, B. (2004), Human prostate cancer risk factors. Cancer, 101: 2371-2490. https://doi.org/10.1002/cncr.20408
7. MILONAS D, VENCLOVAS Z, SASNAUSKAS G, RUZGAS T. The Significance of Prostate Specific Antigen Persistence in Prostate Cancer Risk Groups on Long-Term Oncological Outcomes. Cancers. 2021; 13(10):2453. https://doi.org/10.3390/cancers13102453
8. RODRIGUES, G., WARDE, P., PICKLES, T., CROOK, J., BRUNDAGE, M., SOUHAMI, L., LUKKA, H., and Genitourinary Radiation Oncologists of Canada (2012). Pre-treatment risk stratification of prostate cancer patients: A critical review. Canadian Urological Association journal “Journal de l'Association des urologues du Canada”, 6(2), 121–127. https://doi.org/10.5489/cuaj.11085
9. ZUPAN B., DEMŠAR J., KATTAN M., BECK, J. (2000). Machine Learning for Survival Analysis: A Case Study on Recurrence of Prostate Cancer. Artificial intelligence in medicine. 20. 59-75. 10.1016/S0933-3657(00)00053-1.
10. CAPITANIO, U., BRIGANTI, A., GALLINA, A., SUARDI, N., KARAKIEWICZ, P.I., MONTORSI, F. and SCATTONI, V. (2010), Predictive models before and after radical prostatectomy. Prostate, 70: 1371-1378. https://doi.org/10.1002/pros.21159
11. COX, D.R. (1972), Regression Models and Life-Tables. Journal of the Royal Statistical Society: Series B (Methodological), 34: 187-202. https://doi.org/10.1111/j.2517-6161.1972.tb00899.x
12. KRAUJALIS V., RUZGAS T., MILONAS D. (2022). Mortality Rate Estimation Models for Patients with Prostate Cancer Diagnosis. Baltic Journal of Modern Computing. 10. 10.22364/bjmc.2022.10.2.06.
13. STEPHENSON, A.J. et al. Postoperative Nomogram Predicting the 10-Year Probability of Prostate Cancer Recurrence After Radical Prostatectomy. In Journal of Clinical Oncology . 2005. Vol. 23, no. 28, p. 7005–7012.
14. SUARDI, N., PORTER, C.R., REUTHER, A.M., WALZ, J., KODAMA, K., GIBBONS, R.P., CORREA, R., MONTORSI, F., GRAEFEN, M., HULAND, H., KLEIN, E.A. and KARAKIEWICZ, P.I. (2008), A nomogram predicting long-term biochemical recurrence after radical prostatectomy. Cancer, 112: 1254-1263. https://doi.org/10.1002/cncr.23293

15. Susan F. SLOVIN, Andrew S. WILTON, Glenn HELLER, Howard I. SCHER; Time to Detectable Metastatic Disease in Patients with Rising Prostate-Specific Antigen Values following Surgery or Radiation Therapy. Clin Cancer Res 15 December 2005; 11 (24): 8669–8673. https://doi.org/10.1158/1078-0432.CCR-05-1668
16. D'AMICO, A.V. et al. Preoperative PSA Velocity and the Risk of Death from Prostate Cancer after Radical Prostatectomy. In New England Journal of Medicine . 2004. Vol. 351, no. 2, p. 125–135.
17. FREEDLAND SJ, HUMPHREYS EB, MANGOLD LA, et al. Risk of Prostate Cancer–Specific Mortality Following Biochemical Recurrence After Radical Prostatectomy. JAMA. 2005;294(4):433–439. doi:10.1001/jama.294.4.433
18. LUNN, M. and MCNEIL, D. (1995). Applying Cox Regression to Competing Risks. Biometrics, 51(2), 524–532. https://doi.org/10.2307/2532940
19. D'AMICO A, MOUL J., CARROLL P., SUN L., LUBECK D., CHEN M. (2023). Cancer-Specific Mortality After Surgery or Radiation for Patients With Clinically Localized Prostate Cancer Managed During the Prostate-Specific Antigen Era. Journal of clinical oncology: official journal of the American Society of Clinical Oncology, 21(11), 2163–2172. https://doi.org/10.1200/JCO.2003.01.075
20. Jason P. FINE & Robert J. GRAY (1999) A Proportional Hazards Model for the Subdistribution of a Competing Risk, Journal of the American Statistical Association, 94:446, 496-509, DOI: 10.1080/01621459.1999.10474144
21. BEAM AL, KOHANE IS. Big Data and Machine Learning in Health Care. JAMA. 2018;319(13):1317–1318. doi:10.1001/jama.2017.18391
22. TĂTARU OS, VARTOLOMEI MD, RASSWEILER JJ, VIRGIL O, LUCARELLI G, PORPIGLIA F, AMPARORE D, MANFREDI M, CARRIERI G, FALAGARIO U, TERRACCIANO D, DE COBELLI O, BUSETTO GM, GIUDICE FD, FERRO M. Artificial Intelligence and Machine Learning in Prostate Cancer Patient Management - Current Trends and Future Perspectives. Diagnostics. 2021; 11(2):354. https://doi.org/10.3390/diagnostics11020354
23. JENTZER, J.C. et al. Clinical applications of artificial intelligence and machine learning in the modern cardiac intensive care unit. In Intelligence-Based Medicine . 2023. Vol. 7, p. 100089.
24. RAJKOMAR, A. et al. Machine Learning in Medicine. In New England Journal of Medicine . 2019. Vol. 380, no. 14, p. 1347–1358.
25. BIBAULT J-E, HANCOCK S, BUYYOUNOUSKI MK, BAGSHAW H, LEPPERT JT, LIAO JC, XING L. Development and Validation of an Interpretable Artificial Intelligence Model to Predict 10-Year Prostate Cancer Mortality. Cancers. 2021; 13(12):3064. https://doi.org/10.3390/cancers13123064
26. BARLOW H, MAO S, KHUSHI M. Predicting High-Risk Prostate Cancer Using Machine Learning Methods. Data. 2019; 4(3):129. https://doi.org/10.3390/data4030129
27. NITTA, S. et al. Machine learning methods can more efficiently predict prostate cancer compared with prostate-specific antigen density and prostate-specific antigen velocity. In Prostate International . 2019. Vol. 7, no. 3, p. 114–118.
28. TAN, Y. G., FANG, A. H. S., LIM, J. K. S., KHALID, F., CHEN, K., Ho, H. S. S., YUEN, J. S. P., HUANG, H. H., TAY, K. J. (2022). Incorporating artificial intelligence in urology: Supervised machine learning algorithms demonstrate comparative advantage over nomograms in predicting

biochemical recurrence after prostatectomy. The Prostate, 82(3), 298–305. https://doi.org/10.1002/pros.24272

29. ERDEM, E., BOZKURT, F. (2021). A comparison of various supervised machine learning techniques for prostate cancer prediction . Avrupa Bilim ve Teknoloji Dergisi , (21) , 610-620 . DOI: 10.31590/ejosat.802810
30. Ping WANG, Yan LI, and Chandan K. REDDY. 2019. Machine Learning for Survival Analysis: A Survey. ACM Comput. Surv. 51, 6, Article 110 (November 2019), 36 pages. https://doi.org/10.1145/3214306
31. Tianqi CHEN and Carlos GUESTRIN. 2016. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). Association for Computing Machinery, New York, NY, USA, 785–794. https://doi.org/10.1145/2939672.2939785
32. FAWCETT, T. An introduction to ROC analysis. In ROC Analysis in Pattern Recognition . 2006. Vol. 27, no. 8, p. 861–874.
33. Jasper SNOEK, Hugo LAROCHELLE, and Ryan P. ADAMS. 2012. Practical Bayesian optimization of machine learning algorithms. In Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'12). Curran Associates Inc., Red Hook, NY, USA, 2951–2959.

Appendices

Appendix 1. Descriptive tables

Table 17. Descriptive characteristics of 1564 prostate cancer patients.

	(N=1564)
Patient‘s age	
Median	64.0
Min - Max	40.0 - 87.0
Prostate specific antigen (PSA), ng/Ml	
Median	6.4
Quantile 25% – 75%	4.8 - 9.7
PLND01 [7], n (%)	
0, lymph node status 0 and 1	962 (61.5%)
1, lymph node status untreated	602 (38.5%)
Persistent PSA [7], n (%)	
0, undetectable PSA < 0.1 ng/mL	1285 (82.2%)
1, persistent PSA ≥ 0.1 ng/mL at the first measurement 4-8 weeks after RP	279 (17.8%)
Clinical stage (cT), n (%)	
1	434 (27.7%)
2	911 (58.2%)
3	219 (14.0%)
Pathologic stage (pT), n (%)	
0, initial stage, cancer is not detectable by tomography	999 (63.9%)
1, cancer is developing in the prostate area	427 (27.3%)
2, cancer has spread outside the prostate area	138 (8.8%)
Lymph node status (LN), n (%)	
0, pN0, clean	535 (34.2%)
1, pN1, damaged by cancer	67 (4.3%)
2, pNx, untreated	962 (61.5%)
Surgical margin status (SM), n (%)	
0, clean	1052 (67.3%)
1, damaged by cancer	512 (32.7%)
Biopsy gleason score (GS)	
Mean (standard deviation)	6.5 (0.7)
Biopsy gleason grade group, n (%)	
1, biopsy GS (<=6)	894 (57.2%)
2, biopsy GS (3+4)	479 (30.6%)
3, biopsy GS (4+3)	75 (4.8%)
4, biopsy GS (=8)	79 (5.1%)
5, biopsy GS (>=9)	37 (2.4%)
Pathologic gleason score (GS)	

Mean (standard deviation)	6.9 (0.8)
Pathologic gleason grade group, n (%)	
1, pathologic GS (<=6)	429 (27.4%)
2, pathologic GS (3+4)	788 (50.4%)
3, pathologic GS (4+3)	165 (10.5%)
4, pathologic GS (=8)	84 (5.4%)
5, pathologic GS (>=9)	98 (6.3%)
Risk group [7], n (%)	
0, low, pT2, GG1, PSA<10 ng/mL, Nx, pN0	337 (21.5%)
1, intermediate, pT3a, GG2-3, PSA 10-20 ng/mL, Nx, pN0	926 (59.2%)
2, high, pT3b, GG4-5, PSA>20 ng/mL, pN1	301 (19.2%)
Overall mortality (OM), n (%)	
0, no	1300 (83.1%)
1, yes	264 (16.9%)
Cancer specific mortality (CSM), n (%)	
0, no	1514 (96.8%)
1, yes	50 (3.2%)
Death from other causes (DOC), n (%)	
0, no	1350 (86.3%)
1, yes	214 (13.7%)
Survival time (months)	
Median	104.0
Quantile 25% – 75%	65.0 - 159.0
Biochemical recurrence (BCR), n (%) defined as PSA > 0.2 ng/mL at two consecutive measurements [7].	
0, no	1104 (70.6%)
1, yes	460 (29.4%)
Survival time (months) (BCR)	
Median	41.5
Quantile 25% – 75%	14.0 - 85.0
Metastasis (MTS), n (%)	
0, no	1465 (93.7%)
1, yes	99 (6.3%)
Survival time (months) (MTS) defined as skeletal or visceral lesions confirmed by a bone scar, computed tomography or magnetic resonance imaging [7].	
Median	62.0
Quantile 25% – 75%	26.8 - 119.2

Table 18. Descriptive characteristics of 1564 prostate cancer patients across train/test split.

	Train (N=1251)	Test (N=313)
Patient‘s age		
Median	64.0	63.0
Min - Max	40.0 - 78.0	45.0 - 87.0
Prostate specific antigen (PSA), ng/Ml		
Median	6.5	6.2
Quantile 25% – 75%	4.8 - 9.7	4.8 - 9.3
PLND01 [7], n (%)		
0, lymph node status 0 and 1	766 (61.2%)	196 (62.6%)
1, lymph node status untreated	485 (38.8%)	117 (37.4%)
Persistent PSA [7], n (%)		
0, undetectable PSA < 0.1 ng/mL	1041 (83.2%)	244 (78.0%)
1, persistent PSA ≥ 0.1 ng/mL at the first measurement 4-8 weeks after RP	210 (16.8%)	69 (22.0%)
Clinical stage (cT), n (%)		
1	347 (27.7%)	87 (27.8%)
2	729 (58.3%)	182 (58.1%)
3	175 (14.0%)	44 (14.1%)
Pathologic stage (pT), n (%)		
0, initial stage, cancer is not detectable by tomography	796 (63.6%)	203 (64.9%)
1, cancer is developing in the prostate area	348 (27.8%)	79 (25.2%)
2, cancer has spread outside the prostate area	107 (8.6%)	31 (9.9%)
Lymph node status (LN), n (%)		
0, pN0, clean	430 (34.4%)	105 (33.5%)
1, pN1, damaged by cancer	55 (4.4%)	12 (3.8%)
2, pNx, untreated	766 (61.2%)	196 (62.6%)
Surgical margin status (SM), n (%)		
0, clean	847 (67.7%)	205 (65.5%)
1, damaged by cancer	404 (32.3%)	108 (34.5%)
Biopsy gleason score (GS)		
Mean (standard deviation)	6.5 (0.7)	6.5 (0.8)
Biopsy gleason grade group, n (%)		
1, biopsy GS (<=6)	710 (56.8%)	184 (58.8%)
2, biopsy GS (3+4)	390 (31.2%)	89 (28.4%)
3, biopsy GS (4+3)	57 (4.6%)	18 (5.8%)
4, biopsy GS (=8)	67 (5.4%)	12 (3.8%)
5, biopsy GS (>=9)	27 (2.2%)	10 (3.2%)
Pathologic gleason score (GS)		
Mean (standard deviation)	6.9 (0.8)	6.9 (0.8)
Pathologic gleason grade group, n (%)		
1, pathologic GS (<=6)	332 (26.5%)	97 (31.0%)

2, pathologic GS (3+4)	638 (51.0%)	150 (47.9%)
3, pathologic GS (4+3)	136 (10.9%)	29 (9.3%)
4, pathologic GS (=8)	67 (5.4%)	17 (5.4%)
5, pathologic GS (>=9)	78 (6.2%)	20 (6.4%)
Risk group [7], n (%)		
0, low, pT2, GG1, PSA<10 ng/mL, Nx, pN0	261 (20.9%)	76 (24.3%)
1, intermediate, pT3a, GG2-3, PSA 10-20 ng/mL, Nx, pN0	749 (59.9%)	177 (56.5%)
2, high, pT3b, GG4-5, PSA>20 ng/mL, pN1	241 (19.3%)	60 (19.2%)
Overall mortality (OM), n (%)		
0, no	1040 (83.1%)	260 (83.1%)
1, yes	211 (16.9%)	53 (16.9%)
Cancer specific mortality (CSM), n (%)		
0, no	1209 (96.6%)	305 (97.4%)
1, yes	42 (3.4%)	8 (2.6%)
Death from other causes (DOC), n (%)		
0, no	1082 (86.5%)	268 (85.6%)
1, yes	169 (13.5%)	45 (14.4%)
Survival time (months)		
Median	104.0	106.0
Quantile 25% – 75%	64.0 - 159.0	70.0 - 163.0
Biochemical recurrence (BCR), n (%) defined as PSA > 0.2 ng/mL at two consecutive measurements [7].		
0, no	894 (71.5%)	210 (67.1%)
1, yes	357 (28.5%)	103 (32.9%)
Survival time (months) (BCR)		
Median	40.0	45.0
Quantile 25% – 75%	15.0 - 84.0	12.0 - 91.0
Metastasis (MTS), n (%)		
0, no	1171 (93.6%)	294 (93.9%)
1, yes	80 (6.4%)	19 (6.1%)
Survival time (months) (MTS) defined as skeletal or visceral lesions confirmed by a bone scar, computed tomography or magnetic resonance imaging [7].		
Median	61.0	72.0
Quantile 25% – 75%	26.0 - 117.0	31.0 - 122.0

Table 19. Cumulative sum of events across different survival times in training/testing sets

	Biochemical recurrence	Cancer specific mortality	Death from other causes	Metastasis

Month	Train	Test	Train	Test	Train	Test	Train	Test
6	127	40			2	2	4	
12	170	53			4	3	10	
18	199	58	1		8	5	18	2
24	223	63	3		10	7	23	3
30	240	68	6	1	13	8	27	4
36	257	72	7	1	20	11	29	7
42	268	75	11	1	24	12	29	8
48	279	82	13	1	27	12	32	9
54	284	83	13	1	34	12	34	10
60	296	88	14	3	38	15	36	13
66	304	89	16	3	46	18	42	15
72	312	91	17	3	58	20	44	15
78	319	92	22	4	65	21	46	15
84	327	92	26	4	74	22	49	15
90	330	93	27	4	78	25	52	15
96	336	94	29	5	88	27	55	15
102	339	95	30	5	95	31	56	15
108	344	95	33	6	99	33	56	15
114	346	96	34	6	108	35	60	16
120	349	97	35	6	116	37	62	16
126	351	97	36	7	120	40	64	17
132	352	99	37	7	124	42	65	17
138	352	99	38	8	132	43	68	17
144	354	99	38	8	136	43	69	17
150	354	101	39	8	140	44	70	17
156	355	101	39	8	148	44	72	17
162	356	102	39	8	152	44	72	18
168	357	103	39	8	154	44	75	18
174	357	103	39	8	159	45	75	18
180	357	103	39	8	162	45	75	19
186	357	103	39	8	167	45	76	19
192	357	103	39	8	169	45	78	19
198	357	103	41	8	169	45	79	19
204	357	103	42	8	169	45	79	19
210	357	103	42	8	169	45	80	19
216	357	103	42	8	169	45	80	19
Total	**357**	**103**	**42**	**8**	**169**	**45**	**80**	**19**

Appendix 2. Parameter hyperparameter optimization table

Table 20. Hyperparameter optimization experiments

	Avg. train AUC	Avg. test AUC	Rank per model	Rank per response variable
Biochemical recurrence				
Logistic regression				
Experiment 001	0.859 (0.031)	0.840 (0.038)	1	10
Neural network				
Experiment 001	0.865 (0.034)	0.850 (0.040)	1	4
Experiment 002	0.867 (0.034)	0.846 (0.039)	2	6
Experiment 004	0.859 (0.037)	0.846 (0.041)	3	7
Experiment 003	0.859 (0.037)	0.846 (0.041)	4	8
Experiment 005	0.859 (0.039)	0.839 (0.044)	5	11
Random forest				
Experiment 005	0.865 (0.030)	0.855 (0.034)	1	1
Experiment 006	0.867 (0.031)	0.852 (0.034)	2	2
Experiment 007	0.867 (0.032)	0.850 (0.037)	3	3
Experiment 008	0.863 (0.033)	0.848 (0.038)	4	5
Experiment 002	0.823 (0.028)	0.801 (0.034)	5	13
Experiment 003	0.823 (0.028)	0.801 (0.034)	6	14
Experiment 004	0.812 (0.029)	0.778 (0.035)	7	15
Experiment 001	0.780 (0.010)	0.764 (0.021)	8	16
XGBoost				
Experiment 002	0.897 (0.025)	0.841 (0.039)	1	9
Experiment 001	0.990 (0.010)	0.807 (0.048)	2	12
Cancer specific mortality				
Logistic regression				
Experiment 001	0.935 (0.038)	0.890 (0.058)	1	10
Neural network				
Experiment 004	0.903 (0.079)	0.902 (0.092)	1	5
Experiment 005	0.915 (0.069)	0.901 (0.087)	2	7
Experiment 007	0.934 (0.045)	0.887 (0.055)	3	12
Experiment 003	0.920 (0.085)	0.883 (0.077)	4	13
Experiment 006	0.787 (0.106)	0.735 (0.219)	5	16
Experiment 001	0.457 (0.222)	0.462 (0.253)	6	17
Experiment 002	0.416 (0.238)	0.442 (0.249)	7	18
Random forest				
Experiment 005	0.951 (0.037)	0.928 (0.045)	1	1
Experiment 001	0.939 (0.035)	0.923 (0.050)	2	2
Experiment 006	0.939 (0.037)	0.919 (0.067)	3	3
Experiment 007	0.965 (0.028)	0.910 (0.050)	4	4
Experiment 003	0.944 (0.035)	0.901 (0.052)	5	6

Experiment 008	0.961 (0.030)	0.898 (0.049)	6	8
Experiment 004	0.942 (0.037)	0.887 (0.055)	7	11
Experiment 002	0.946 (0.029)	0.873 (0.070)	8	14
XGBoost				
Experiment 002	0.919 (0.051)	0.896 (0.053)	1	9
Experiment 001	0.999 (0.001)	0.853 (0.069)	2	15
Death from other causes				
Logistic regression				
Experiment 001	0.699 (0.046)	0.637 (0.087)	1	10
Neural network				
Experiment 003	0.670 (0.060)	0.654 (0.078)	1	8
Experiment 002	0.628 (0.071)	0.628 (0.094)	2	11
Experiment 001	0.712 (0.102)	0.614 (0.080)	3	12
Random forest				
Experiment 003	0.663 (0.049)	0.689 (0.046)	1	1
Experiment 002	0.663 (0.049)	0.689 (0.046)	2	2
Experiment 004	0.663 (0.048)	0.689 (0.045)	3	3
Experiment 005	0.802 (0.095)	0.686 (0.041)	4	4
Experiment 006	0.805 (0.084)	0.678 (0.045)	5	5
Experiment 008	0.737 (0.076)	0.661 (0.043)	6	6
Experiment 007	0.791 (0.078)	0.641 (0.077)	7	9
Experiment 001	0.500 (0.000)	0.500 (0.000)	8	14
XGBoost				
Experiment 002	0.722 (0.070)	0.659 (0.066)	1	7
Experiment 001	1.000 (0.001)	0.584 (0.092)	2	13
Metastasis				
Logistic regression				
Experiment 001	0.926 (0.041)	0.907 (0.027)	1	7
Neural network				
Experiment 002	0.920 (0.042)	0.903 (0.030)	1	8
Experiment 001	0.921 (0.051)	0.902 (0.031)	2	11
Experiment 003	0.504 (0.113)	0.494 (0.136)	3	14
Random forest				
Experiment 005	0.933 (0.038)	0.921 (0.028)	1	3
Experiment 007	0.943 (0.037)	0.919 (0.028)	2	4
Experiment 006	0.927 (0.038)	0.912 (0.035)	3	5
Experiment 008	0.939 (0.042)	0.912 (0.030)	4	6
Experiment 002	0.909 (0.043)	0.902 (0.022)	5	9
Experiment 003	0.909 (0.043)	0.902 (0.022)	6	10
Experiment 004	0.910 (0.044)	0.900 (0.020)	7	12
Experiment 001	0.904 (0.044)	0.889 (0.022)	8	13

XGBoost				
Experiment 001	0.997 (0.005)	0.927 (0.035)	1	1
Experiment 002	0.926 (0.033)	0.921 (0.024)	2	2

Appendix 3. Patient mortality figures

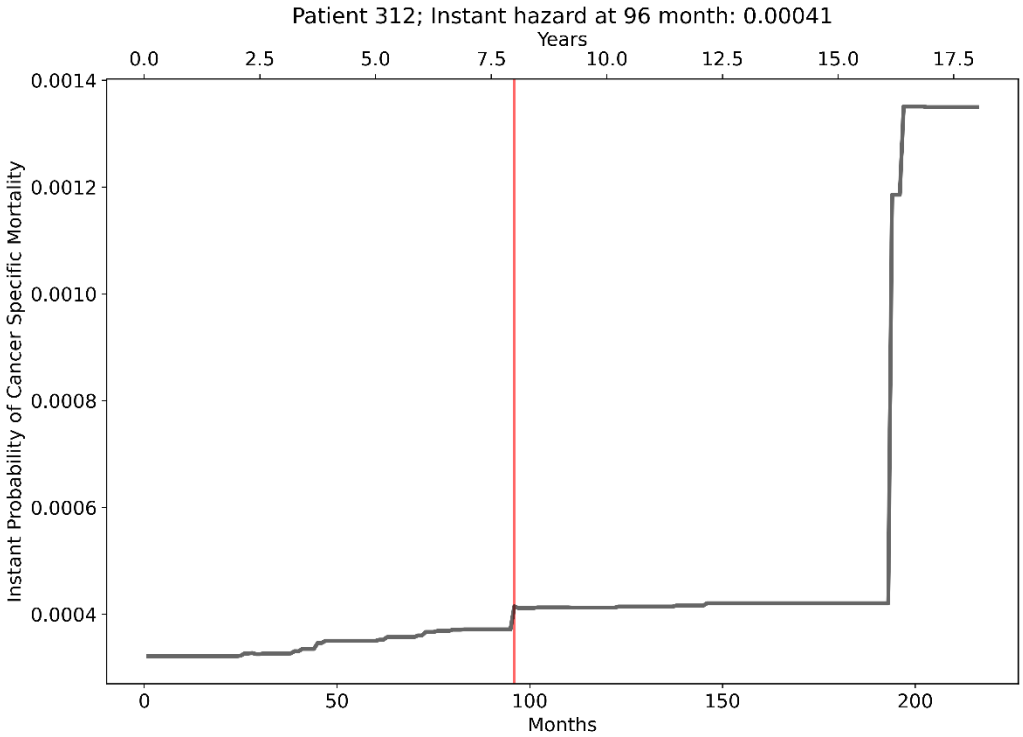

Fig. 27. Patient‘s 312 discrete mortality probability from training set

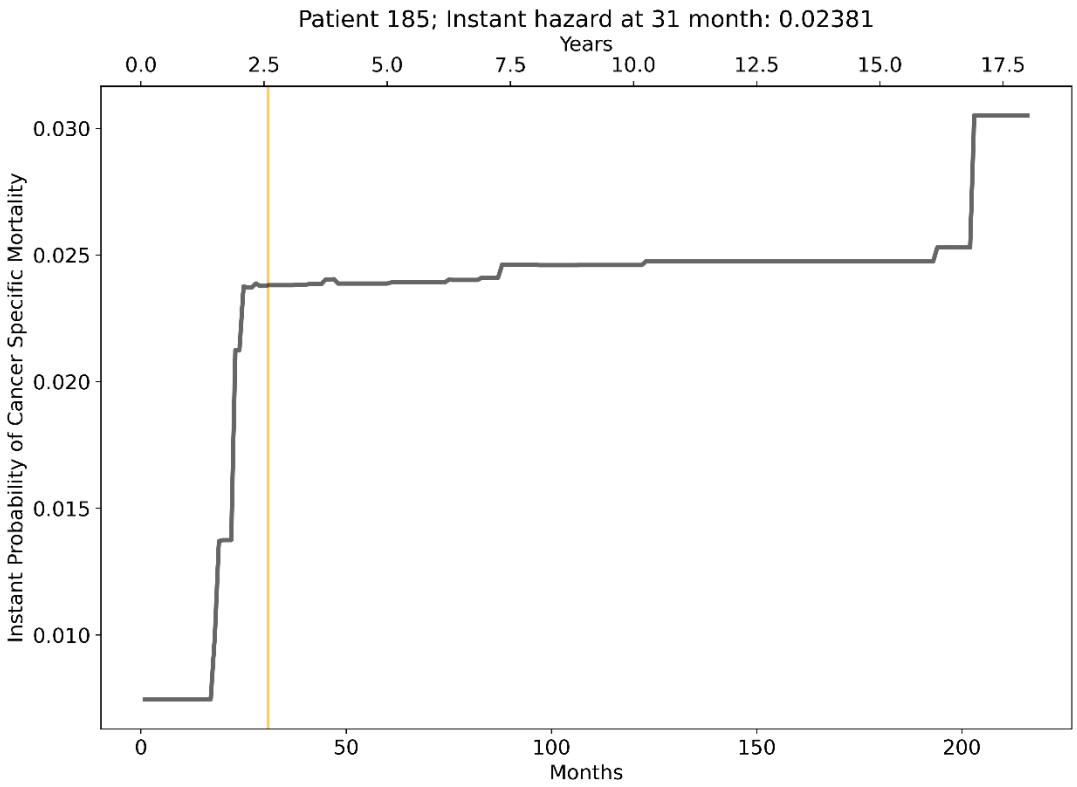

Fig. 28. Patient‘s 185 discrete mortality probability from testing set

Appendix 4. Graphs for model comparison

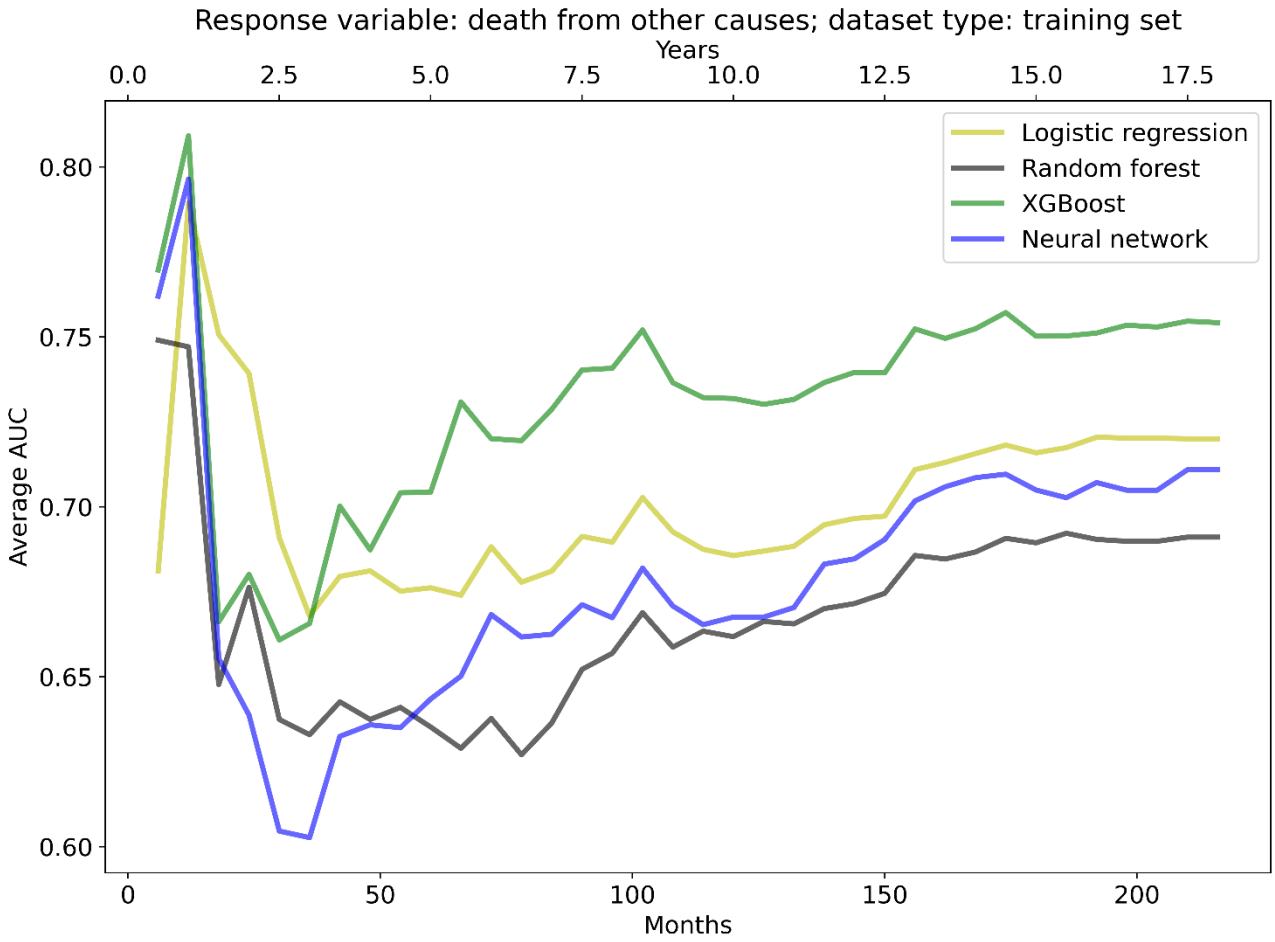

Fig. 29. Average training AUC across different prediction periods for deaths from other causes

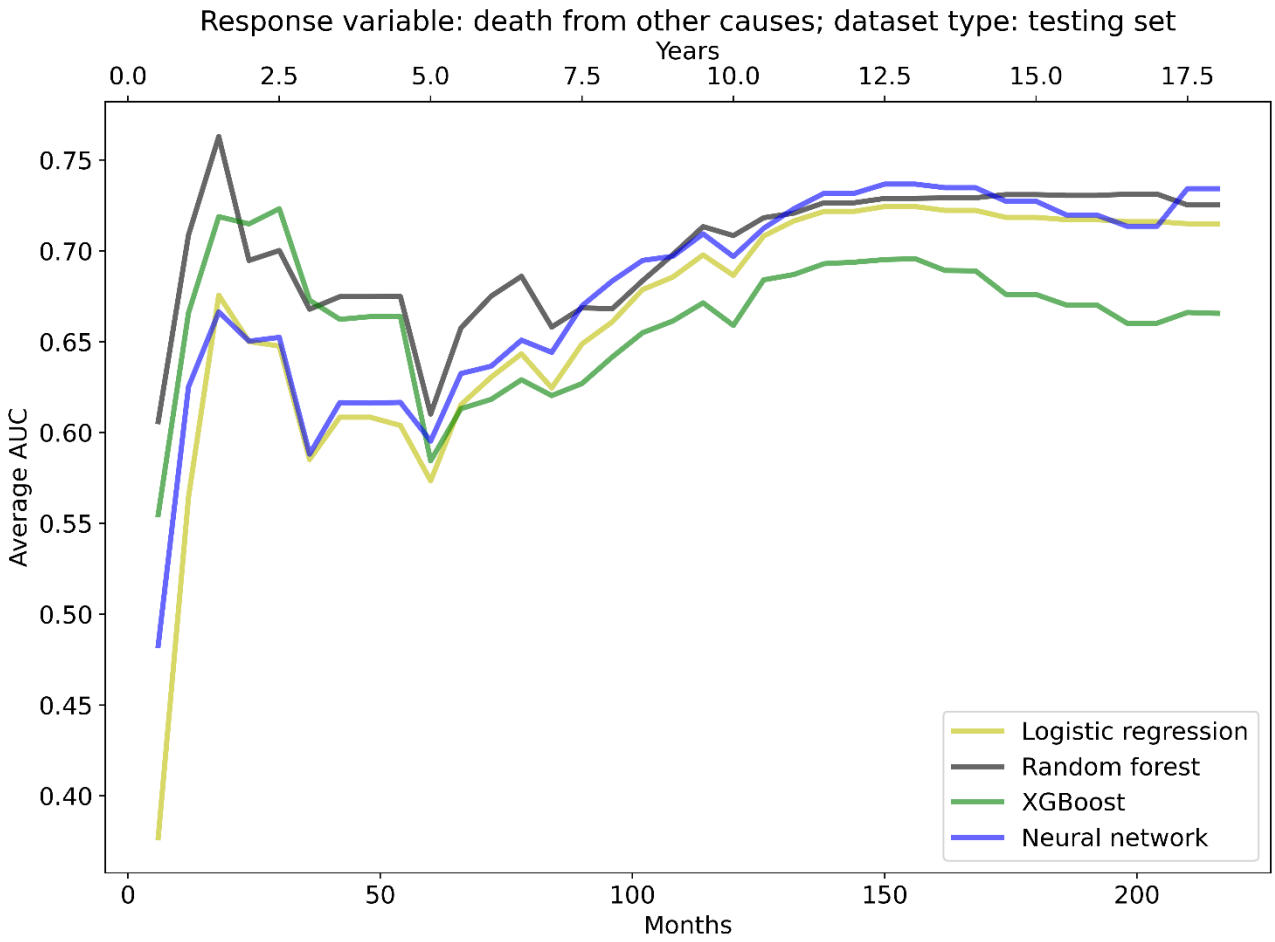

Fig. 30. Average testing AUC across different prediction periods for deaths from other causes

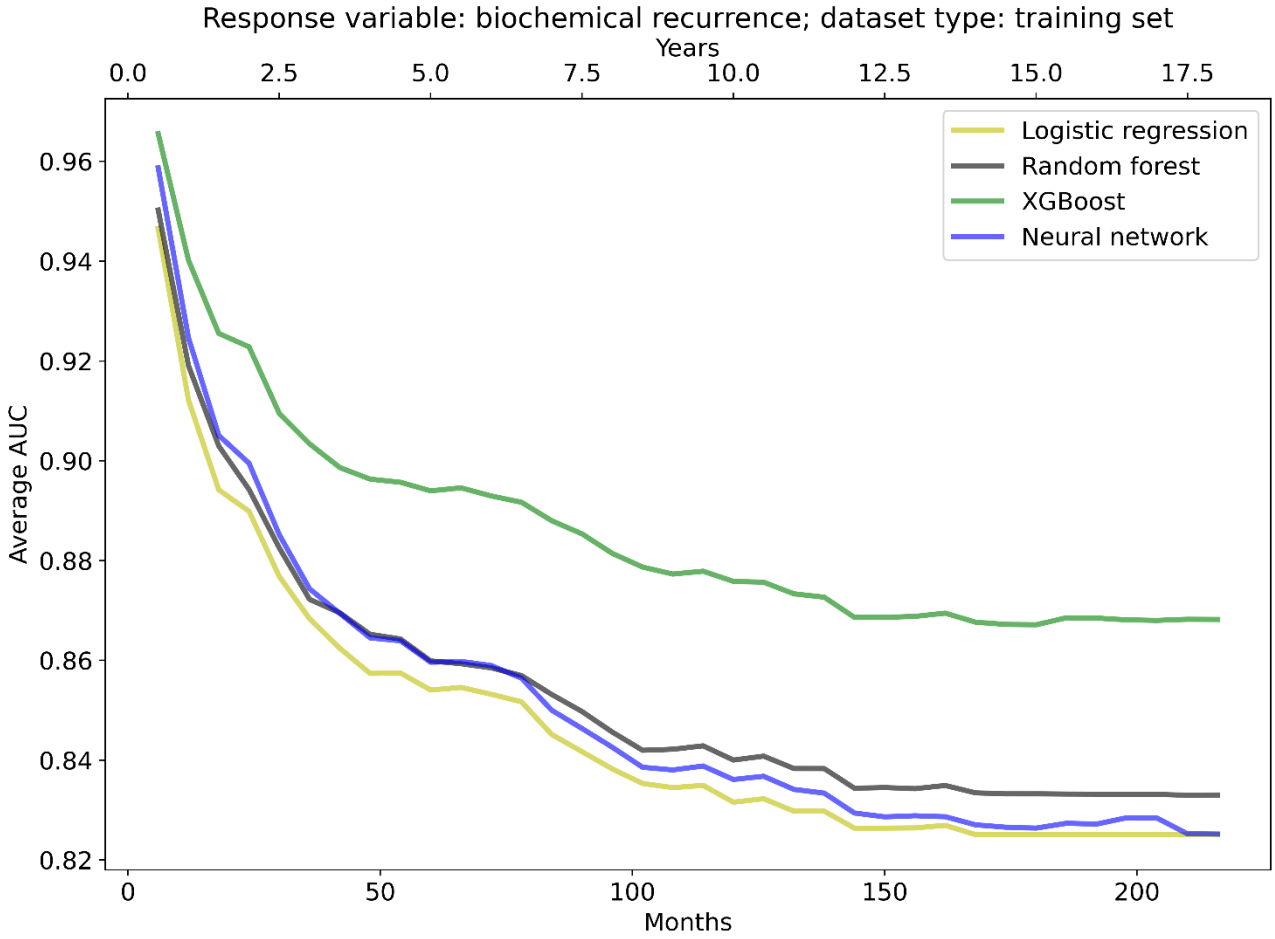

Fig. 31. Average training AUC across different prediction periods for biochemical recurrence

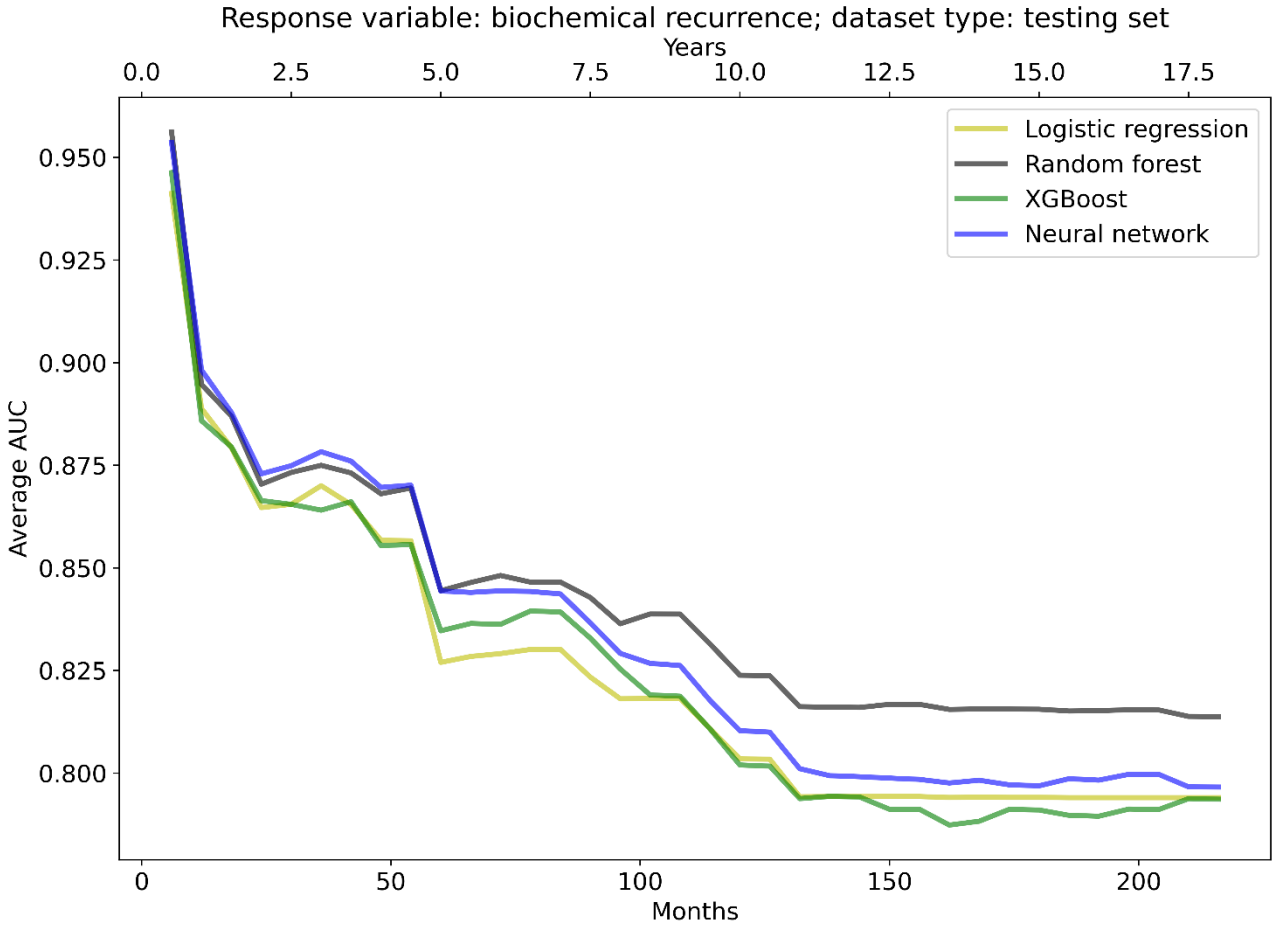

Fig. 32. Average testing AUC across different prediction periods for biochemical recurrence

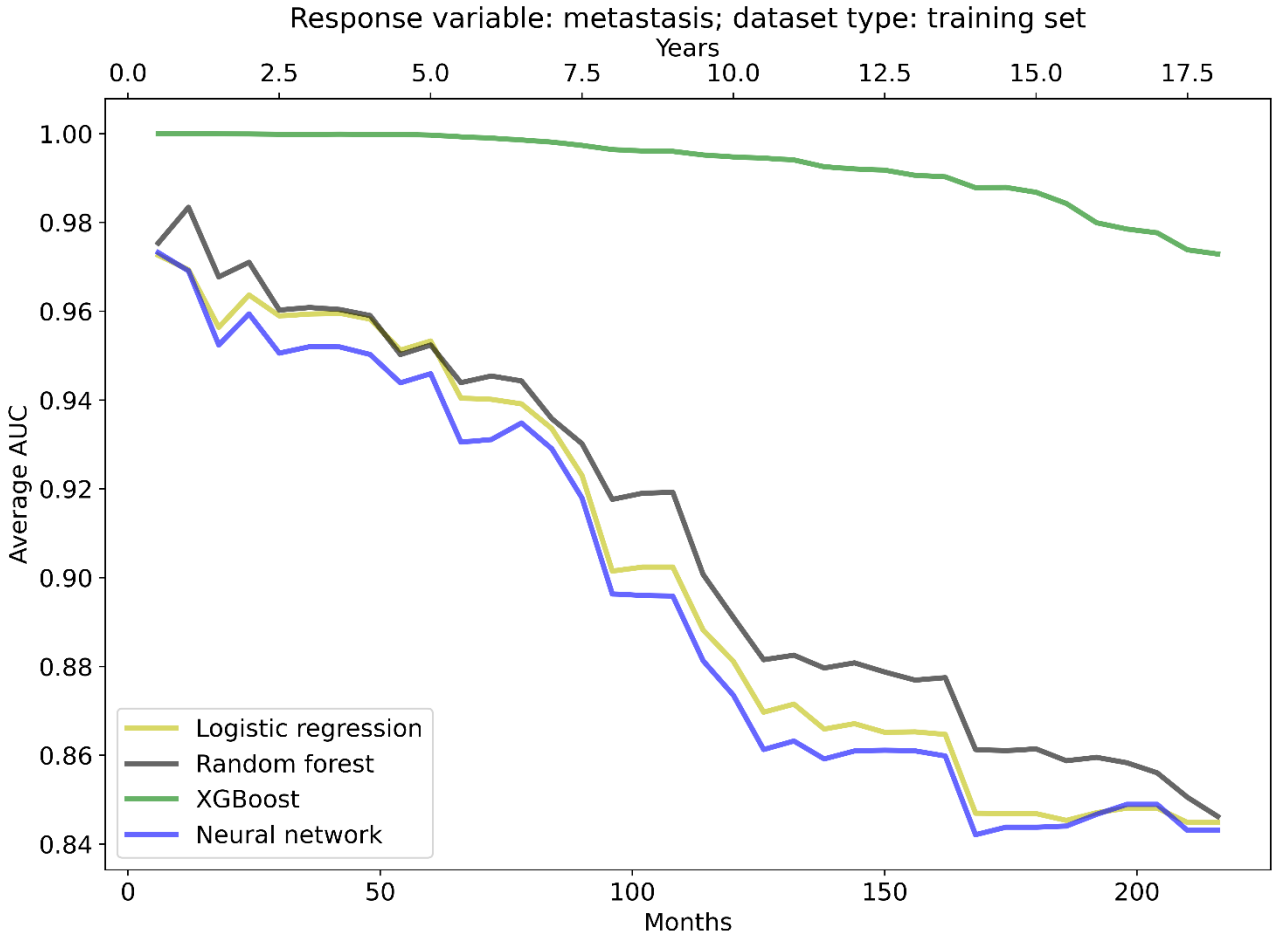

Fig. 33. Average training AUC across different prediction periods for metastasis

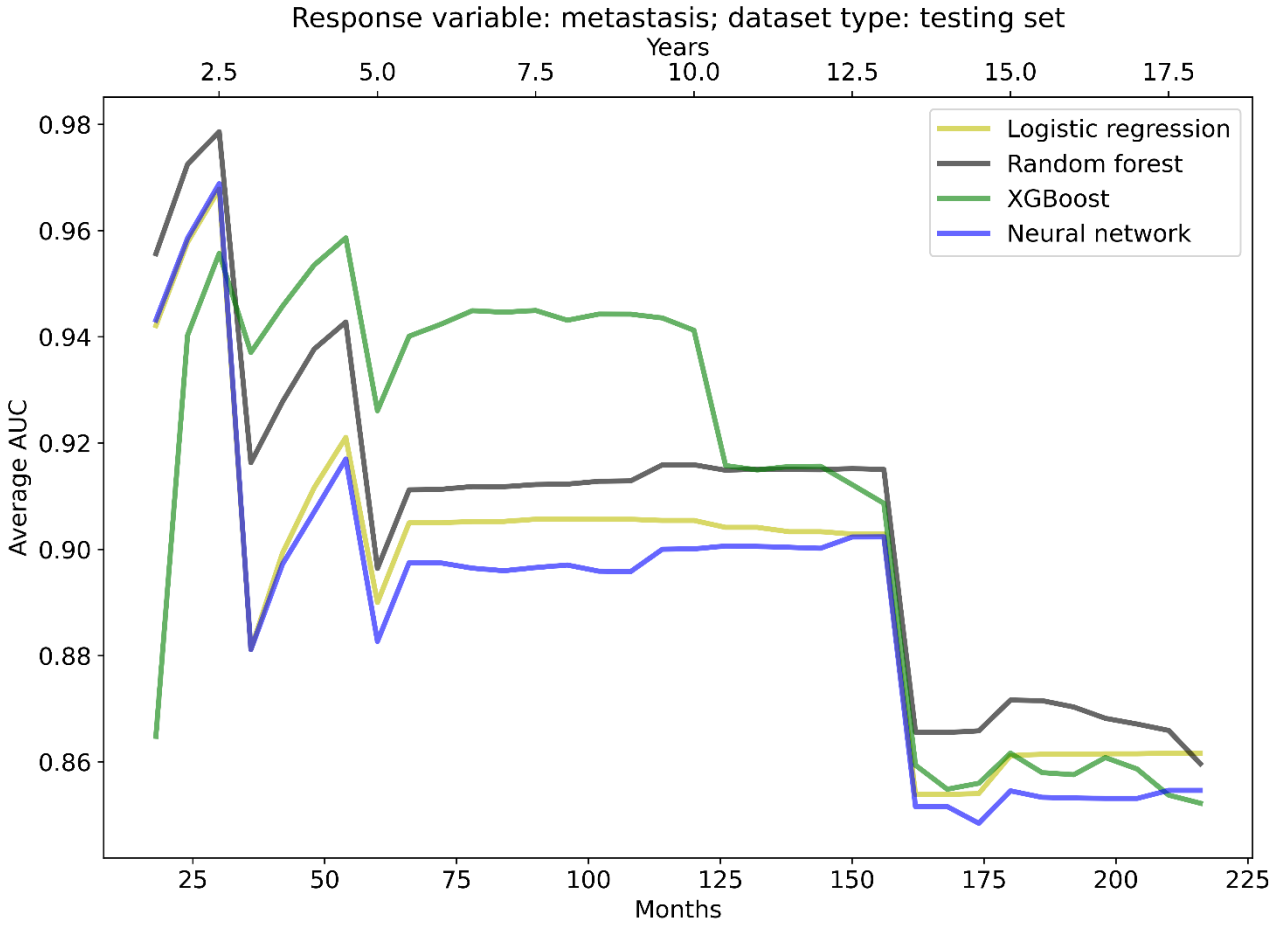

Fig. 34. Average testing AUC across different prediction periods for metastasis