Erasmus School of Economics
Master Thesis
MSc Econometrics and Management Science
Business Analytics and Quantitative Marketing

Certainty Estimation of a Probabilistic Neural Network using Quantile Regression

Author:
Stefan Bovij (456250)

Supervisor:
Kathrin Gruber
Second Assessor:
Andreas Alfons

July 30, 2023

Abstract

Machine learning has become a pivotal factor that distinguishes companies and grants them a competitive advantage. However, a prevailing issue in machine learning is that numerous models make decisions based on historical data without providing explanations for their reasoning — commonly known as the "Black Box Problem". The resulting inability to trust machine learning algorithms' decisions can have catastrophic consequences, especially in high-stakes environments like healthcare and finance. To address this challenge, this study proposes a novel method that facilitates the identification of a machine learning model's decision-making process by performing a quantile regression analysis on its predictions. Remarkably, the quantile regression curves can reveal the variables that have the most substantial impact on the predictions made by the probabilistic neural network, which is the chosen machine learning model for this research. Additionally, the method's capability to provide prediction interval sizes enables the visualization of the prediction certainty of the probabilistic neural network for different values of the feature variables. This approach holds promise in enhancing transparency and understanding in machine learning models, potentially reducing the risks associated with black box algorithms.

keywords: Black Box Problem, machine learning, Parzen window, k-fold cross-validation, probabilistic neural network, deep quantile regression neural network

Contents

1 Introduction

Microsoft investing ten billion dollars in the firm behind ChatGPT, the artificial intelligence (AI) chatbot created by OpenAI, was a major technology-related news event that made headlines in January of this year (The Guardian, 2023b). When OpenAI launched version 3.5 of ChatGPT for public use in November 2022 it took the world by storm. Users have been amazed by its diverse capabilities, ranging from writing emails and translating text to debugging code. It is seen as one of the most innovative AI technologies in the industry and therefore it is no surprise that the website broke the record for the fastest-growing consumer internet app ever by reaching 100 million unique users just two months after launch (The Guardian, 2023a). In the meantime, ChatGPT has been experiencing steady growth and in April it reached approximately 1.8 billion website visits in a single month (Similarweb, 2023).

Alongside the rise of AI, machine learning algorithms have experienced a resurgence in popularity. Simply using Google Trends and searching for machine learning related keywords shows a clear uptrend over the last decade. Deep learning models, in particular, have become instrumental in driving advanced AI applications such as ChatGPT. This large language model (LLM) uses deep learning techniques to generate text that closely resembles human language. The competition among major companies, such as Microsoft, Meta, and Alphabet in the realm of machine learning platforms is fierce. These businesses strive to attract customers by offering comprehensive platform services that encompass various machine learning activities, from data collection to model building. As AI becomes more practical in enterprise environments and machine learning becomes crucial to business operations, the battle for dominance in the machine learning industry will only intensify. A notable instance of such a battle occurred when Alphabet unveiled their chatbot named Bard shortly after Microsoft's announcement of their investment in OpenAI (CNBC, 2023).

Machine learning refers to a branch of AI where software applications can improve their ability to make predictions without being explicitly programmed. By utilizing historical data, machine learning algorithms generate predictions for new data inputs. Machine learning finds widespread application in various domains. For instance, recommendation systems utilize machine learning techniques to provide personalized suggestions (Isinkaye et al., 2015). Additionally, it is employed for tasks like spam filtering, automating business processes, and fraud detection. Machine learning holds immense significance as it enables organizations to gain insights into customer behavior and analyze patterns in business operations. Prominent companies rely heavily on machine learning as a fundamental component of their operations. Consequently, machine learning has become a crucial factor that sets companies apart and gives them a competitive edge.

One of the problems with machine learning nowadays is that many models make decisions based on historical data without providing any explanations for their reasoning. While some

models allow us to grasp their inner workings by understanding the underlying mathematics like neural networks, they can still pose a challenge in comprehending how the individual neurons cooperate to produce the final output. Certain models are even more perplexing since they have internal workings that are entirely invisible to the user. This lack of explainability and the hiding of internal computations within multiple layers of a model are commonly referred to as the "Black Box Problem" (Castelvecchi, 2016). While being capable of applying a machine learning model is valuable, understanding how these models arrive at their decisions, which variables influenced those decisions, and the level of certainty associated with their predictions is equally crucial. When a data scientist uses a model without understanding how it utilizes the data or if it incorporates all available information, they may overlook the need to adjust model parameters, perform data cleaning, or even remove certain variables (Ribeiro et al., 2016b). Additionally, recognizing that a certain model might not be the best fit for the data set could prove challenging in such situations. Apart from the data scientist who works on the model, every machine learning platform also has different stakeholders who require varying degrees of insight into the model's functionality and reasoning (Zednik, 2021).

The significance of solving the "Black Box Problem" for the wider acceptance and integration of AI and machine learning has been a prominent topic in the literature over the past few years. Various researchers have emphasized that the lack of interpretability leads to a crucial consequence: the inability to trust and act upon the decisions made by a model. Users are less likely to trust and relinquish control to machines when they do not understand how the models operate (Ribeiro et al., 2016b). While the appeal of AI lies in its potential to be more reliable than humans in handling complex tasks, it is crucial to differentiate between trust and reliability, especially when moral implications are involved (Von Eschenbach, 2021). In critical domains where transparency is essential for accountability and regulatory compliance, the limited interpretability of machine learning models significantly restricts their applicability. For instance, in healthcare, finance, and autonomous vehicles, interpretability is crucial for establishing trust and ensuring the model is making ethical and reliable decisions (Guidotti et al., 2018). Take the finance industry, where critical decisions such as loan approvals are involved. Compliance teams seek to understand which variables are influencing the model's predictions to ensure adherence to regulatory requirements. In addition to the lack of trust arising from inadequate understanding, using uninterpretable machine learning applications has another drawback. According to Guidotti et al. (2018), these models also raise ethical concerns related to bias and robustness. If the training data contains human biases, there is a significant risk that the machine learning model might unintentionally make incorrect and unfair decisions. Especially, if we lack a comprehensive understanding of the model it becomes challenging to discern whether the issue stems from biased or insufficiently representative training data, or if the model itself is simply unreliable. As a result, this sequence of events could further erode trust in such opaque decision systems.

There are essentially two approaches to address the "Black Box Problem". The first way involves exercising caution by limiting the use of complex deep learning applications and implementing regulations, particularly in critical industries where the potential harm of an unreliable or untrustworthy application is high. For example, Price (2017) pleads for well-suited regulatory oversight in the healthcare sector. However, he expresses the concern that certain government agencies' existing regulations may be too rigid, potentially impeding the innovation and advancement of more effective algorithms. Therefore, an alternative path should be considered, emphasizing a more flexible regulatory strategy. This approach would seek a delicate balance between public and private oversight, granting government agencies a mediating role instead of a dominating one. Similarly, Bathaee (2017) states that we should avoid implementing an excessively detailed and rigid regulatory framework outlining transparency standards for AI design and utilization. Instead, he opts for a sliding scale approach, where the regulatory system adjusts existing causation and intent tests based on whether the algorithm is allowed to operate autonomously and the model's level of transparency. Rudin (2019) takes a more assertive stance and proposes that moving forward, critical decisions should solely rely on inherently interpretable models. She states that black box algorithms could cause severe harm to society and may preserve undesirable practices. Moreover, adopting solely interpretable models would alleviate the need for extensive regulatory efforts.

The other approach aims to provide explanations for the inner workings and decisions of black box models. Methods attempting to achieve this goal can be categorized into two groups. The first category primarily focuses on describing the black box algorithm itself, while the second category is more concerned with explaining the model's decisions even without comprehending its internal operations (Guidotti et al., 2018). Recent advancements in explainable AI that focus on identifying influential features and improving model interpretability have shown promise in addressing the black box problem. Researchers have explored various techniques such as neural network visualization, attribution methods, local interpretable model-agnostic explanations (LIME), and Shapley additive explanations (SHAP) to gain insights into the decision-making process of machine learning models. Going through this list, Yosinski et al. (2015) stated that our comprehension of the inner workings of convolutional neural networks (CNNs) and deep neural networks (DNNs) lags behind the significant advancements these models have achieved in recent years. Therefore, they provide two tools to enable the visualization of these types of neural networks. The first tool allows visualization of the activations generated at each layer of a CNN, while the second tool can visualize the features for each layer of a DNN. Attribution methods form another popular research domain aimed at improving users' understanding of certain machine learning models. For instance, Sundararajan et al. (2017) developed an axiomatic attribution method, enabling the attribution of a DNN's prediction to its inputs. Interestingly, this method does not focus on interpreting the computations or representations of individual neurons. Instead, it examines the overall behavior of the network based on a specific input.

Another fascinating method is LIME, introduced by Ribeiro et al. (2016a). They argue that this versatile technique can provide interpretable explanations for the predictions made by any classifier. It achieves this by utilizing an interpretable model to locally approximate the classifier around a given prediction. LIME proves to be valuable for different users, whether they aim to gain insights into a model's predictions, improve unreliable models, make model comparisons, or evaluate trustworthiness. Lastly, Lundberg and Lee (2017) propose the SHAP framework for interpreting model predictions. This comprehensive framework combines six existing methods, including LIME. According to the researchers, it addresses the common challenge of understanding the relationships between individual methods and determining when one method is more advantageous than another. Furthermore, the SHAP framework provides insights that lead to new and enhanced methods. Despite the progress in addressing the "Black Box Problem", challenges and limitations persist. The complexity of modern machine learning models, trade-offs between accuracy and interpretability, and the need for domain-specific explanations present ongoing research challenges (Lipton, 2018).

The primary objective of this thesis is to introduce a new method that facilitates the identification of a machine learning model's decision-making process while providing insights into the certainty of its decisions. To achieve this, a probabilistic neural network (PNN) is selected as the machine learning model for demonstration purposes (Specht, 1990). The PNN is chosen due to its simplicity, intuitiveness, and ability to produce probabilities as output, making it an ideal candidate for showcasing the method while ensuring comprehension. Furthermore, this model is widely used for classification problems and to ensure clarity and interpretability of the results, a binary classification data set is selected. It is crucial to note that the model's accuracy is not a primary concern in this study. Instead, the objective is to gain insights into the underlying factors that drive the model's classifications rather than prioritizing its overall accuracy. After the PNN is applied to the data set, a quantile regression (Koenker & Bassett Jr, 1978) is fitted on the model predictions for all feature variables, using a deep quantile regression neural network (DQRNN). The main question that this research tries to answer revolves around whether the resulting quantile curves can provide a deeper understanding of the model's decision-making process and help identify the variables that play the most critical role in making these decisions. Moreover, examining the distance between the curves, which represents the size of the prediction intervals for the probabilities outputted by the PNN, for various values of a feature variable indicates the model's level of certainty when making predictions. Undoubtedly, quantile regression has been utilized multiple times in the literature to establish the relationship between two variables or generate prediction intervals for a specific dependent variable. However, it has not yet been applied to the output of a machine learning model as a method to gain deeper insights into the model's decisions.

This paper will with the methodology section, where the proposed method will be explained and clearly outlined. Afterward, the binary classification data set employed for this thesis and

the necessary data cleaning and transformations are thoroughly discussed. Following that, the results obtained from the conducted quantile regression analysis will be shown. Remarkably, the quantile regression curves appear capable of revealing the variables that had the most substantial impact on the predictions made by the PNN. Finally, a comprehensive summary of the observations made in this study is provided. Based on these findings, appropriate conclusions will be drawn and future research proposals will be discussed.

2 Methodology

This section will delve into the various methods employed in this study. Firstly, the functioning of an ANN and a PNN will be explained. To gain a deeper understanding of the PNN's workings, we will briefly describe the Parzen window method utilized within the PNN, along with the cross-validation technique used to determine the optimal bandwidth for this Parzen window estimator. Afterward, the key differences between an ANN and PNN will be highlighted. Lastly, we will define the quantile regression loss function and the DQRNN. The purpose of this DQRNN is to fit a quantile regression on the posterior probabilities predicted by the PNN.

2.1 Artificial Neural Network

An ANN is a feedforward neural network widely used in machine learning applications. The basic structure of an ANN consists of multiple layers of interconnected nodes, or neurons, which process and transmit information (see Figure 1). Each neuron receives information from the neurons in the previous layer, applies an activation function to this input, and produces an output transmitted to the neurons in the next layer. The first layer of neurons is the input layer, the layers in the middle are called hidden layers, and the last layer is the output layer (Goodfellow et al., 2016). Usually, the input data is represented as a vector or an array, and each neuron in the input layer receives a single element of the input vector. The neurons in the last layer contain the neural network's output, which can be used to make predictions or classify data.

The neurons in an ANN are associated with a set of parameters or weights learned during the training process. The weights determine how strongly the input from each node in the previous layer affects the output of the current neuron. The training process involves adjusting the weights to minimize a loss function, which measures the difference between the predicted and true output of the neural network (Haykin, 2009).

These weights are updated using a technique called backpropagation, which involves computing the gradient of the loss function with respect to the weights and using this gradient to adjust the weights in the direction of the steepest descent. This process is repeated for several iterations until the loss function is minimized, and as a result, the neural network's performance should have improved (Rumelhart et al., 1986).

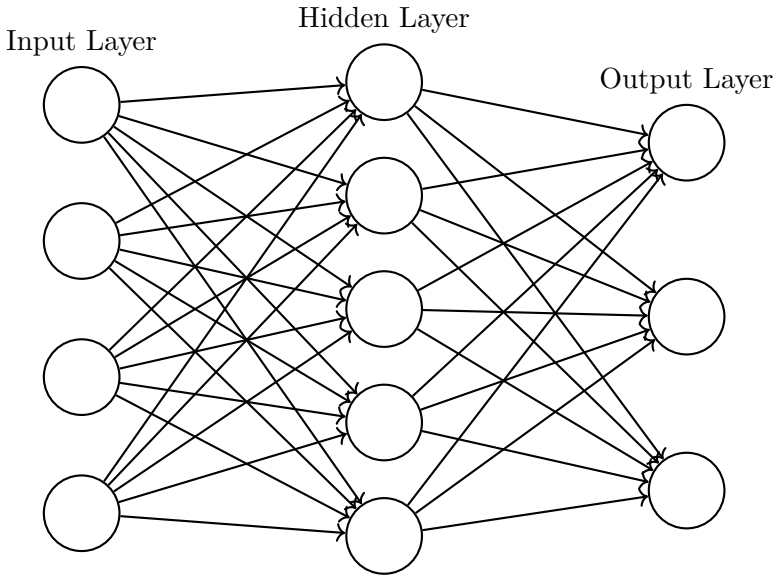


Figure 1: The structure of a three-layer ANN with four input features and three output classes.

To determine whether the input of a neuron is important to the neural network, or in other words, whether a neuron should be activated, the neural network makes use of activation functions. These activation functions are used to introduce nonlinearity into the network so it can better fit the results and improve its accuracy. One commonly used activation function in an ANN is the sigmoid function (see Section 2.4, Equation 10), which maps any input value to a value between zero and one. Another widely used activation function is the rectified linear unit (ReLU) function. It maps any input value less than zero to zero and any value greater than zero to the input value itself (LeCun et al., 2015). The ReLU function offers several advantages over the sigmoid function. Firstly, it reduces the likelihood of a vanishing gradient problem and it is more computationally efficient. Additionally, networks using ReLU activation functions generally exhibit better convergence performance (Krizhevsky et al., 2012). Therefore, in this study, ReLU activation functions will be employed for the DQRNN (see Section 2.7).

Many variations exist of an ANN, including DNNs, CNNs, and recurrent neural networks (RNNs). They are designed to fit specific data types, such as images or sequential data. This shows that the ANN forms the basis for many solutions to various machine learning problems.

2.2 Parzen Window

The Parzen window or Parzen-Rosenblatt window method, invented independently by Rosenblatt (1956) and Parzen (1962), is a non-parametric kernel density estimation technique. It is used to estimate the probability density function of a random variable using a set of samples. The basic idea is to estimate the density at a point by averaging the contributions of kernel functions centered at each sample point. The kernel function is a probability density function that determines the shape of the window around each sample. The window's width is also

determined by a bandwidth parameter, which controls the trade-off between bias and variance of the density estimate. Later, Cacoullos (1964) extended Parzen's results to cover the case of estimating a multivariate density function.

Mathematically, the Parzen window estimator for a d-dimensional random variable X with probability density function f_X can be expressed as:

$$\hat{f}_X(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right), \tag{1}$$

where x is the input vector, n is the number of samples, h is the bandwidth, K is the kernel function, and x_i is the i-th sample from X. The kernel function K is often chosen to be a symmetric probability density function centered at zero, such as the standard multivariate Gaussian kernel:

$$K(u) = \frac{1}{(2\pi)^{\frac{d}{2}}} \exp\left(-\frac{1}{2}u^\intercal u\right), \tag{2}$$

where u is the argument of the kernel and d is the dimension of u. After plugging in the chosen kernel function in Equation 1, the Parzen window estimator $\hat{f}_X(x)$ gives an estimate of the probability density function of X evaluated at x.

The choice of bandwidth h is critical for the performance of the Parzen window method. If the bandwidth is too small, the estimate will have a high variance and be sensitive to noise in the data. If the bandwidth is too large, the estimate will have a high bias and miss important features of the data. Various techniques, such as cross-validation and plug-in methods, have been proposed to select the optimal bandwidth (Silverman, 1986). This thesis will use cross-validation to determine the optimal bandwidth h (see Section 2.3).

The Parzen window method is especially useful for non-parametric density estimation, where the underlying probability distribution is unknown or difficult to model using parametric distributions. Although the choice of the kernel function is required, this kernel density estimation method is still considered non-parametric as it does not assume any specific shape for the distribution being estimated (Silverman, 1986). Additionally, Epanechnikov (1969) found that any reasonable kernel yields nearly optimal results. The Parzen window method has found widespread application in various fields, especially in scenarios where non-parametric classifiers are commonly used, such as pattern recognition (Babich & Camps, 1996) and image processing (Gao, 2010). In particular, it serves as a fundamental component for the PNN (see Section 2.4) introduced by Specht (1990). This model uses the Parzen window method to estimate the class-conditional probability densities of input features.

2.3 Cross-validation

Cross-validation is a widely used technique for assessing and selecting optimal parameters in machine learning and statistical modeling (Hastie et al., 2009). Furthermore, it provides a reliable estimate of a model's performance by partitioning the available data set into training and validation subsets. This method allows for an objective evaluation of different parameter settings and helps to prevent overfitting. In the context of the Parzen window estimator, cross-validation can be employed to determine the optimal bandwidth parameter h, which controls the width of the window function and affects the bias-variance trade-off in density estimation (see Section 2.2). Various cross-validation strategies exist, such as leave-one-out cross-validation, k-fold cross-validation, or stratified cross-validation. The choice of which version to use depends on the specific requirements of the problem at hand (Kohavi et al., 1995).

With cross-validation, the data set is divided into a training and validation set. The model is trained on the training set using a specific value for the bandwidth parameter. Then its performance is evaluated on the validation set using an appropriate performance metric. This process is repeated multiple times, with different data set partitions used as the training and validation set each time. Finally, the performance metrics obtained from each iteration are averaged to estimate the model's performance with the current bandwidth parameter.

One popular cross-validation technique is leave-one-out cross-validation, where each data point is sequentially used as the validation set, and the model is trained on the remaining data. This approach provides a reasonably unbiased estimate of the model's performance. However, it can be computationally expensive for large data sets and suffer from high variance in some problems (Efron & Tibshirani, 1997).

Another commonly used technique, also used for this thesis, is k-fold cross-validation. In k-fold cross-validation, the data set is divided into k equally-sized subsets or folds (Kohavi et al., 1995). The model is trained k times, where each time $k-1$ folds are utilized as the training set while the remaining fold is used as the validation set. The performance metrics obtained from each validation set are averaged to obtain the overall performance estimate for that specific parameter setting. This process is repeated for different bandwidth values, and afterward, the bandwidth with the highest performance metric across all folds is selected. The choice of the number of folds and this performance metric should be carefully considered based on the characteristics of the data and the goal of the density estimation task (Forman & Scholz, 2010). Commonly used metrics include accuracy, precision, mean squared error, or log-likelihood. In this thesis, 10-fold cross-validation will be employed and the accuracy of the PNN (see Section 2.4) will be used as the performance metric.

In summary, the cross-validation method is a robust and objective way to estimate the performance of the PNN and select the optimal bandwidth parameter value. It also helps to assess the generalization performance of the Parzen window estimator and provides insights into its bias-variance trade-off. This allows us to avoid overfitting or underfitting.

2.4 Probabilistic Neural Network

PNNs have been applied successfully in numerous domains, including speech recognition (Morin & Bengio, 2005), medical diagnosis (Hirschauer et al., 2015), image classification (Varuna Shree & Kumar, 2018), and bioinformatics (Georgiou et al., 2004). It is a feedforward neural network often used for classification tasks. It utilizes a non-parametric approach to classification based on the Bayes decision rule and consists of four layers (see Figure 2). The basic idea of a PNN is to represent each input pattern as a probability density function (PDF) evaluated at that point. Hence, for a certain input pattern, multiple PDFs are estimated by deriving the Parzen window estimator (see Section 2.2) for each class separately using only the training data of the corresponding class (Specht, 1990). The input pattern will ultimately be assigned to the class with the highest class-conditional density estimate.

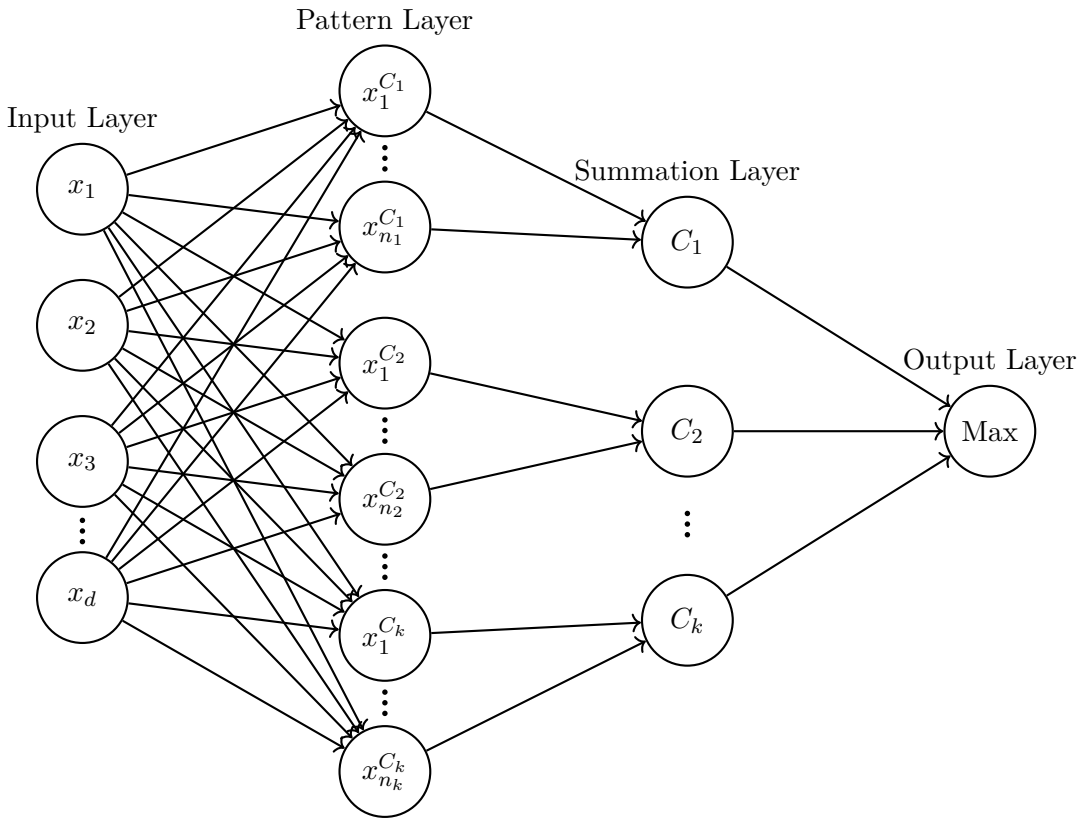


Figure 2: The structure of a PNN for a multi-class classification problem with d input features, $\sum_{j=1}^{k} n_j$ training samples, and k classes. Furthermore, $x_i^{C_j}$ is the i-th training sample belonging to class C_j.

The first layer of the PNN is the input layer, where similar to the ANN (see Section 2.1), the input features of an input pattern are distributed to all of the neurons in the next layer. Next, in the pattern layer, each pattern neuron uses the input pattern and a single training sample to calculate the value of the kernel function that is part of the Parzen window estimator

(see Equation 1). This implies that the number of neurons in the pattern layer equals the number of training samples (see Figure 2). Furthermore, the activation function of a PNN is different from the commonly used sigmoid or ReLU activation functions in neural networks that use backpropagation (Rumelhart et al., 1986). For example, if the Gaussian kernel is chosen (see Equation 2), the i-th pattern neuron of the PNN uses the following exponential activation function:

$$g(x) = \exp\left(-\frac{(x - x_i)^\intercal (x - x_i)}{2h^2}\right), \tag{3}$$

where x is the input vector, x_i is the i-th training sample and h is again the bandwidth parameter. Multiplying Equation 3 with the Gaussian kernel constant $(2\pi)^{-\frac{d}{2}}$ is not necessary since it will cancel out in the last layer of the PNN. This will be discussed in more detail below. Other variations of exponential or even non-exponential functions can also be used. It all depends on the chosen kernel function for the Parzen window estimator, and no single function is proven to be always better than the others (Specht, 1990).

After, the calculated values from the pattern layer are summed up in the summation layer. In this layer, each summation neuron sums the outputs of all the pattern neurons that used a training sample belonging to the same class. Therefore, in the resulting PNN, every output class will have exactly one corresponding summation neuron. Finally, the input pattern is assigned to a class in the output layer. This decision is based on the posterior probability of class C_j for an input vector x, which can be calculated using Bayes' rule:

$$P(C_j|x) = \frac{P(x|C_j)P(C_j)}{P(x)} \text{ for } j = 1, \ldots, k\,, \tag{4}$$

where $P(x|C_j)$ is the likelihood function of x, $P(C_j)$ is the prior probability of class C_j and $P(x)$ is the marginal probability of x. Theoretically, the PNN classifies the input feature vector as the class with the highest posterior probability. Therefore, assuming a binary classification problem with classes C_1 and C_2, this results in the following Bayes' decision rule:

$$\begin{aligned}
d(x) = C_1 \text{ if } \quad & P(C_1|x) > P(C_2|x) && \Rightarrow \\
& \frac{P(x|C_1)P(C_1)}{P(x)} > \frac{P(x|C_2)P(C_2)}{P(x)} && \Rightarrow \\
& P(x|C_1)P(C_1) > P(x|C_2)P(C_2) && \Rightarrow \\
& P(x|C_1) > \frac{P(C_2)}{P(C_1)} P(x|C_2) && (5) \\
d(x) = C_2 \text{ if } \quad & P(C_1|x) < P(C_2|x) && \Rightarrow \\
& P(x|C_1) < \frac{P(C_2)}{P(C_1)} P(x|C_2), && (6)
\end{aligned}$$

where $d(x)$ is the function representing the class assigned to the input pattern x. The decision rule above can easily be extended to a multi-class problem, where x will be classified as the class with the highest corresponding posterior probability of all classes. However, this section will discuss mainly binary classification for simplicity purposes. The likelihood functions $P(x|C_1)$ and $P(x|C_2)$ are the class-conditional densities of class C_1 and C_2 evaluated at x. They can be estimated using the Parzen window method (see Section 2.2). According to Equations 5 and 6, multiplying the class-conditional density of class C_2 with the ratio of prior probabilities $\left(\frac{P(C_2)}{P(C_1)}\right)$ and checking if the result is smaller or larger than the class-conditional density of class C_1 determines whether x will be assigned to class C_1 or C_2.

So far, each neuron in the summation layer calculated the summation in Equation 1 (without the chosen Kernel's constant term) for its corresponding class. These values are all transferred to the output layer, where they are plugged into Equation 5 and 6 to make a decision. If the Gaussian kernel from Equation 2 is chosen, this results in the following expressions:

$$
\begin{aligned}
d(x) = C_1 \text{ if } \quad & P(x|C_1) > \frac{P(C_2)}{P(C_1)} P(x|C_2) && \Rightarrow \\
& \frac{1}{(2\pi)^{\frac{d}{2}} n_1 h^d} \sum_{C_1} g(x) > \frac{P(C_2)}{P(C_1)} \cdot \frac{1}{(2\pi)^{\frac{d}{2}} n_2 h^d} \sum_{C_2} g(x) && \Rightarrow \\
& \sum_{C_1} g(x) > \frac{P(C_2)}{P(C_1)} \cdot \frac{n_1}{n_2} \sum_{C_2} g(x) && \Rightarrow \\
& \sum_{C_1} g(x) > Z \cdot \sum_{C_2} g(x) && (7) \\
d(x) = C_2 \text{ if } \quad & P(x|C_1) < \frac{P(C_2)}{P(C_1)} P(x|C_2) && \Rightarrow \\
& \sum_{C_1} g(x) < Z \cdot \sum_{C_2} g(x), && (8)
\end{aligned}
$$

where $\sum_{C_j} g(x)$ (see Equation 3) is the output of the summation neuron belonging to class C_j and n_j is the number of training samples from class C_j. As mentioned earlier, the Gaussian kernel constant and the scalar h^d, which do not depend on one of the classes, cancel out in Equation 7 and 8. What is left are the outputs of the summation layer and the constant Z. Note that Z is the ratio of prior probabilities $\left(\frac{P(C_2)}{P(C_1)}\right)$ divided by the ratio of training samples $\left(\frac{n_2}{n_1}\right)$. If it is the case that the number of training samples from class C_1 and C_2 are obtained in the same proportion as their prior probabilities, the constant Z simplifies to one. This implies that, for binary and multi-class classification problems, the output layer of the PNN will assign x to the class with the largest corresponding summation neuron output. In other words, it simply takes the maximum of all the individual summations calculated for each class in the summation layer and classifies x accordingly.

However, this thesis is not only interested in correctly classifying as many input patterns as possible. Instead, the main focus is to retrieve the PDF estimates for each class evaluated at an individual input pattern. After, these estimates can be used to calculate the posterior probability

that an input pattern x belongs to a certain class, $P(C_j|x)$. To compute these probabilities, Bayes' rule from Equation 4 is again used as the starting point. According to Gelman et al. (2013), we can substitute the marginal probability of x by the likelihood functions times their corresponding prior, summed over all k classes. This results in the following expression for the posterior probability of class C_j:

$$P(C_j|x) = \frac{P(x|C_j)P(C_j)}{P(x)} = \frac{P(x|C_j)P(C_j)}{\sum_{i=1}^{k} P(x|C_i)P(C_i)} \text{ for } j = 1, \ldots, k\,, \tag{9}$$

where it must hold that $\sum_{i=1}^{k} P(C_i) = 1$. Calculating these posterior probabilities for a binary classification problem can be done using a specific version of a sigmoid function (Bishop & Nasrabadi, 2006), also mentioned earlier as a common activation function for an ANN (see Section 2.1). The following derivations of the posterior probabilities for C_1 and C_2 show this:

$$\begin{aligned}
P(C_1|x) &= \frac{P(x|C_1)P(C_1)}{P(x|C_1)P(C_1) + P(x|C_2)P(C_2)} = \frac{1}{1 + \frac{P(x|C_2)P(C_2)}{P(x|C_1)P(C_1)}} \\
&= \frac{1}{1 + \exp\left(\log\left(\frac{P(x|C_2)P(C_2)}{P(x|C_1)P(C_1)}\right)\right)} = \frac{1}{1 + \exp\left(-\log\left(\frac{P(x|C_1)P(C_1)}{P(x|C_2)P(C_2)}\right)\right)} \\
&= \frac{1}{1 + \exp\left(-y(x)\right)} \qquad (10) \\
P(C_2|x) &= \frac{1}{\exp\left(\log\left(\frac{P(x|C_1)P(C_1)}{P(x|C_2)P(C_2)}\right)\right) + 1} = \frac{1}{\exp\left(y(x)\right) + 1} \qquad (11) \\
&= \frac{\exp\left(-y(x)\right)}{1 + \exp\left(-y(x)\right)} = \frac{1 + \exp\left(-y(x)\right)}{1 + \exp\left(-y(x)\right)} - \frac{1}{1 + \exp\left(-y(x)\right)} \\
&= 1 - P(C_1|x).
\end{aligned}$$

Here $y(x)$ is introduced as a substitute for $\log\left(\frac{P(x|C_1)P(C_1)}{P(x|C_2)P(C_2)}\right)$ to simplify both expressions. As can be seen, Equation 10 is an example of a logistic sigmoid function (Han & Moraga, 1995). Equation 11 is slightly different and is simply an inverted logistic S-curve, where the logistic sigmoid function is reflected in the y-axis. Furthermore, the last part of the derivation above shows that the following symmetric property holds: $\gamma(-y) = 1 - \gamma(y)$, where $\gamma(y)$ is the logistic sigmoid function. If again a Gaussian kernel is chosen and the number of training samples from class C_1 and C_2 are obtained in the same proportion as their prior probabilities, $y(x)$ simplifies to the following expression:

$$\begin{aligned}
y(x) &= \log\left(\frac{P(x|C_1)P(C_1)}{P(x|C_2)P(C_2)}\right) = \log\left(\frac{(2\pi)^{\frac{d}{2}} n_2 h^d}{(2\pi)^{\frac{d}{2}} n_1 h^d} \cdot \frac{\sum_{C_1} g(x)}{\sum_{C_2} g(x)} \cdot \frac{P(C_1)}{P(C_2)}\right) \\
&= \log\left(\frac{P(C_1)}{P(C_2)} \cdot \frac{n_2}{n_1} \cdot \frac{\sum_{C_1} g(x)}{\sum_{C_2} g(x)}\right) = \log\left(Z^{-1} \cdot \frac{\sum_{C_1} g(x)}{\sum_{C_2} g(x)}\right),
\end{aligned} \tag{12}$$

where Z^{-1}, similar to Equations 7 and 8, equals one again. Hence, to calculate the posterior probabilities for a binary classification problem, functions 10 and 11 can be used where $y(x)$ is the logarithm of the ratio of the two summation neuron outputs $\left(\frac{\sum_{C_1} g(x)}{\sum_{C_2} g(x)}\right)$.

For a multi-class classification problem with k classes, the posterior probabilities are calculated using the softmax function (Bishop & Nasrabadi, 2006). This can be derived in the following way:

$$\begin{aligned} P(C_j|x) &= \frac{P(x|C_j)P(C_j)}{\sum_{i=1}^{k} P(x|C_i)P(C_i)} \\ &= \frac{\exp\left(\log\left(P(x|C_j)P(C_j)\right)\right)}{\sum_{i=1}^{k} \exp\left(\log\left(P(x|C_i)P(C_i)\right)\right)} \\ &= \frac{\exp\left(y_j(x)\right)}{\sum_{i=1}^{k} \exp\left(y_i(x)\right)} \qquad \text{for } j = 1, \ldots, k\ , \end{aligned} \tag{13}$$

which is also known as the normalized exponential. This time $y_j(x)$ is used as a substitute for $\log\left(P(x|C_j)P(C_j)\right)$ to simplify the notation. If the Gaussian kernel is chosen and training samples for each class C_j are again obtained in the same proportion as their corresponding prior probabilities, $y_j(x)$ can be simplified as follows:

$$\begin{aligned} y_j(x) = \log\left(P(x|C_j)P(C_j)\right) &= \log\left(\frac{P(C_j)}{(2\pi)^{\frac{d}{2}} n_j h^d} \cdot \sum_{C_j} g(x)\right) \\ &= \log\left(L \cdot \sum_{C_j} g(x)\right) \qquad \text{for } j = 1, \ldots, k\ , \end{aligned} \tag{14}$$

where L is a constant that is equal for all $j = 1, \ldots, k$. This holds since the ratio of prior probabilities and training samples, $\frac{P(C_j)}{n_j}$, is equal for all classes C_j if it is true that training samples and prior probabilities are obtained in the same proportions. Plugging Equation 14 into Equation 13 gives the final softmax function:

$$\begin{aligned} P(C_j|x) &= \frac{\exp\left(\log\left(L \cdot \sum_{C_j} g(x)\right)\right)}{\sum_{i=1}^{k} \exp\left(\log\left(L \cdot \sum_{C_i} g(x)\right)\right)} = \frac{\exp\left(\log\left(L\right)\right) \exp\left(\log\left(\sum_{C_j} g(x)\right)\right)}{\sum_{i=1}^{k} \exp\left(\log\left(L\right)\right) \exp\left(\log\left(\sum_{C_i} g(x)\right)\right)} \\ &= \frac{\exp\left(\log\left(\sum_{C_j} g(x)\right)\right)}{\sum_{i=1}^{k} \exp\left(\log\left(\sum_{C_i} g(x)\right)\right)} \qquad \text{for } j = 1, \ldots, k. \end{aligned} \tag{15}$$

Hence, to compute the posterior probabilities, the softmax function of Equation 15 is used where the summation neuron output corresponding to a certain class C_j is basically divided by the sum over all summation neuron outputs from the PNN. After calculating these posterior probabilities for a binary or multi-class classification problem, they will serve as input for the quantile regression illustrated in a separate section below (see Section 2.6). However, the key differences between an ANN and PNN will be outlined first.

2.5 Differences ANN and PNN

ANNs and PNNs are two types of neural networks that mainly differ in how they are trained and how they model uncertainty. The training phase of an ANN is a crucial component that involves the iterative optimization of the weights to minimize a loss function, such as the mean squared error (MSE), that measures the differences between the predicted and actual output values. This iterative optimization process continues until the network converges to a set of weights that produce accurate predictions on the training data. The resulting network can then classify new input data based on the learned mapping between the training input and output values (Haykin, 2009). In contrast, the training phase of a PNN involves the estimation of class-conditional density functions using the Parzen window estimator (see Section 2.2). Once all the class-conditional densities for a specific input pattern are estimated, the PNN can be used to classify this input pattern based on these densities and Bayes' theorem (Specht, 1990). Therefore, unlike ANNs, PNNs do not require iterative adjustment of weights during the training phase.

The difference in the training phase between ANNs and PNNs reflects their different approaches to classification. ANNs learn a mapping from input variables to output variables, while PNNs use a non-parametric approach to estimate PDFs representing the distribution of the data within each class. Hence, PNNs have multiple advantages over other neural networks for some classification tasks. They require relatively small training data and can have a fast classification speed with high accuracy. On the other hand, PNNs can be computationally expensive for large data sets, and the choice of kernel function and bandwidth can affect the network's performance.

Another key distinction between ANNs and PNNs is how they model the uncertainty associated with classification. ANNs provide point estimates of class probabilities, while PNNs provide PDFs representing the uncertainty associated with the estimates. Consequently, PNNs can be more effective than ANNs when dealing with noisy or incomplete data, as they can account for uncertainty in the classification process (Bishop et al., 1995). This makes PNNs more suitable for applications where uncertainty is an important factor, such as any kind of risk assessment.

2.6 Quantile Regression

Quantile regression is a robust statistical method that extends traditional regression analysis by estimating the conditional quantiles of a response variable given one or more predictor variables. Unlike ordinary least squares, which estimates the conditional mean of the dependent variable, quantile regression focuses on estimating the response variable's conditional median (or any other quantile) across different values of the features (Koenker & Hallock, 2001). This methodology is particularly useful when investigating asymmetric relationships, prediction intervals, or specific quantiles of interest. Hence, it has been successfully used in various fields, such as economics

(Fitzenberger et al., 2001), finance (Baur et al., 2012), and environmental science (Cade & Noon, 2003), to analyze the relationships between variables and predict future outcomes.

Normally, when using a certain model to make a prediction, it is hard to see which predictor variables mainly drove the model to eventually make this prediction. Moreover, the model will always provide a prediction even though it might be uncertain about what to predict. Therefore, this thesis aims to show how quantile regression can be applied to provide more insight into a model's (in this case, a PNN's) decision-making process and how confident it is about its predictions. Performing a standard regression analysis on the posterior probabilities outputted by the PNN is insufficient since it will only result in a single-point estimate across the different values of the predictor variables. Instead, to be able to get more insight, quantile regression is performed on the probabilities of the PNN. This will result in predicting a range of values with a certain amount of confidence. Moreover, quantile regression offers another advantage as it provides a more comprehensive picture of the relationship between variables compared to traditional regression analysis (Koenker & Hallock, 2001).

To derive the quantile regression model, it is best to start with the traditional linear regression model, which is based on the following equation:

$$\hat{y}_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip} \qquad \text{for } i = 1, \ldots, n. \tag{16}$$

Here n is the number of samples, p is the number of feature variables, $\hat{y}_i$ is the predicted value of the i-th response variable, x_{ip} is the value of the p-th feature of sample i and β_p is the corresponding coefficient we wish to estimate. To find the optimal β-coefficients, the sum of squared errors is minimized with respect to β:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \Big(y_i - (\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}) \Big)^2, \tag{17}$$

where $\hat{\beta}$ is the p-dimensional vector containing all coefficient estimates and y_i is the actual value of the i-th response variable (Montgomery et al., 2021).

The basic idea in quantile regression is to estimate the conditional quantile function by minimizing a loss function using a structure similar to linear regression. The formula for this conditional quantile is slightly different from Equation 16:

$$\hat{Q}_{y_i}(\tau) = \beta_0(\tau) + \beta_1(\tau) x_{i1} + \ldots + \beta_p(\tau) x_{ip} \qquad \text{for } i = 1, \ldots, n. \tag{18}$$

Here τ is the quantile level and $\hat{Q}_Y(\tau)$ is the τ-th conditional quantile. Furthermore, all β coefficients depend on τ now. Estimating these coefficients is done by replacing the square in Equation 17 by the asymmetric absolute loss function $\rho_\tau(u)$ and implementing the expression for the conditional quantile (see Equation 18). This gives the following minimization problem (Koenker & Bassett Jr, 1978):

$$\hat{\beta}(\tau) = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \rho_\tau\Big(y_i - \hat{Q}_{y_i}\Big) = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \rho_\tau\Big(y_i - (\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip})\Big), \tag{19}$$

where $\hat{\beta}(\tau)$ is the p-dimensional vector containing all coefficient estimates for quantile level τ. Moreover, the asymmetric absolute loss function $\rho_\tau(u)$, also called the check function, is defined as:

$$\rho_\tau(u) = (\tau - I(u < 0))u \tag{20}$$

where $I(\cdot)$ is the indicator function and τ is again the quantile level (Koenker & Hallock, 2001). The check function is robust to outliers and allows for different slopes for different quantile levels (see Figure 3). Hence, depending on the overall sign of the error u and the quantile level, this loss function gives asymmetric weights to the individual errors. For example, if we are interested in the 10^{th} quantile, negative errors will get a weight of 0.9, while positive errors will receive a weight of 0.1. This implies that positive errors are preferred over negative ones resulting in the 10^{th} quantile being lower than the median quantile. Note that if $\tau = 0.5$, the weights given to the errors are symmetric. Now, $\rho_\tau(u)$ in Equation 19 can be replaced by the absolute value function since, in this case, they are proportional. Therefore, solving this median quantile regression minimization problem is the same as linear regression by least absolute deviations (Pollard, 1991).

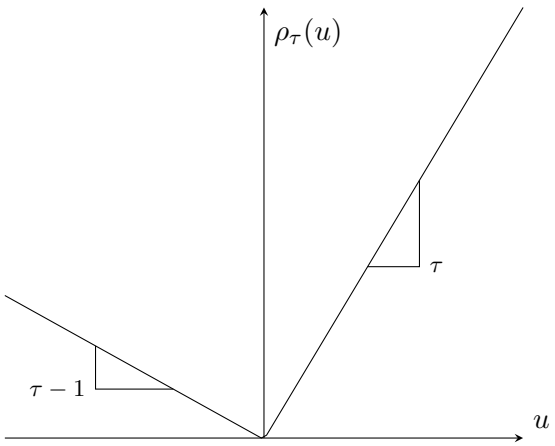


Figure 3: Quantile regression asymmetric absolute loss function, $\rho_\tau(u)$.

2.7 Deep Quantile Regression Neural Network

A DQRNN is a variant of an ANN (see Section 2.1) specifically designed for estimating the quantiles of a target variable. It extends the traditional neural network architecture by incorporating quantile-specific loss functions and enables direct modeling of the conditional quantile function (Cannon, 2011). For this thesis, a DQRNN is used to estimate different quantiles of the posterior probabilities estimated by the PNN.

The architecture of a DQRNN typically consists of multiple hidden layers with nonlinear activation functions, similar to a standard feedforward neural network (see Section 2.1). Furthermore, if a standard neural network is used to fit a linear regression on a target variable, the output layer generally consists of one node corresponding to a single point estimate. However, the output layer of a DQRNN is different, as it outputs multiple nodes representing the different quantiles of the target variable. Hence, the number of nodes in the output layer of the DQRNN will equal the number of quantiles we wish to estimate.

A specific loss function, known as the quantile loss or asymmetric absolute loss function (Koenker & Hallock, 2001), is used to train a DQRNN. This function was already defined above in Equation 20. It measures the discrepancy between the predicted and actual values of the target variable. Depending on the desired quantile level, the asymmetric absolute loss function penalizes underestimation and overestimation differently (see Section 2.6). During training, the parameters of the DQRNN are updated by minimizing the overall quantile loss across the training data set (see Equation 19). Where, for this study, the gradient-based optimization algorithm Adam (Kingma & Ba, 2014) is used to solve this minimization problem. This process adjusts the weights and biases of the DQRNN to improve the accuracy of the estimated quantiles.

DQRNNs have been effectively utilized in various domains, including finance (Taylor, 2000), environmental sciences (Cannon, 2018), and the energy industry (Zhang et al., 2018), where estimating quantiles is crucial for risk assessment and decision-making.

3 Data

This research uses the Titanic data set, downloaded from Kaggle. It contains the passengers' characteristics and which passengers survived the Titanic shipwreck. On Kaggle it is used for a machine learning competition called, Machine Learning from Disaster, with over 16,000 participants. This makes it a well-known data set in the machine learning industry. Furthermore, the data set is easy to understand which helps to interpret the results of this thesis. Once downloaded it has 12 variables, but not all of them are useful. Hence, data cleaning is performed, only keeping the variables that seem to contain important information about the passengers. As a result, the variables passengerID, name, ticket number, and cabin number are removed. Of course, the ticket or cabin number could have indicated on which part of the Titanic a passenger was staying. For example, some passengers might have had a cabin closer to the lifeboats increasing their survival chances. However, the aim of this paper is to demonstrate how fitting a quantile regression on the output of a machine learning model shows how certain this model is about its predictions. Therefore, keeping these variables might increase the accuracy of the model, but since they would compromise the interpretability of the results from the proposed method, they are removed from the data set.

After dropping those four variables, one more data cleaning step is performed before the

data set is split up into a train and test set. Since the variable age contains a significant amount of unknown values, all data for passengers with an unknown age are removed. Again, this will probably hurt the accuracy of the PNN, but that is not the main interest of this research. Various descriptive statistics of the data set before and after data cleaning are displayed in Table 1. When comparing the descriptive statistics of the data set before and after cleaning they do not seem to change much. However, the percentage of passengers that embarked at Queenstown after data cleaning seems to have dropped significantly from 0.09 to 0.04. This implies that a relatively large number of passengers that boarded the Titanic at Queenstown have an unknown age. Furthermore, we can see from the table that the age is missing for 177 passengers, resulting in a final data set of 714 passengers with one dependent and seven feature variables.

Table 1: Descriptive statistics of the full data set, the data set after cleaning, the train data, and the test data.

	Total data set	After cleaning	Train set	Test set
Number of observations	891	714	571	143
Survival rate	0.38	0.41	0.41	0.40
Average age	29.70	29.70	29.93	28.78
Average passenger fare	32.20	34.69	34.73	34.56
Average ticket class	2.31	2.24	2.23	2.27
Average number of parents or children aboard	0.38	0.43	0.43	0.45
Average number of siblings or spouses aboard	0.52	0.51	0.49	0.60
Percentage female	0.35	0.37	0.37	0.34
Percentage male	0.65	0.63	0.63	0.66
Percentage embarked at Cherbourg	0.19	0.18	0.19	0.15
Percentage embarked at Southampton	0.72	0.78	0.77	0.82
Percentage embarked at Queenstown	0.09	0.04	0.04	0.03

The dependent variable is survived and it indicates whether a passenger survived the Titanic shipwreck or not. The first two feature variables, age and passenger fare, are both continuous and they range between 0.42-80.00 and 0.00-512.33, respectively. To reduce the scale of these variables they are both standardized before the PNN is applied. Then, we have the variable ticket class showing if a passenger had an upper (1), middle (2), or lower (3) class ticket. The next two independent variables are the number of parents or children and the number of siblings

or spouses on board the ship. These are both categorical variables that can attain values from 0-5 and 0-6, respectively. The low averages of these two variables (see Table 1) imply that the majority of the passengers fall into the lower categories. This is confirmed by Figure 13 in Appendix A which clearly shows that passengers with a value of 3 or higher for one of these variables are not common. Feature variable number six is gender, indicating whether a passenger is male or female. To make this input variable usable for the PNN it is made binary where male equals 0 and female equals 1. The last feature variable is the port of embarkation showing if a passenger boarded the Titanic at Cherbourg (C), Southampton (S), or Queenstown (Q). To make this variable numerical one-hot encoding is used (Hancock & Khoshgoftaar, 2020). This is where a new binary variable is added for each unique value of the initial variable and afterward, the initial variable is removed. An overview of the eight variables used for this research can be found in Table 2 in Appendix A.

After all variables are prepared the data set is separated into a train and test set following an 80/20 split. The differences between the train and test set can be viewed in Table 1. From this table, we can see that the average number of siblings or spouses aboard is significantly higher for the test set and the average age in the test set is over a year lower. Furthermore, a larger fraction of passengers from the test set seem to have boarded at Southampton instead of Cherbourg. However, overall the descriptive statistics of both sets look comparable. Hence, in the next section, these two data sets will be used to gather the results.

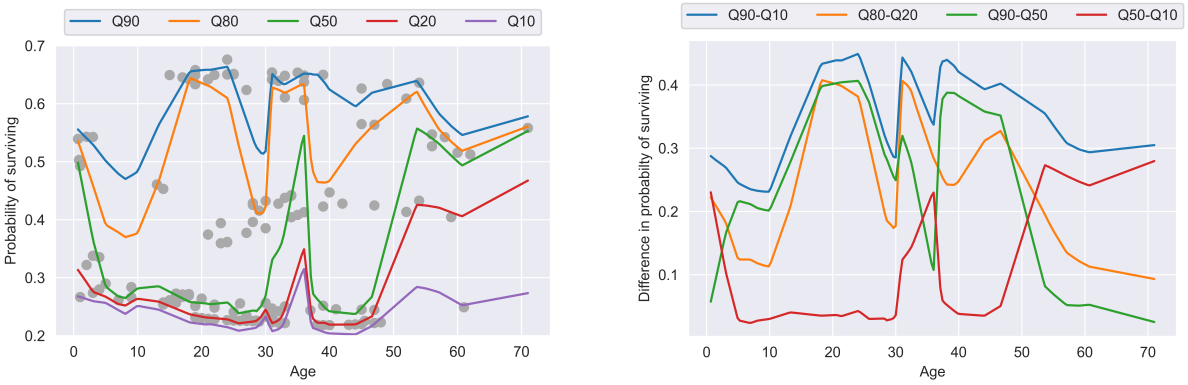
4 Results

After the Titanic data set is cleaned (see Section 3) the PNN is implemented. To determine the optimal bandwidth parameter h for the PNN, 10-fold cross-validation is used on the train set. This results in a bandwidth parameter of 0.55. Subsequently, using this bandwidth parameter for the PNN results in an accuracy of 0.71 on the test set. Note that the aim of this thesis is not to display a machine-learning model that can compete with the best-performing models in terms of accuracy. Instead, it will show how fitting a quantile regression on the probabilities outputted by a machine learning model can provide insights into its decision-making process. Moreover, it can be retrieved with what certainty the model made its predictions. This also explains why a PNN was chosen as the model for this research since it is designed to provide probabilities as its output. The predicted probabilities for the test set obtained by the PNN and the quantile regression curves fitted for all feature variables are shown in the figures below. This section will discuss and interpret these figures in depth.

Before analyzing and discussing the results it is important to explain what the figures display exactly. Each of the seven feature variables of the Titanic data set will have two corresponding figures. The left figure shows the different quantile regression curves where the 90^{th}, 80^{th}, 50^{th}, 20^{th}, and 10^{th} quantile are presented. These quantile regression curves are fitted on the

probabilities of surviving computed by the PNN which are also shown as a scatter plot in the same figure. The right figure shows the size of the different prediction intervals for different values of the feature variable. The size of these intervals is derived from the distance between the corresponding quantile regression lines in the left figure. For example, a line or bar with "Q90-Q10" in the legend shows the prediction interval size between the 90^{th} and 10^{th} quantile regression curves.

Figure 4 shows the two plots for the first feature variable age. First, focusing on the left figure, the scatter plot shows that the probabilities of surviving seem to be pretty spread out for all values of age. Something similar is observed when looking at the quantile regression lines. However, the lines seem to be closer together for ages around 0-13 and 55-71. This implies that the PNN is more certain about its estimated probability of surviving for passengers aged below 13 or between 55 and 71. Especially for passengers younger than 13, there seems to be a very high chance of the model estimating a probability of surviving below 0.5. This suggests that the model believes children below 13 years old have a low chance of surviving the sinking of the Titanic.



(a) Quantile regression curves

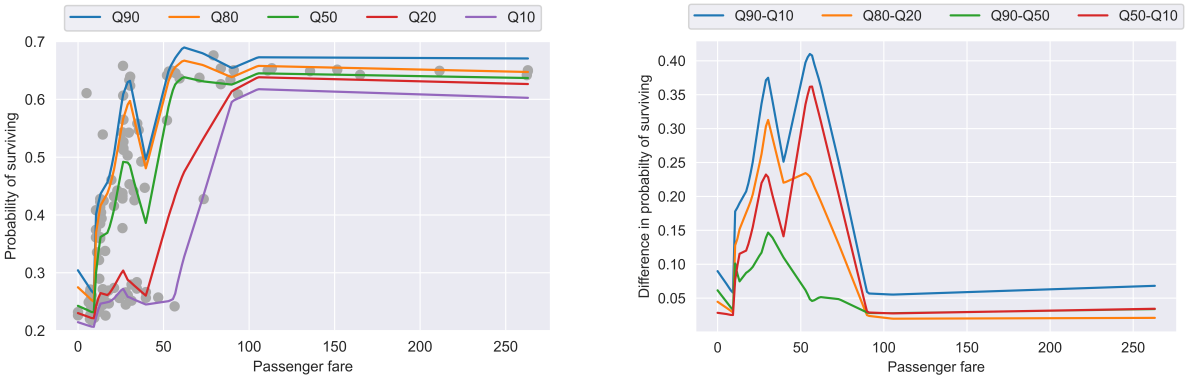
(b) Line graph of the prediction interval size

Figure 4: The quantile regression curves and their corresponding prediction interval size for different values of age.

This is further confirmed by studying the right figure (see Figure 4b). The blue and orange lines in this figure show small prediction intervals for these ranges of age. Of course, it needs to be noted that the number of observations for passengers with an age inside these two ranges seems to be relatively low, which could explain this behavior of the quantile curves. Looking further it can also be observed that the quantile curves get closer to each other around the age ranges 25-30 and 33-40 (see blue and orange lines in Figure 4b), although the size of the prediction intervals for these ranges is still larger compared to the 0-13 and 55-71 age ranges mentioned previously. However, it does again imply that for a passenger with an age between 25-30 or 33-40, the model seems to be more certain about its estimated probability of surviving.

Furthermore, when looking at the right figure the blue, orange, and green lines seem to follow each other closely while the red line moves differently. This shows that the scatter plot is on average denser at the bottom for lower probabilities of surviving since most of the time the red line is below the green line. Although, for some ages the red line seems to increase and cross the green line, indicating that for those ages the probabilities estimated by the PNN are denser at the top.

The figures of the next feature variable, the passenger fare, show different results (see Figure 5). Looking at the quantile regressions curves in the left figure we can observe three interesting ranges for the fare where the quantile lines seem to be closer to one another. First, for passengers that paid a fare between 0 and 10, the model seems to be very certain that the probability of surviving lies somewhere between 0.2 and 0.3. This is also displayed by the small prediction interval sizes in the figure on the right side. Once the fare becomes larger than 10 we can see an immediate increase of the upper two quantile curves, but the size of the 80% prediction interval (blue line in Figure 5b) stays smaller than 0.3 until a fare of around 25 is reached. This implies that the model is confident that passengers that paid a low fare have a lower chance of surviving.



(a) Quantile regression curves

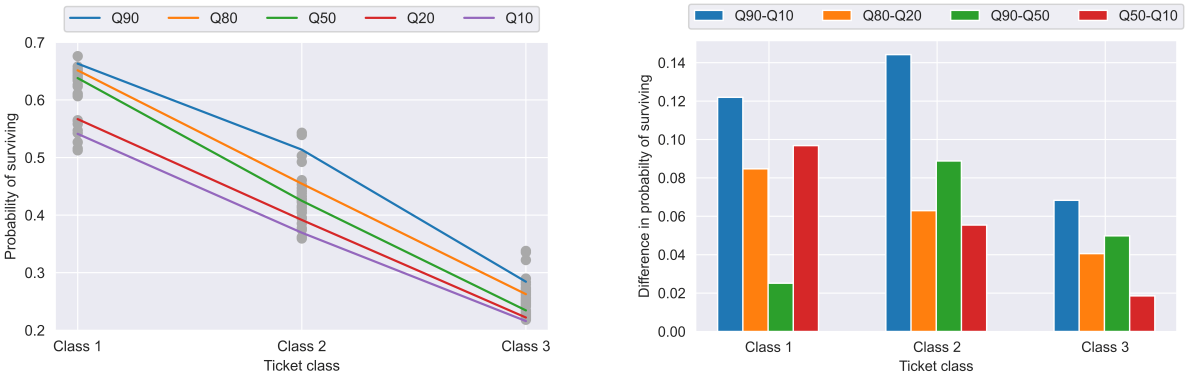
(b) Line graph of the prediction interval size

Figure 5: The quantile regression curves and their corresponding prediction interval size for different values of the passenger fare.

Moving further to the right, the 90^{th} and 80^{th} quantile curves go down somewhat for passenger fares of approximately 30 to 40 and start to go back up from 40 to 50. Both figures show a downward spike in this fare range (see the blue and orange lines in Figures 5a and 5b), indicating the PNN is more certain about its estimated probabilities for passengers that paid a fare between 30 and 50. The last interesting observation is regarding fares of around 90 and above. Here the quantile curves get really close to each other again, which is also confirmed by the blue and orange lines in the right figure being very low. This shows that the model is very confident that passengers who paid a high fare have a high probability of surviving the accident. Again, similar to age, note that this behavior of the quantiles might be caused by the lack of

passengers that paid a fare of 90 or more. The green and red lines in the right figure seem to move pretty similarly, apart from fares between approximately 40 and 90. For fares within this range, the red line is clearly above the green line which implies that the data is denser at the top.

The third independent variable from the Titanic data set is the ticket class and the corresponding results are perhaps the most interesting out of all seven features. If we look at the scatter plot combined with quantile regression lines it is interesting to see that the dots, and therefore also the quantile lines, are very close together for each individual ticket class (see Figure 6a). By closer inspection, for class 1 it seems very likely that the model predicts a probability of surviving above 0.5. On the contrary, for class 3 it looks as if the PNN always predicts a probability below 0.5, where most of the estimated probabilities are between 0.2 and 0.3. This is similar, but the other way around, to what we saw earlier for the passenger fare. Low estimated probabilities of surviving for low fares and high probabilities for high fares. This would suggest that the passenger fare and ticket class are negatively correlated which is indeed confirmed by Figure 14 in Appendix B and the corresponding Pearson correlation coefficient of -0.58. Intuitively, this also makes sense since a high passenger fare is likely to correspond to an upper-class or class 1 ticket and vice versa. Lastly, for class 2 we see some probabilities higher and some lower than 0.5. However, in general, it seems as if the model will output a probability below 0.5 for passengers with middle-class tickets.



(a) Quantile regression curves

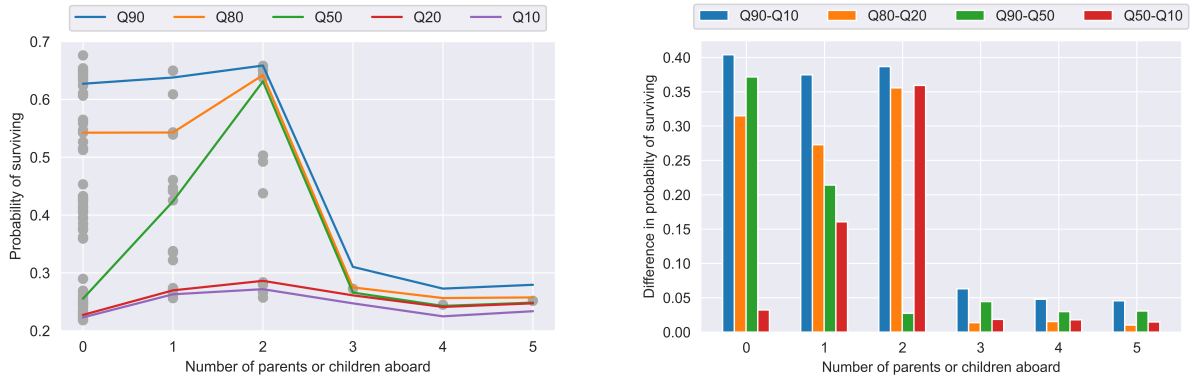
(b) Bar chart of the prediction interval size

Figure 6: The quantile regression curves and their corresponding prediction interval size for different values of the ticket class.

The fact that the quantile regression lines are close together for all three classes shows that the PNN is very confident about the probabilities that it computed, especially for the lower-class or class 3 tickets. This is confirmed by the right-side figure which indeed shows small prediction interval sizes for all three ticket classes where the size of the prediction intervals for class 3 are the smallest (see the blue and orange bars in Figure 6b). Furthermore, when comparing the

y-axis of this figure to the corresponding figures of the other features, it looks like the PNN largely based its estimated probabilities of surviving on the ticket class variable. At some point, all other figures seem to have an 80% prediction interval size of around 0.4, but the maximum distance between the 90^{th} and 10^{th} quantile for the ticket class variable is only around 0.14. Lastly, for class 1 the red bar is higher than the green bar which indicates that for passengers with an upper-class ticket, the probabilities estimated by the PNN are denser at the top whereas the opposite is true for ticket classes 2 and 3.

The next variable that will be analyzed is the number of parents or children aboard the Titanic. Looking at the values 0, 1, and 2 first, we can see that the quantile regression curves in Figure 7a are far apart. The sizes of the 80% and 60% prediction intervals lie around 0.4 and 0.3, respectively, for these three values of this variable (see the blue and orange bars in Figure 7b). This implies that, based on this variable, the model is not confident about what probability of surviving it should assign to a passenger with 0, 1, or 2 parents or children on board. In other words, we can see that the PNN did not base its estimated probabilities for these passengers on this variable. Furthermore, when examining the green and red bars for the values 0, 1, and 2 in Figure 7b they imply that for 0 the data is denser at the bottom (green bigger than red), for 1 the data is pretty evenly distributed and for 2 the data seems to be denser at the top.



(a) Quantile regression curves

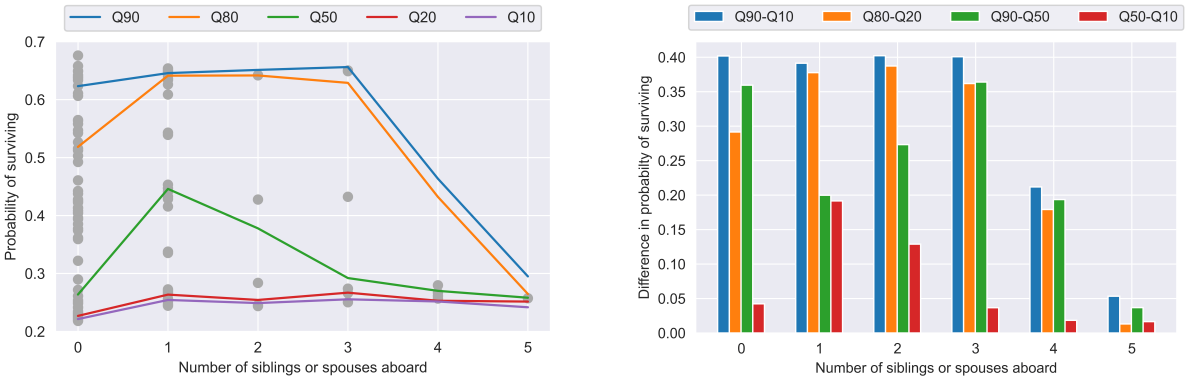
(b) Bar chart of the prediction interval size

Figure 7: The quantile regression curves and their corresponding prediction interval size for different values of the number of parents or children aboard.

For the remaining three values 3, 4, and 5 we observe the opposite. Here the quantile regression curves are close together and the sizes of the corresponding prediction intervals are small if we look at their blue and orange bars in the right-side figure. This again indicates that for passengers with 3, 4, or 5 parents or children on board, the PNN seems certain about the probabilities that it calculated. Moreover, the model seems to believe that these passengers have a low chance of surviving. Note however that the amount of data points with a value of 3, 4, or 5 seems to be limited (see Figure 13 in Appendix A) which could explain why the quantile lines

are close to each other. This is something that was also observed earlier for other variables.

If we look at the figures of the next variable, which is the number of siblings or spouses aboard, they look similar to the figures of the previous variable we discussed. Again, for certain values of the variable, the quantile regression lines seem to be far apart. Although, this time this is the case for the first four values: 0, 1, 2, and 3. Of course, we also observe large corresponding prediction interval sizes for these values (see blue and orange bars in Figure 8b). Looking at the distribution of the data, for the values 0, 2, and 3 the probabilities seem to be denser at the bottom (green bigger than red) while for value 1 the data seems to be pretty evenly spread. For values 4 and 5 the quantile lines get closer together, suggesting that for passengers with 4 or 5 siblings or spouses, the PNN seems confident that they will have a low probability of surviving. However, again it has to be noted that the data is very sparse for these values (see Figure 13 in Appendix A) which means we have to be careful when drawing any conclusions.



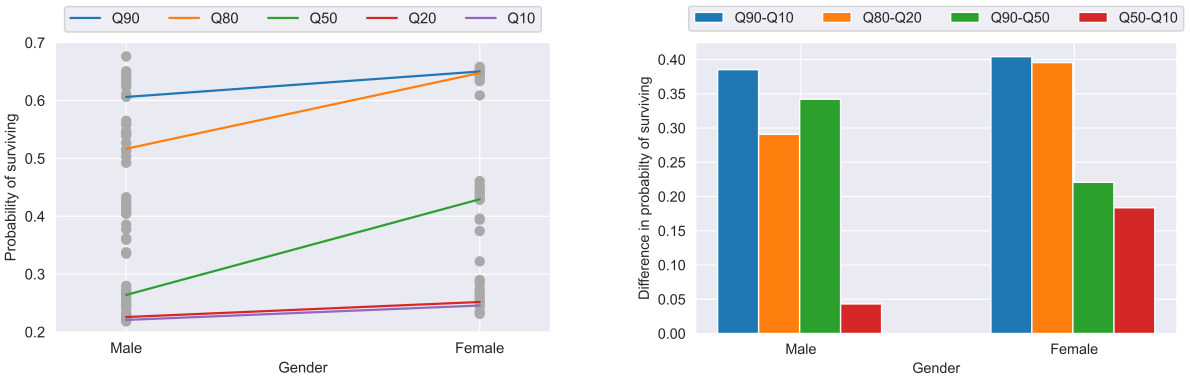
(a) Quantile regression curves

(b) Bar chart of the prediction interval size

Figure 8: The quantile regression curves and their corresponding prediction interval size for different values of the number of siblings or spouses aboard.

The sixth variable of the data set that will be discussed is gender. For this variable, there are not that many interesting observations to be mentioned. This is due to the fact that for both males and females, the quantile lines are far apart (see Figure 9a). They both have blue bars close to 0.4 (see Figure 9b) which implies that the probabilities estimated by the PNN were not really influenced by the gender of a passenger. However, what this quantile regression analysis does show is that the sizes of the 80% and 60% prediction intervals are slightly smaller for males if we compare the blue and orange bars. This suggests that the probabilities predicted by the model are most of the time a little closer together for males implying that the model is more certain about these predictions. This would be hard to spot if we only had the scatter plot without the quantile lines since the total spread of the probabilities for males is actually larger compared to females. However, because the estimated probabilities for males are very dense at the bottom (green is larger than red in Figure 9b) while the probabilities for females are more

evenly distributed the prediction interval sizes turn out to be smaller for males. Moreover, it seems to be more likely for the PNN to predict a probability of surviving below 0.5 for males than it is for females.

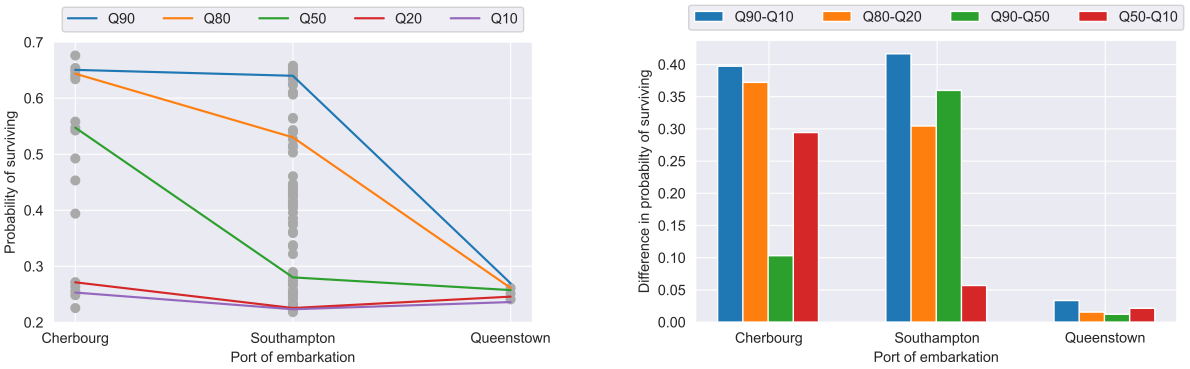


(a) Quantile regression curves

(b) Bar chart of the prediction interval size

Figure 9: The quantile regression curves and their corresponding prediction interval size for different values of gender.

The last variable that will be examined is the port of embarkation (see Figure 10). The quantile curves for this variable are far apart for Cherbourg and Southampton, but close to each other for Queenstown. This is also confirmed by the size of the blue and orange bars in Figure 10b. Again, it seems as if the PNN did not really let the port of embarkation play a role when estimating the probability of surviving for passengers that got on board the Titanic in Cherbourg or Southampton. However, it does look as if there is a higher chance of the model predicting a higher probability of survival for a passenger that got on board in Cherbourg compared to a passenger that boarded the ship in Southampton. This can be seen from the green and red bars in Figure 10b, where for Cherbourgh the data seems to be denser at the top (red larger than green) while for Southampton the opposite is observed. For Queenstown the sizes of the prediction intervals are small, indicating that the model is certain that passengers who boarded in Queenstown have a low probability of surviving. However, it should be noted that for this port of embarkation, the data is again sparse.



(a) Quantile regression curves

(b) Bar chart of the prediction interval size

Figure 10: The quantile regression curves and their corresponding prediction interval size for different values of the port of embarkation.

To summarise the above observations a bar chart is made that shows the average prediction interval sizes for each individual feature variable (see Figure 11). To get this graph, for each variable, the means of the 80% and 60% prediction interval sizes (Q90-Q10 and Q80-Q20) are calculated by taking the average of the blue and orange lines or bars from the b figures shown above. This results in the figure below where in this case a large blue or orange bar indicates a large average 80% or 60% prediction interval size for that specific feature variable.

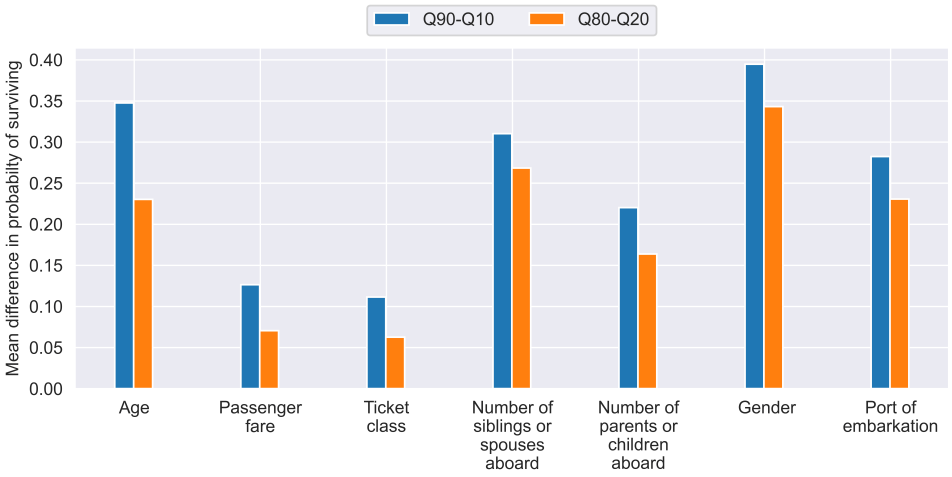


Figure 11: Bar chart of the average prediction interval size for each feature variable

This plot shows that we get the lowest average prediction interval sizes for ticket class and passenger fare. This supports the conclusions drawn earlier where it was already observed that the PNN seemed to be basing its estimated probabilities of surviving mainly on the ticket class and to a lesser extent on passenger fare. Other variables, such as the number of parents or children aboard and the port of embarkation also seem to play a slight role in the probabilities computed by the PNN since they have the third and fourth lowest average prediction interval

sizes. For age, we also observed some interesting ranges where the quantile lines were closer together, but from this chart, it does not seem as if age played a significant role in the computation of the probabilities by the PNN. Although, especially its average 60% prediction interval size is relatively low.

The issue with the bar chart in Figure 11 is that it does not take the distribution of the estimated probabilities into account. If at some point the quantile regression curves were close together for a feature variable, but this was only based on a few probabilities in that range, the size of the prediction intervals at that point would have the same weight in the average as the prediction interval sizes which are based on a large number of probabilities. Therefore, it seems more reasonable to calculate a weighted mean of the prediction interval sizes for each feature. This means that now the size of a prediction interval which is based on a lower number of data points also has a lower weight in the calculated average. Figure 12 shows the bar chart with the weighted average prediction interval size.

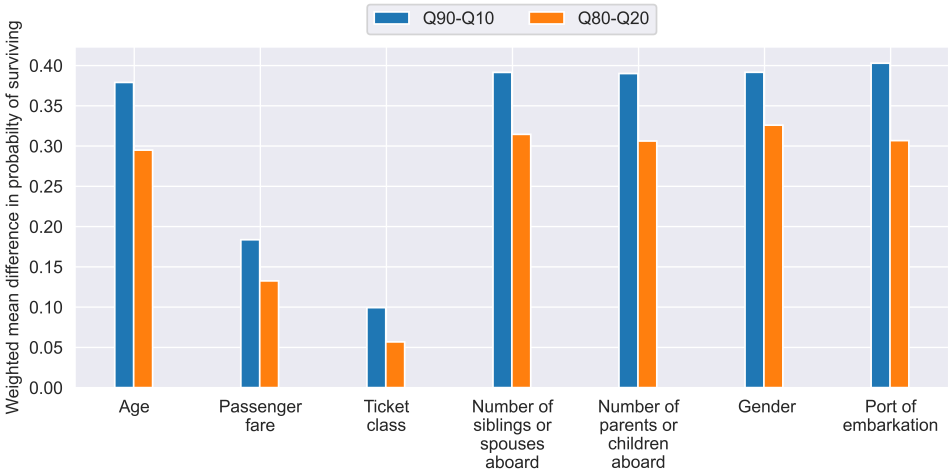


Figure 12: Bar chart of the weighted average prediction interval size for each feature variable

Looking at this figure a few interesting changes are noticed. Again, the weighted average prediction interval sizes for ticket class and passenger fare are still the smallest. However, both interval sizes went slightly down for ticket class while they got somewhat larger for the passenger fare. Especially, for passenger fare this was to be expected since the area of a fare of 90 or higher, where the quantile lines are close to each other, seemed to contain a relatively low number of probabilities (see Figure 5b). Therefore, this range for the passenger fare with its corresponding small prediction interval sizes has a lower weight in the average this time. Furthermore, the remaining variables all look very similar now. They all have a weighted average 80% and 60% prediction interval of around 0.4 and 0.3, respectively. Although, in this figure, age is now the variable with the third lowest weighted average prediction interval sizes. This seems to resemble the observations made above more closely since we did notice some interesting ranges for age (especially the age ranges 0-13 and 55-71) where the model seemed more certain about

what probabilities it should give to passengers with an age inside these ranges. The weighted average prediction interval sizes for the number of parents or children aboard and the port of embarkation are significantly larger than their unweighted averages. Similar to the passenger fare, this can be explained by the sparsity of the data for values that had a small prediction interval size. All in all, it seems as if the PNN mainly looks at the ticket class and the passenger fare of the passengers to compute the corresponding probabilities of surviving where, on top of that, these two variables are also negatively correlated. The remaining five variables on the other hand seem to be somewhat disregarded. This suggests that the current model might not be fully exploiting the data and improving this model or using a more sophisticated model would probably result in a higher accuracy.

5 Conclusion and Discussion

In this thesis, we focused on the issue posed by the rising complexity of machine learning models, which pursue higher accuracy at the expense of interpretability. To address this problem and gain deeper insights into black box models, we proposed a new method. The objective was to investigate whether quantile regression analysis could assist in identifying the decision-making process of a machine learning model. Additionally, the distance between the quantile regression curves at specific values of a feature variable could provide useful insights into the model's level of certainty when making its predictions. For illustrative purposes and the capability to produce probabilities, we selected a PNN as the machine learning model for evaluating the performance of this quantile regression analysis. Furthermore, while the PNN is intuitive, interpreting the calculations conducted by the individual neurons and explaining why it made certain predictions can still be challenging.

The performed quantile regression analysis yielded some intriguing conclusions. First of all, the quantile curves effectively demonstrated which specific feature variables were utilized by the model and which ones were disregarded. The analysis revealed that the PNN mainly relied on the variables ticket class and passenger fare for its predictions, providing deeper insights into the PNN's decision-making process. Furthermore, a smaller prediction interval size associated with certain feature variable values indicated a more stable output by the PNN, suggesting a stronger influence of those feature variable values on the model's predictions. Therefore, the method's ability to provide prediction interval sizes allowed for visualizing the prediction certainty of the PNN for different values of the feature variables. The advantage of using a relatively small and comprehensible data set for demonstrating the method was clear when examining the ticket class variable, where the stable pattern of the predictions was already easily observable from the scatter plot (see Figure 6a). The confirmation of this observation by the proposed analysis, showing the smallest average distance between the quantile curves for ticket class, further supports the validity of the method. Moreover, the analysis revealed interesting

ranges for the variables age and passenger fare which are harder to spot purely based on the scatter plots (see Figures 4a and 5a). Currently, the method used to measure the overall impact of a variable is the unweighted and weighted average prediction interval size of the different feature variables. From our observations, the weighted average prediction interval size appears to be a fairer measuring method when compared to the unweighted distance. For the relatively simple PNN utilized in this research, the (weighted) average distance seems sufficient. However, as the machine learning model becomes more sophisticated and incorporates additional variables for predictions, it may exhibit stable outputs for certain feature variable values while showing instability for other values of the same variable, as we already observed for age and passenger fare in this study. This variability could even out the (weighted) average distance, highlighting the importance of also examining individual feature variable figures. In the case of more complex models, exploring alternative measuring methods to quantify the impact of variables on the model's predictions is another viable option.

Regarding future research, several intriguing proposals arise. Firstly, keeping it closely related to this study, investigating the outcomes of the quantile regression analysis after removing the current most influential feature variable, ticket class, would be insightful. Will the passenger fare take on the role of the ticket class, or will an unused variable suddenly have the most significant impact? Additionally, incorporating the variables that were excluded from this research, such as ticket and cabin numbers, and re-running the analysis could offer valuable insights. As mentioned earlier, the performed quantile regression analysis reveals that the current form of the PNN merely utilizes the variables ticket class and passenger fare, which are also negatively correlated. This could explain the relatively low accuracy of the PNN. Therefore, it would be interesting to try to improve the model and perform the same analysis again to observe whether more variables are incorporated by the model this time.

Another interesting area for further research could be exploring different data sets. Analyzing how the quantile regression analysis performs on a larger data set or a high-stake finance classification data set could be insightful. Additionally, investigating the analysis for a multiple-output classification problem would be enlightening. In this case, identifying the variables that have the most significant impact on the PNN's predictions might not be straightforward by just examining the scatter plot. Hence, it would be intriguing to observe whether the distance between the quantile curves could still provide us with valuable insights regarding the influential feature variables. Furthermore, it would be valuable to examine how the method performs when replacing the PNN with a less interpretable and more sophisticated model, such as a DNN. Can the proposed approach still identify the most influential feature variables and determine the DNN's prediction certainty? Lastly, a comparison of the performance of the quantile regression analysis with existing methods, such as neural network visualizations and LIME (see Section 1), would be beneficial in understanding the strengths and limitations of each technique.

References

Babich, G. A., & Camps, O. I. (1996). Weighted parzen windows for pattern classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *18*(5), 567–570.

Bathaee, Y. (2017). The artificial intelligence black box and the failure of intent and causation. *Harv. JL & Tech.*, *31*, 889.

Baur, D. G., Dimpfl, T., & Jung, R. C. (2012). Stock return autocorrelations revisited: A quantile regression approach. *Journal of Empirical Finance*, *19*(2), 254–265.

Bishop, C. M., et al. (1995). *Neural networks for pattern recognition*. Oxford university press.

Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern recognition and machine learning* (Vol. 4). Springer.

Cacoullos, T. (1964). *Estimation of a multivariate density* (tech. rep.). University of Minnesota.

Cade, B. S., & Noon, B. R. (2003). A gentle introduction to quantile regression for ecologists. *Frontiers in Ecology and the Environment*, *1*(8), 412–420.

Cannon, A. J. (2011). Quantile regression neural networks: Implementation in r and application to precipitation downscaling. *Computers & geosciences*, *37*(9), 1277–1284.

Cannon, A. J. (2018). Non-crossing nonlinear regression quantiles by monotone composite quantile regression neural network, with application to rainfall extremes. *Stochastic environmental research and risk assessment*, *32*, 3207–3225.

Castelvecchi, D. (2016). Can we open the black box of ai? *Nature News*, *538*(7623), 20.

CNBC. (2023). Google announces bard a.i. in response to chatgpt. https://www.cnbc.com/2023/02/06/google-announces-bard-ai-in-response-to-chatgpt.html

Efron, B., & Tibshirani, R. (1997). Improvements on cross-validation: The 632+ bootstrap method. *Journal of the American Statistical Association*, *92*(438), 548–560.

Epanechnikov, V. A. (1969). Non-parametric estimation of a multivariate probability density. *Theory of Probability & Its Applications*, *14*(1), 153–158.

Fitzenberger, B., Koenker, R., & Machado, J. A. (2001). *Economic applications of quantile regression*. Springer Science & Business Media.

Forman, G., & Scholz, M. (2010). Apples-to-apples in cross-validation studies: Pitfalls in classifier performance measurement. *Acm Sigkdd Explorations Newsletter*, *12*(1), 49–57.

Gao, G. (2010). A parzen-window-kernel-based cfar algorithm for ship detection in sar images. *IEEE Geoscience and Remote Sensing Letters*, *8*(3), 557–561.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian data analysis*. CRC press.

Georgiou, V., Pavlidis, N., Parsopoulos, K., Alevizos, P. D., & Vrahatis, M. (2004). Optimizing the performance of probabilistic neural networks in a bioinformatics task. *Proceedings of the EUNITE 2004 Conference*, 34–40.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, *51*(5), 1–42.

Han, J., & Moraga, C. (1995). The influence of the sigmoid function parameters on the speed of backpropagation learning. *From Natural to Artificial Neural Computation: International Workshop on Artificial Neural Networks Malaga-Torremolinos, Spain, June 7–9, 1995 Proceedings 3*, 195–201.

Hancock, J. T., & Khoshgoftaar, T. M. (2020). Survey on categorical data for neural networks. *Journal of Big Data*, *7*(1), 1–41.

Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (Vol. 2). Springer.

Haykin, S. (2009). *Neural networks and learning machines*. Pearson. https://books.google.nl/books?id=KCwWOAAACAAJ

Hirschauer, T. J., Adeli, H., & Buford, J. A. (2015). Computer-aided diagnosis of parkinson's disease using enhanced probabilistic neural network. *Journal of medical systems*, *39*, 1–12.

Isinkaye, F. O., Folajimi, Y. O., & Ojokoh, B. A. (2015). Recommendation systems: Principles, methods and evaluation. *Egyptian informatics journal*, *16*(3), 261–273.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Koenker, R., & Bassett Jr, G. (1978). Regression quantiles. *Econometrica: journal of the Econometric Society*, 33–50.

Koenker, R., & Hallock, K. F. (2001). Quantile regression. *Journal of economic perspectives*, *15*(4), 143–156.

Kohavi, R., et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *Ijcai*, *14*(2), 1137–1145.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436–444.

Lipton, Z. C. (2018). The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, *16*(3), 31–57.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, *30*.

Montgomery, D. C., Peck, E. A., & Vining, G. G. (2021). *Introduction to linear regression analysis*. John Wiley & Sons.

Morin, F., & Bengio, Y. (2005). Hierarchical probabilistic neural network language model. *International workshop on artificial intelligence and statistics*, 246–252.

Parzen, E. (1962). On estimation of a probability density function and mode. *The annals of mathematical statistics*, *33*(3), 1065–1076.

Pollard, D. (1991). Asymptotics for least absolute deviation regression estimators. *Econometric Theory*, *7*(2), 186–199.

Price, W. N. I. (2017). Regulating black-box medicine. *Mich. L. Rev.*, *116*, 421.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016a). "why should i trust you?" explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1135–1144.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016b). Model-agnostic interpretability of machine learning. *arXiv preprint arXiv:1606.05386*.

Rosenblatt, M. (1956). Remarks on some nonparametric estimates of a density function. *The annals of mathematical statistics*, 832–837.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, *1*(5), 206–215.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, *323*(6088), 533–536.

Silverman, B. W. (1986). *Density estimation for statistics and data analysis* (Vol. 26). CRC press.

Similarweb. (2023). Chat.openai.com. https://www.similarweb.com/website/chat.openai.com/#overview

Specht, D. F. (1990). Probabilistic neural networks. *Neural networks*, *3*(1), 109–118.

Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic attribution for deep networks. *International conference on machine learning*, 3319–3328.

Taylor, J. W. (2000). A quantile regression neural network approach to estimating the conditional density of multiperiod returns. *Journal of forecasting*, *19*(4), 299–311.

The Guardian. (2023a). Chatgpt reaches 100 million users two months after launch. https://www.theguardian.com/technology/2023/feb/02/chatgpt-100-million-users-open-ai-fastest-growing-app

The Guardian. (2023b). Microsoft confirms multibillion dollar investment in firm behind chatgpt. https://www.theguardian.com/technology/2023/jan/23/microsoft-confirms-multibillion-dollar-investment-in-firm-behind-chatgpt

Varuna Shree, N., & Kumar, T. (2018). Identification and classification of brain tumor mri images with feature extraction using dwt and probabilistic neural network. *Brain informatics*, *5*(1), 23–30.

Von Eschenbach, W. J. (2021). Transparency and the black box problem: Why we do not trust ai. *Philosophy & Technology*, *34*(4), 1607–1622.

Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., & Lipson, H. (2015). Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*.

Zednik, C. (2021). Solving the black box problem: A normative framework for explainable artificial intelligence. *Philosophy & technology*, *34*(2), 265–288.

Zhang, W., Quan, H., & Srinivasan, D. (2018). An improved quantile regression neural network for probabilistic load forecasting. *IEEE Transactions on Smart Grid*, *10*(4), 4425–4434.

Appendix

Appendix A

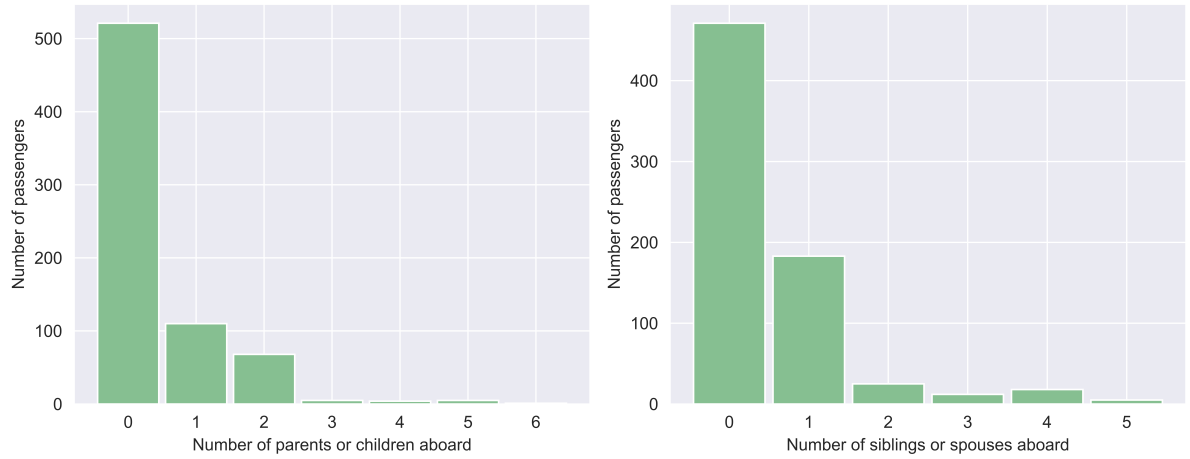


Figure 13: The distribution of the data after cleaning for the variables number of parents or children aboard and number of siblings or spouses aboard.

Table 2: Overview of all variables and their corresponding variable type, used for this research.

Variable	Description	Variable type
Survived	Whether or not the passenger survived the sinking of the Titanic, where 0 = no and 1 = yes	Categorical (Binary)
Age	The age of the passenger	Continuous
Passenger fare	The price the passenger paid for the ticket	Continuous
Ticket class	The class of the passenger's ticket, where 1 = upper class, 2 = middle class, and 3 = lower class	Categorical (Ordinal)
Number of parents or children aboard	The number of parents or children the passenger had aboard the Titanic, where a parent is either a mother or father and a child is either a daughter, son, stepdaughter or stepson	Categorical (Ordinal)
Number of siblings or spouses aboard	The number of siblings or spouses the passenger had aboard the Titanic, where a sibling is either a brother, sister, stepbrother or stepsister and a spouse is either a husband or wife	Categorical (Ordinal)
Gender	The gender of the passenger, where 0 = male and 1 = female	Categorical (Binary)
Port of embarkation	The port at which the passenger boarded the Titanic, where C = Cherbourg, S = Southampton, and Q = Queenstown	Categorical (Nominal)

Appendix B

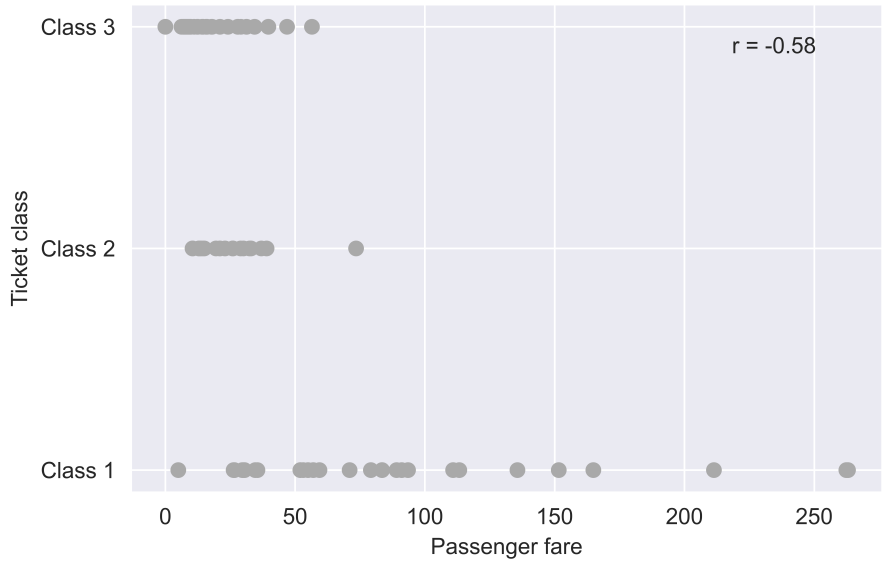


Figure 14: Scatter plot and the Pearson correlation coefficient of ticket class and passenger fare.