MACHINE LEARNING METHODS FOR EARTHQUAKE RISK PREDICTION

Math 490 - Graduation Project

2025-2026 Spring Semester

Author
Tuğçe NALVURAN

Supervisor
Dr. Esra KARAOĞLU

Department of Mathematics

Çankaya University

Contents

List of Figures

Machine Learning Methods for Earthquake Risk Prediction

Tuğçe Nalvuran

Abstract

Earthquakes, occurring as a result of sudden energy releases in the lithosphere, pose a significant risk, especially for countries situated on active tectonic belts such as Turkey. Traditional statistical seismology approaches may be limited in modeling the non-linear and multidimensional complex structures inherent in seismic data. In this study, the seismic hazard analysis of Turkey is examined from the perspective of machine learning algorithms, utilizing current earthquake catalog data between the years 2020 and 2026 obtained through the Boğaziçi University Kandilli Observatory and Earthquake Research Institute Regional Earthquake Monitoring and Evaluation Center. The primary objective of this thesis, which is a mathematics senior project, is to investigate the foundations of probability theory, optimization, and information theory behind data-driven prediction models, specifically Logistic Regression, Support Vector Machines (SVM), and Random Forest algorithms. Models will be trained to classify and predict earthquake magnitudes, and the mathematical behaviors and prediction performances of these algorithms on seismic data will be analyzed comparatively. Consequently, the study will demonstrate through numerical data which algorithm generates statistically more consistent results in modeling complex seismic networks.

Keywords: Seismic Risk Analysis, Machine Learning, Logistic Regression, Support Vector Machines (SVM), Random Forest, Optimization Theory.

1 Introduction

Earthquakes are complex natural disasters with devastating effects, occurring as a result of sudden energy releases in the lithosphere. Especially in regions situated on active tectonic belts such as Turkey, the analysis of seismic activities and the modeling of potential future seismic hazards are of vital importance. Traditionally, earthquake predictions and hazard analyses rely on statistical seismology theories such as Poisson processes and the Gutenberg-Richter relationship. However, the non-linear complexity and multidimensionality inherent in the nature of seismic data challenge the assumptions of classical statistical models (e.g., the independence of events) and limit their predictive capabilities.

In light of increasing computational power and algorithmic developments in recent years, data-driven approaches, namely Machine Learning (ML) techniques, have begun to be integrated into the field of seismology. This study aims to bring a mathematical approach to seismic risk analysis by utilizing earthquake catalog data occurring in Turkey between 2020 and 2026.

The primary objective of this thesis is to examine the mathematical and statistical foundations behind machine learning algorithms—Logistic Regression, Support Vector Machines, and Random Forest—which are often perceived as "black boxes," and to analyze their performances in seismic magnitude prediction comparatively. By employing concepts from optimization theory, probability distributions, and information theory, Turkey's seismic patterns will be modeled in a mathematical space, and the theoretical background of the obtained empirical results will be discussed.

Studies on earthquake analysis and prediction have undergone a distinct evolution from statistical modeling toward multidimensional space analysis based on machine learning over the last two decades. In the literature, the three main components of the earthquake prediction problem—time, location, and magnitude estimation—have been mathematically modeled. Early seismic research was based on fundamental probability theories. Particularly, Poisson distributions and Markov Chains were frequently utilized in studies where seismic occurrences were treated as random (stochastic) processes. While the Gutenberg-Richter law ($\log_{10} N = a - bM$), which models the magnitude-frequency relationship, is accepted as the cornerstone of long-term hazard analysis in the literature, it has remained insufficient regarding short-term pattern recognition.

Examining current literature reveals that Machine Learning (ML) algorithms have become prevalent in deciphering complex non-linear seismic data. Researchers often use Logistic Regression models as a fundamental baseline for classifying earthquake occurrence probabilities. The ability of logistic regression to weight seismic indicators via Maximum Likelihood Estimation (MLE) and transform them into an occurrence probability between 0 and 1 (through the Sigmoid function) has led to its frequent reference in probabilistic seismic hazard analyses. In cases where the size of the dataset and the number of features increase, the literature has turned toward information theory-based ensemble algorithms. Particularly, the Random Forest algorithm stands out as one of the methods offering the highest stability in seismic catalog data due to its variance reduction property and its ability to calculate the mathematical "importance degrees" (feature importance) of variables (e.g., depth or historical earthquake counts) in earthquake prediction via Gini impurity. On the other hand, in complex spaces where seismic data cannot be linearly separated, the literature resorts to Support Vector Machines (SVM). SVM's mathematical structure,

which seeks the maximum margin using constrained optimization (Lagrange multipliers), and its ability to project data into higher-dimensional Hilbert spaces thanks to the Kernel trick, have been presented by researchers as a powerful tool, especially in classifying foreshocks and microseismic events. In summary, the literature converges on the necessity of selecting models that make the most appropriate mathematical assumptions for the spatial and temporal structure of the data, rather than searching for a single "perfect" algorithm. Accordingly, our thesis aims to provide a mathematical synthesis by applying this theoretical framework to Turkey's local seismic data.

In this study, three different machine learning algorithms based on statistics, optimization, and information theory will be utilized to analyze the non-linear relationships in Turkey's seismic catalog data and to perform earthquake hazard prediction.

Logistic Regression will be used as the baseline model in the seismic risk analysis. Unlike linear regression, it calculates the probability of an earthquake occurring that exceeds a certain threshold (e.g., $M \geq 5.0$) rather than predicting a continuous value. *Mathematical Basis:* It is based on statistical probability theory. The model places the linear combination of seismic features into the Sigmoid function ($S(x) = \frac{1}{1+e^{-x}}$), transforming the result into a probability value between 0 and 1. The model coefficients will be optimized using Maximum Likelihood Estimation (MLE), a differentiable optimization problem, and Gradient Descent algorithms.

Support Vector Machines (SVM) will be employed for classification in complex cases where seismic data cannot be linearly separated (e.g., the overlapping of hazardous and non-hazardous seismic zones). *Mathematical Basis:* It is based on geometry and constrained optimization theory. The primary goal of the algorithm is to find the optimal hyperplane that separates data belonging to different classes and maximizes the margin (distance) between them. To project data into higher-dimensional spaces and make them linearly separable, the Kernel Trick (Mercer's Theorem) will be utilized, and the optimization process will be modeled using Lagrange Multipliers.

Random Forest, an ensemble version of Decision Trees, is included in the project to achieve high accuracy rates and prevent overfitting, particularly in tabular seismic catalog data. *Mathematical Basis:* It is based on information theory and statistical variance analysis. The algorithm generates hundreds of independent decision trees by selecting random subsets from the dataset (Bootstrap Aggregating / Bagging). The branching points of the trees are determined by calculating mathematical metrics such as Gini Impurity or Shannon Entropy, which minimize the uncertainty at each node. This model will enable the identification of which variable, such as depth or location, is mathematically more "important" in earthquake prediction.

2 Literature Survey

Seismic hazard assessment and earthquake prediction have long been focal points of geophysical research due to the catastrophic socio-economic impacts of major tremors. While conventional seismological models provide fundamental insights into fault mechanics and wave propagation, the increasing availability of granular seismic catalogs has paved the way for advanced data-driven approaches. A significant portion of the literature focuses on temporal wave activities and regional ground motion parameters. However, accurately classifying seismic risk—specifically determining whether an impending event will exceed

a destructive magnitude threshold—requires a multidimensional understanding of spatial parameters such as latitude, longitude, and focal depth. The following literature survey explores various methodologies ranging from ground motion analyses to machine learning predictions, highlighting the evolution of earthquake forecasting techniques that establish the theoretical foundation for the binary classification models employed in this study.

Q. Wang et al.[13] employed a deep learning approach called LSTM (Long short-term memory) networks to apprentice the Spatio-temporal relation amidst earthquakes in varied regions and make foresight by taking the benefit of that particular relationship. The outcomes show that these networks with 2-D input can exploit the correlations that are Spatio-temporal to make far better predictions. K.M. Asim et al.[14] worked on the prediction of the magnitude of earthquakes using the temporal arrangement of seismic wave activities, combined through various machine learning classification algorithms. The prognosis was done on the foundation of eight seismic indicators utilizing the catalog of earthquakes. In these four techniques, including recurrent neural networks, pattern recognition neural networks, linear programming boost, and ensemble classifier and random forest classifiers were used to calculate seismic parameters and further occurrences of earthquakes. H. Cam et al.[15] worked on a feed-forward back propagation artificial neural network related to Gutenberg-Richter relation, based on which b values are used in earthquakes is developed. G. Asencio-Cortes et al.[16] analyzed the effect of using various parameterizations for inputs in the supervised learning algorithms through a new framework. Five varied analyses were conducted, which involved the tweaking of training and testing sets for the scheming of b-value and the tuning of collected gauges.

V.G. Gitis et al.[17] suggested a new technique to estimate the constraints of inhomogeneous spatiotemporal marked point fields. It is built on the idea of adaptive weights smoothing (AWS). In this paper, a wide variety of the AWS algorithm is constructed to calculate the spatial and spatiotemporal fields of density, the mean values, along with the correlation dimension. This algorithm is utilized to assess the seismic process criterion fields from certain earthquake litanies. The AWS forecasting method surpassed the forecasting using kernel estimation. J. R. Holliday et al.[18] worked on the informatic pattern analysis by using the complex eigenvectors and created the short-term forecast of hotspot maps that are different from hotspot maps which are created by using real-valued data. They also suggested various methods of analyzing differences and computing the information gain. G. M. Molchan et al.[19] analyzed the portentous seismicity methodology, also described as pattern B, which is evaluated in 13 areas of the world. Its great demographic connotation is assured. The mathematical accession advanced here is useful in the analysis of the harbinger of earthquakes.

G. Lanzano et al.[20] worked on revising the framework of ground agitation for trifling the crustal earthquakes that are taking place in Italy, tapered in the 4.0 to 6.9 magnitude range. It utilizes durable-motion data that is measured up to the 2009 L'Aquila Sequence. Further, in this, the new collection of data allows us to extend the range of the magnitude exceeding 6.9, including vibration periods of up to 10 s. The ground agitation variability is broken down within components amidst event and site to site to form the model, which is suitable for the assessment of non-governable probabilistic seismic hazard. B. Idini et al.[21] worked on a database of robust agitation records for the Chilean subduction earthquake zones. They made a ground motion prescience equation (GMPE) for apex ground acceleration along with a riposte spectral expedition with a 5 % damping proportion for periods in between 0.01 and 10 s. D. Ju et al.[22] proposed two contemporary procedures

for the evaluation of fault parameters of asperity frameworks for the prognosis of tenacious ground agitations from crustal level earthquakes. One is for the long strike-slip faults, and the other one is for lengthy antipode faults.

C. Papantonopoulos et al.[23] used the unique element method to foretell the earthquake counter of the multi-drum marble framework of a restrained column. The outcomes are compared with the experimental data for a similar specimen under similar excitation. The experiments and the numerical analysis both took place in 3D. The results tell that the distinct element methodology can captivate the main features of the response. G. F. Panza et al.[24] presented an extended development of assimilation of seismological as well as geodetic instruction, showing the benefaction of geodesy to the realization and prognosis of earthquakes. P. kundu et al.[25] proposed a probabilistic accession for the estimation of the expected rebound time of an earthquake of a particular magnitude, within an anchored life span of structure succeeded by the determination of the peak grounded acceleration at the site of structure in Chilean area based on the Gutenberg Richter Law. The data here is held from the USGS (United States Geological Survey), and the procedure here can be applied even before the construction of structure at a site to appease the death toll caused due to the collapse of the structure.

As observed in the aforementioned studies, the majority of current research relies heavily on seismic wave characteristics, temporal patterns, and geodetic data for magnitude prediction or ground motion estimation. However, the direct utilization of fundamental geological coordinates—namely latitude, longitude, and depth—to perform a robust classification of seismic risk remains relatively underexplored in regional contexts. This study directly addresses this gap by shifting the focus from continuous magnitude regression to a probabilistic risk classification framework (Risky vs. Safe for $M \geq 5.0$) tailored specifically to the complex fault topologies of the Central Anatolia Region. By applying and analytically comparing algorithms such as Logistic Regression, Support Vector Machines (SVM), and Random Forest, this work aims to capture the non-linear spatial dependencies of earthquakes. Ultimately, concentrating on these core spatial parameters offers a computationally efficient approach to identifying high-risk seismic zones, thereby contributing to proactive disaster management and the mitigation of structural and human losses. Figure 1 outlines the comprehensive workflow proposed to evaluate the seismic risk, detailing the stages from data preprocessing to model evaluation.

Figure 1: Workflow of the Seismic Risk Classification Methodology

3 Some Machine Learning Methods

3.1 Linear and Logistic Regression

What Does Regression Mean?

Regression in statistics means finding a relationship between variables and predicting values. For example, estimate a house price in TL based on its square meter area.

3.1.1 What Is Linear Regression?

Linear regression is one of the most fundamental supervised learning algorithms in machine learning and statistical analysis. Its main purpose is to model the relationship between one or more independent variables (input features, X) and a continuous dependent variable (output, y), which takes values in the set of real numbers, using a linear equation.

Geometrically, this process corresponds to drawing the "best-fit line" or "hyperplane" in an n-dimensional space that minimizes the squared distances (errors) between the data points and the model.

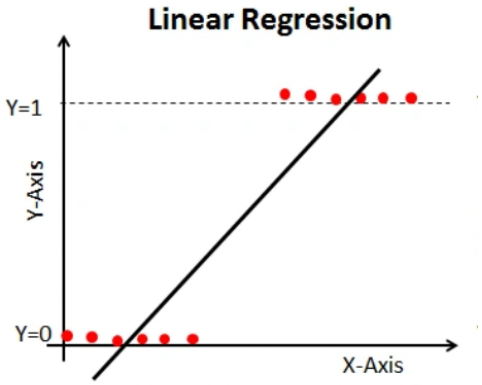

Figure 2: Linear Regression Model

The mathematical model of linear regression (hypothesis function, $h_\theta(x)$) is obtained by multiplying the input variables by their coefficients (weights) and summing them:

$$h_\theta(X) = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \cdots + \theta_n X_n \tag{1}$$

- $h_\theta(X)$: The predicted value produced by our model (Output).
- $X_1, X_2, \ldots, X_n$: Our input variables (Features).
- θ_0: The point where the line intersects the y-axis (Bias / Constant Term).
- $\theta_1, \theta_2, \ldots, \theta_n$: Weights (Coefficients). These are slope values that indicate how much each feature affects the result.

Using matrix and vector algebra, this equation can be expressed much more concisely through the dot product rule in linear algebra:

$$h_\theta(X) = \theta^T X \tag{2}$$

Cost Function (MSE):
To measure how much the line drawn by the model deviates from the actual data points (that is, how much error it makes), The Mean Squared Error (MSE) function is used:

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(X^{(i)}) - Y^{(i)})^2 \tag{3}$$

Our objective is to find the θ (weight) parameters that minimize this error function.

- $J(\theta)$ **(Cost Function):** Represents the total amount of "error" made by the model. Our aim is always to reduce this J value to its minimum.
- θ **(Theta - Parameters/Weights):** These are the coefficients that our model attempts to learn ($\theta_0, \theta_1, \theta_2 \ldots$).
- $X^{(i)}$ **(ith Input Vector):** Represents the features (inputs) of the ith data point in the dataset (for example, the features in the 5th row). (Important Note: The (i) here is not an exponentiation operation; it only indicates an index, that is, the row number).

- $Y^{(i)}$ **(ith Actual Value):** The actual result of the ith data point in the dataset. (Example: the actual price of the house in the 5th row).
- $h_\theta(X^{(i)})$ **(Hypothesis - Predicted Value):** The prediction produced by our model for the input $X^{(i)}$ using its own θ coefficients. (Example: the price predicted by the model for the house in the fifth row).

3.1.1 Analytical Solution (Normal Equation / Ordinary Least Squares)

If our dataset is not extremely large, machine learning algorithms perform a direct and one-time calculation using matrix algebra instead of "making predictions and correcting the error." This method is called the Least Squares method.

We write all input data we have (for example, the square meter values of houses) as the matrix X, and the actual results (house prices) as the vector Y. To find the point where our cost function, the Mean Squared Error (MSE), is minimized, we take the partial derivative of the function and set it equal to 0:

$$\frac{\partial J(\theta)}{\partial \theta} = 0 \tag{4}$$

When this derivative operation is solved in matrix form, the well-known Normal Equation is obtained:

$$\theta = (X^T X)^{-1} X^T Y \tag{5}$$

- X^T: The transpose of the input matrix.
- $(X^T X)^{-1}$: The inverse of the product matrix.

The coefficients are not learned "by themselves"; in the background, the computer finds the θ values by performing this complex matrix inversion operation.

In simple linear regression (single-variable) ($h_\theta(x) = \theta_0 + \theta_1 x$), these coefficients are calculated directly using statistical variance and covariance formulas as follows:

$$\theta_1 = \frac{\sum_{i=1}^{m}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{m}(x_i - \overline{x})^2} \tag{6}$$

$$\theta_0 = \overline{y} - \theta_1 \overline{x} \tag{7}$$

(Here, $\overline{x}$ and $\overline{y}$ are the arithmetic means of the entries in x and y vectors, respectively).

3.2.1 Example 1: Manual Calculation of the Coefficients

This example is presented independently of earthquakes to demonstrate how the model performs mathematical calculations in the background.

Let us model the relationship between students' "Study Hours" (x) and their "Exam Scores" (y) in a class. Suppose that we only have data from 3 students:

- Student 1: $x_1 = 2$ hours, $y_1 = 40$ points
- Student 2: $x_2 = 4$ hours, $y_2 = 60$ points

- Student 3: $x_3 = 6$ hours, $y_3 = 80$ points

Step 1: Finding the Averages

$$\overline{x} = \frac{2+4+6}{3} = 4$$

$$\overline{y} = \frac{40+60+80}{3} = 60$$

Step 2: Calculation of the θ_1 (Slope / Effect) Coefficient
Numerator (Covariance part): $(2-4)(40-60) + (4-4)(60-60) + (6-4)(80-60)$

$$\text{Numerator} = (-2)(-20) + (0)(0) + (2)(20) = 40 + 0 + 40 = 80$$

Denominator (Variance part): $(2-4)^2 + (4-4)^2 + (6-4)^2$

$$\text{Denominator} = (-2)^2 + 0^2 + 2^2 = 4 + 0 + 4 = 8$$

$$\theta_1 = \frac{80}{8} = 10$$

The mathematical meaning of this is as follows: For every additional 1 hour a student studies, the student gains 10 points.

Step 3: Calculation of the θ_0 (Constant / Intercept) Coefficient

$$\theta_0 = \overline{y} - \theta_1 \overline{x} \implies \theta_0 = 60 - (10 \times 4) = 60 - 40 = 20$$

The mathematical meaning of this is as follows: A student who does not study at all ($x = 0$) would receive a baseline score of 20 on paper.

By analyzing the dataset, the machine learning algorithm has constructed the hypothesis function that reduces the error to zero as follows:

$$h_\theta(x) = 20 + 10x \tag{8}$$

The model can now predict, with mathematical certainty, the score of a new student who says "I studied for 5 hours" as $h_\theta(5) = 20 + 10(5) = 70$.

3.2.2 Example 2: House Price Prediction

Let us manually calculate how the coefficients in the previous example were obtained. Suppose we have a single-variable model only (Square Meters = x, Price = y).

In simple linear regression, the formula for the θ_1 (slope) coefficient is as follows (Covariance / Variance):

$$\theta_1 = \frac{\sum_{i=1}^{m}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{m}(x_i - \overline{x})^2}$$

The θ_0 (intercept) is found using the following formula:

$$\theta_0 = \overline{y} - \theta_1 \overline{x}$$

Our Dataset (3 houses):

1. House: $x_1 = 80$ m^2, $y_1 = 900.000$ TL
2. House: $x_2 = 100$ m^2, $y_2 = 1.100.000$ TL
3. House: $x_3 = 120$ m^2, $y_3 = 1.300.000$ TL

Step-by-Step Calculation:
Averages: $\overline{x} = 100$, $\overline{y} = 1.100.000$
Numerator (Covariance part): $(80 - 100)(900.000 - 1.100.000) + (100 - 100)(0) + (120 - 100)(1.300.000 - 1.100.000)$

$$\text{Numerator} = (-20)(-200.000) + 0 + (20)(200.000) = 4.000.000 + 4.000.000 = 8.000.000$$

Denominator (Variance part): $(80 - 100)^2 + (100 - 100)^2 + (120 - 100)^2$

$$\text{Denominator} = (-20)^2 + 0 + (20)^2 = 400 + 400 = 800$$

Finding the θ_1 Coefficient: $\theta_1 = \frac{8.000.000}{800} = 10.000$
Finding the θ_0 Coefficient: $\theta_0 = 1.100.000 - (10.000 \times 100) = 1.100.000 - 1.000.000 = 100.000$

Our model has mathematically calculated the equation as $h_\theta(x) = 100.000 + 10.000x$.

3.1.2 The Insufficiency of Linear Regression in Classification Problems

As seen in the example above, linear regression is an excellent mathematical tool for predicting continuous values such as scores, prices, and temperature. However, in binary classification problems such as "Will the earthquake exceed a magnitude of 5.0?" (1: Yes / Risky, 0: No / Not Risky), which is the focus of this thesis, it creates two fundamental mathematical problems:

1. Violation of Output Bounds (Probability Paradox)
In classification problems, the expected output of the model is the probability of an event occurring, and according to probability theory, this value must lie within the interval $[0, 1]$ ($0 \leq h_\theta(x) \leq 1$). However, since the hypothesis function of Linear Regression is a straight line, its output ranges from $-\infty$ to $+\infty$. The model may produce undefined and meaningless results such as "the risk probability of this earthquake is -4.2 or 18.5."

2. Excessive Sensitivity to Outliers
In linear regression, a threshold value of 0.5 is generally used to separate classes ($h_\theta(x) \geq 0.5$ means 1, otherwise 0). If an outlier with very high values is added to the dataset (for example, an extremely large seismic movement independent of other earthquakes), Linear Regression tilts the line it draws toward that distant point in order to reduce the Mean Squared Error (MSE). Since the slope changes, the 0.5 decision boundary also shifts, and the model begins to misclassify seismic movements that it had previously predicted correctly.

Therefore, switching to the **Logistic Regression** algorithm is a mathematical necessity in order to restrict the hypothesis function and transform the linear output into a probability distribution within the interval $[0, 1]$.

3.1.3 Where Does the Word Logistic Come From?

The word "logistic" comes from the special mathematical curve at the core of this model, known as the Logistic Function or Sigmoid Function. Since the name of this function is "Logistic," the prediction method that uses it is called Logistic Regression. The origin of the term dates back to Pierre François Verhulst, who used this curve in the 19th century to model population growth. Table 1 outlines the fundamental theoretical differences and performance characteristics comparing the continuous nature of Linear Regression with the binary classification capabilities of Logistic Regression.

Feature	**Linear Regression**	**Logistic Regression**
Predicted Output	Numerical Value (Continuous)	Probability/Class (Categorical)
Output Range	Unbounded $(-\infty, +\infty)$	Bounded (between 0 and 1)
Application Area	"What will the air temperature be?"	"Will it rain tomorrow? (Yes/No)"

Table 1: Linear Regression Model

3.1.4 What Is Logistic Regression and Why Is It Used?

Although the term "regression" appears in its name, logistic regression is a statistical binary classification algorithm. Instead of producing a continuous real number as output, it produces the probability (P) that a specific event will occur. This probability value must lie within the closed interval $[0, 1]$, which represents strict mathematical boundaries.

Logistic regression analysis is used to examine the relationship between independent variable(s), which may be categorical or continuous, and a binary dependent variable [3]. This method differs from linear regression analysis, where the dependent variable is continuous.

A special function is required to compress, or map, the unbounded linear hypothesis produced by linear regression, $h_\theta(x) = \theta^T X$, which lies in the interval $(-\infty, +\infty)$, into the probability interval $[0, 1]$. This function is the **Sigmoid (Logistic) Function**, which is widely used in differential equations and population growth models.

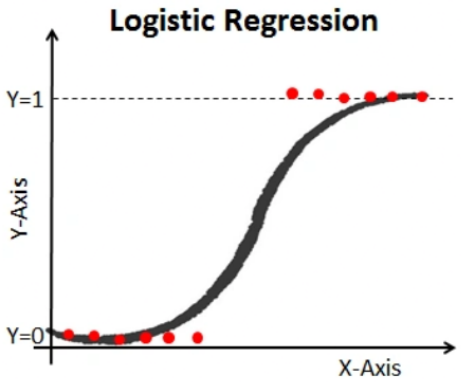

Figure 3: Logistic Regression Model

3.1.5 Sigmoid Function

To compress the output value obtained from linear regression, which may range from $-\infty$ to $+\infty$, into the interval between 0 and 1, we use the Sigmoid (Logistic) Function. The sigmoid function is a continuous and differentiable S-shaped function that transforms any real number z given to it into a value between 0 and 1. The equation of the function is as follows:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{9}$$

e: Euler's number, the base of the natural logarithm (≈ 2.71828).

In logistic regression, we take our linear equation $z = \theta^T X$ and place it inside the Sigmoid function. In this way, our new **Logistic Hypothesis Function** is obtained:

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T X}} = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n)}} \tag{10}$$

- If $\theta^T X$ is a very large positive number ($z \to \infty$), the value of e^{-z} approaches 0. The result becomes $1/(1 + 0) = 1$.
- If $\theta^T X$ is a very small negative number ($z \to -\infty$), the value of e^{-z} approaches infinity. Then, the result goes to 0.
- If $\theta^T X = 0$, then $e^0 = 1$, and the result is $1/(1 + 1) = 0.5$. This value is the **Decision Boundary** that separates the classes.

In general, if $h_\theta(x) \geq 0.5$, the event is assumed to occur ($y = 1$), whereas if $h_\theta(x) < 0.5$, it is assumed not to occur ($y = 0$).

3.1.6 The Concept of Log-Odds (Logit): Where Does the Model Get Its Linearity From?

If the probability of being "Risky" is p, then the probability of being "Not Risky" is $(1 - p)$. The ratio of these two probabilities to each other is called the Odds:

$$\text{Odds} = \frac{p}{1 - p}$$

The main mathematical fact at the core of Logistic Regression is this: The model does not directly equate the probability (p) to a linear equation. The actual linear equation of logistic regression is equal not to the probability itself, but to the **natural logarithm of the odds ratio (Log-Odds)**:

$$\text{Logit}(p) = \ln\left(\frac{p}{1 - p}\right) = \theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n \tag{11}$$

Logistic Regression models the logarithm of probabilities over odds ratios as a linear line, that is, a regression. The sigmoid function is essentially the inverse function of this Logit equation. For this reason, it is called Logistic Regression rather than Logistic Classification.

3.1.7 Optimization: Why Cannot Mean Squared Error (MSE) Be Used? (The Need for a Cost Function / Log-Loss)

In linear regression, we used the formula $J(\theta) = \frac{1}{2m}\sum(h_\theta(x) - y)^2$ as the cost, or error, function. However, in Logistic Regression, the function $h_\theta(x)$ is no longer linear; it contains a complex exponential structure such as $1/(1 + e^{-z})$.

If this Sigmoid hypothesis is inserted into the MSE formula, the resulting cost function graph takes on a **Non-Convex** structure. In other words, many fluctuations, peaks, and local minima occur on the graph. While the optimization algorithm attempts to find the deepest valley, the Global Minimum, it may get stuck in these smaller valleys and produce incorrect coefficients (θ) [11].

To solve this mathematical problem, a new fully convex cost function, shaped like a bowl with a single bottom point, called **Cross-Entropy / Log-Loss**, is derived using Maximum Likelihood Estimation (MLE) in statistics [12]:

$$\text{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases} \tag{12}$$

- **Case $y = 1$ $(-\log(h_\theta(x)))$:** If the model's prediction is 1, meaning it is correct, then $-\log(1) = 0$. In other words, there is no penalty. If the model's prediction approaches 0, meaning it is completely wrong, then $-\log x \to \infty \quad \text{as} \quad x \to 0^+$. In other words, we tell the model: "You made a very large error!"
- **Case $y = 0$ $(-\log(1 - h_\theta(x)))$:** If the model's prediction is 0, meaning it is correct, then $-\log(1 - 0) = 0$. Again, there is no penalty. If the model's prediction approaches 1, then $-\log x \to \infty \quad \text{as} \quad x \to 0^+$.

This two-part function is combined into a single equation for all m samples in the dataset:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log(h_\theta(x^{(i)})) + (1 - y^{(i)})\log(1 - h_\theta(x^{(i)}))\right] \tag{13}$$

3.1.8 Parameter Optimization: Gradient Descent

After the cost function is determined, we must update our parameters so that the model's prediction becomes accurate. We perform the parameter update process using Gradient Descent. Our objective is to find the θ coefficients that bring this error function $J(\theta)$ as close as possible to zero, that is, to its minimum value.

In logistic regression, this process cannot be solved using the matrix inverse method, namely the Normal Equation. Instead, the partial derivative of $J(\theta)$ with respect to θ is taken. The partial derivative, or gradient, turns into a surprisingly simple formula:

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)} \tag{14}$$

At each iteration, the algorithm updates the θ_j coefficients using the following rule ($\alpha =$ Learning Rate):

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \tag{15}$$

Critical Theoretical Information: In linear regression, we had a single formula, the Normal Equation, to find the coefficients. However, since our hypothesis in Logistic Regression contains an exponential function such as the Sigmoid function, there is no closed-form analytical solution when we set the derivative equal to zero. In other words, due to the nature of mathematics, it is impossible to obtain an algebraic formula that isolates θ.

Therefore, to find the coefficients manually, we must use an iterative method, a step-by-step numerical approximation approach, called Gradient Descent.

3.1.9 Example Independent of Earthquake Risk Analysis: How Are the Coefficients (θ) Found in Logistic Regression?

Cholesterol-Heart Attack Risk Analysis:
Let us model the effect of the patient's Cholesterol Level (x) on Heart Attack Risk (y). Suppose we only have 2 patients.

- Patient 1: Cholesterol (x_1) = 2, No Heart Attack (y_1) = 0
- Patient 2: Cholesterol (x_2) = 5, Heart Attack Occurred (y_2) = 1

Number of Data Points (m) = 2

At the beginning, the computer, and we as well, always assumes the coefficients to be zero (0). Let our learning coefficient (α) be 0.1.

- **Parameters (θ):** These are the values that the machine learns by itself by examining the dataset and performing derivative-based optimization. In manual calculation, they are assigned by the researcher.
- **Learning Rate (α) and Hyperparameter Selection:** The α value in the Gradient Descent equation, known as the Learning Rate, is a hyperparameter that is not learned by the model but is assigned externally by the researcher. The optimization of this value, which controls the step size, is critical for model convergence.
  - If α is **Chosen Too Large** (Example: $\alpha = 10$): The steps become so large that the algorithm misses the lowest point of the error function, the minimum, and jumps to the opposite side of the valley. It gradually climbs even higher, and the error explodes toward infinity, leading to divergence.
  - If α is **Chosen Too Small** (Example: $\alpha = 0.000001$): The algorithm progresses with tiny steps. It may take days or months to descend into the deepest point. In addition, it may get stuck in a small valley on the map, a Local Minimum, and remain there without finding the actual deepest point.

$\theta_0 = 0$, $\theta_1 = 0$, $\alpha = 0.1$

Step 1: Making the Initial Predictions
Let us estimate the patients' risks using the current zero weights:

Prediction for Patient 1 ($h_\theta(x^{(1)})$):

$$z_1 = \theta_0 + \theta_1 x^{(1)} = 0 + 0(2) = 0$$

$$\text{Sigmoid } P_1 = \frac{1}{1+e^0} = \frac{1}{1+1} = 0.5$$

Prediction for Patient 2 ($h_\theta(x^{(2)})$):

$$z_2 = \theta_0 + \theta_1 x^{(2)} = 0 + 0(5) = 0$$

$$\text{Sigmoid } P_2 = \frac{1}{1+e^0} = 0.5$$

Since the model does not know anything at this point, it assigns a 50% risk to everyone.

Step 2: Calculating the Derivative of the Error (Gradient)
Let us calculate the partial derivatives, or gradients, of our $J(\theta)$ Log-Loss cost function.

Partial Derivative for θ_0 (Constant) (x_0 is always 1):

$$\frac{\partial J}{\partial \theta_0} = \frac{1}{2}\left[(P_1 - y^{(1)}) \cdot 1 + (P_2 - y^{(2)}) \cdot 1\right]$$

$$\frac{\partial J}{\partial \theta_0} = \frac{1}{2}\left[(0.5 - 0) + (0.5 - 1)\right] = \frac{1}{2}[0.5 - 0.5] = 0$$

Partial Derivative for θ_1 (Cholesterol Slope):

$$\frac{\partial J}{\partial \theta_1} = \frac{1}{2}\left[(P_1 - y^{(1)}) \cdot x^{(1)} + (P_2 - y^{(2)}) \cdot x^{(2)}\right]$$

$$\frac{\partial J}{\partial \theta_1} = \frac{1}{2}\left[(0.5 - 0) \cdot 2 + (0.5 - 1) \cdot 5\right]$$

$$\frac{\partial J}{\partial \theta_1} = \frac{1}{2}\left[1 + (-0.5 \cdot 5)\right] = \frac{1}{2}[1 - 2.5] = \frac{-1.5}{2} = -0.75$$

Step 3: Updating the Weights
Let us update the coefficients using these derivative values:

$$\theta_{\text{new}} = \theta_{\text{old}} - \alpha(\text{Derivative})$$

New θ_0:

$$\theta_0 = 0 - 0.1 \times (0) = 0$$

New θ_1:

$$\theta_1 = 0 - 0.1 \times (-0.75) = 0 + 0.075 = 0.075$$

$$Z = \theta_0 + 0.075x$$

The coefficient θ_1 increased from 0 to +0.075. In other words, using mathematics, the model made the following inference: Patient 2 had a higher cholesterol level ($x = 5$), and that patient had a heart attack ($y = 1$). Therefore, as cholesterol increases, the risk of heart attack also increases. For this reason, we should increase the coefficient, or slope, of x in the positive direction.

3.2 Support Vector Machines (SVM) and Geometric Optimization

The decision boundary produced by Logistic Regression ($Z = 0$) is, by its nature, a linear hyperplane and leads to underfitting in complex seismic problems where the data are interwoven. To overcome this limitation and separate the classes in the dataset (Risky and Non-Risky) with the highest possible reliability, the study proceeds to the Support Vector Machines (SVM) algorithm, which is based not on statistical probability but on analytic geometry and constrained optimization.

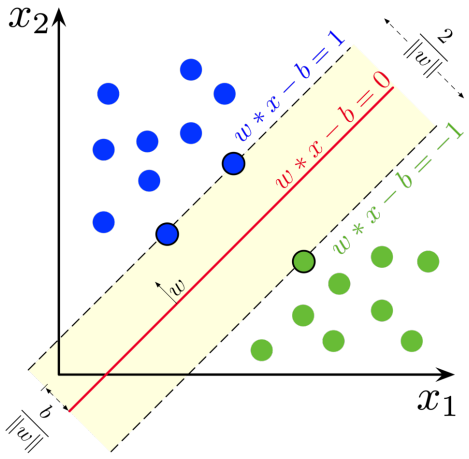

Figure 4: Support Vector Machine Graph

3.2.1 Primal Problem: Geometry and Maximum Margin

Our objective is to draw the "widest street (Maximum Margin)" that separates risky ($M \geq 5.0$) and non-risky earthquakes. In order to establish a symmetric geometry, the dependent variable labels in SVM are updated from $y \in \{0, 1\}$ to $y \in \{-1, 1\}$.

Our decision boundary separating the classes (the main hyperplane) and the parallel lines determining the boundaries of the street (margins) are expressed by the following equations:

$$w^T x + b = 0 \quad \text{(Main Decision Boundary)} \tag{16}$$

$$w^T x + b \geq 1 \quad \text{(Boundary for the positive class / risky earthquakes)} \tag{17}$$

$$w^T x + b \leq -1 \quad \text{(Boundary for the negative class / non-risky earthquakes)} \tag{18}$$

Meanings of the variables in these equations:

- w **(Weight / Normal Vector):** This is the coefficient vector perpendicular to the hyperplane and determines the orientation, or slope, of the plane in space.
- x **(Input Vector):** This is the coordinate vector representing the independent variables in our dataset (e.g., x_1 = Depth, x_2 = Latitude).
- b **(Bias / Constant Term):** This is the scalar value determining the perpendicular distance, or shift, of the hyperplane from the origin.

According to the rules of analytic geometry, the perpendicular distance between these two parallel hyperplanes, that is, the width of the street, is given by the formula $\frac{2}{||w||}$. Our objective is to **maximize** this width. Maximizing a fraction is equivalent to minimizing its denominator, namely $||w||$. To simplify the derivative operations, we take the square of the norm, and the following **Primal Optimization Problem** emerges:

$$\min_{w,b} \frac{1}{2}||w||^2 \tag{19}$$

Constraint: $y^{(i)}(w^T x^{(i)} + b) \geq 1 \quad$ (for each data point i)

Meanings of the variables in this formula:

- $||w||$ **(Euclidean Norm of the vector w):** This is the geometric length of the weight vector.
- $y^{(i)}$ **(ith True Label):** This is the true class of the ith data point (if Risky, $+1$; if Non-Risky, -1).
- $x^{(i)}$ **(ith Input Vector):** This is the data point containing the features of the ith seismic event.
- **Meaning of the Constraint:** This is a strict rule expressing mathematically that no data point should fall inside the street, that is, the margin, or remain on the wrong side.

3.2.2 The Heart of the Mathematics: Lagrange Multipliers and Duality

The equation above is a problem of **constrained optimization**. To solve constrained problems, inequality constraints are integrated into the main function using **Lagrange Multipliers (α)**, yielding a single **Lagrangian ($\mathcal{L}$)** function:

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{m} \alpha_i \left[y^{(i)}(w^T x^{(i)} + b) - 1\right] \tag{20}$$

Meanings of the variables in this formula:

- $\mathcal{L}$ **(Lagrangian Function):** This is the combined function that transforms the constrained optimization problem into a single equation by incorporating the constraints.
- α_i **(Lagrange Multiplier):** This is the mathematical multiplier assigned to each data point (i), measuring how much that point violates the constraint or how tightly it lies against it ($\alpha_i \geq 0$).
- m **(Number of Data Points):** This is the total number of samples, or rows, in the dataset.
- $\sum$ **(Summation Symbol):** This denotes the cumulative summation over the entire dataset.

When the partial derivatives of this equation with respect to w and b are taken and set equal to zero, it is proven that the weight vector w is in fact a linear combination of the

data:

$$w = \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)} \tag{21}$$

Geometric Proof: This derivative result shows that the **weight vector (w) is nothing more than a linear combination of the inputs (x) in the dataset.**

3.2.3 KKT Conditions and the Mathematical Proof of Support Vectors

The reason why the algorithm is called "Support Vector Machines" is proven through the rule of **Complementary Slackness**, one of the **Karush-Kuhn-Tucker (KKT) Conditions**:

$$\alpha_i \left[y^{(i)}(w^T x^{(i)} + b) - 1 \right] = 0 \tag{22}$$

For this product to be zero, there are mathematically two possibilities:

1. **The Data Point Is Outside the Street (Safe):** The expression inside the brackets is greater than zero. For the equality to hold, α_i must necessarily be equal to 0. In other words, distant seismic data have no effect, or weight, on the model equation.
2. **The Data Point Lies Exactly on the Margin Boundary:** In this case, $y^{(i)}(w^T x^{(i)} + b) = 1$ holds, meaning the expression inside the brackets is zero. Then $\alpha_i > 0$.

These critical points, which determine the margin boundary and whose α_i values are nonzero, are called **Support Vectors**. The algorithm disregards the rest of the data and constructs the hyperplane based only on these vectors.

3.2.4 Class Imbalance and Asymmetric Penalty (Soft Margin)

In the studied dataset of 1362 rows, there are only 8 earthquakes with magnitude $M \geq 5.0$. If hard constraints (Hard Margin) are applied to such a dataset that is not perfectly linearly separable, the system becomes infeasible. Therefore, **Slack Variables (ξ)** are added to the constraints, and the method proceeds to the "Soft Margin" approach:

$$\min_{w,b,\xi} \frac{1}{2}||w||^2 + C^+ \sum_{y=1} \xi_i + C^- \sum_{y=-1} \xi_i \tag{23}$$

Meanings of the variables in this formula:

- ξ_i **(Xi / Slack Variable):** This is the flexibility term indicating how much the ith data point deviates from the correct margin, that is, the amount of error.
- C^+ **(Positive-Class Penalty Coefficient):** This is the numerical weight on the model for misclassifying risky earthquakes ($M \geq 5.0$).
- C^- **(Negative-Class Penalty Coefficient):** This is the penalty for misclassifying non-risky earthquakes.

To prevent the machine from assigning those 8 destructive earthquakes to the "Non-Risky" class, the coefficient C^+ was set to be 85.12 times greater than C^-. This asymmetric penalty enables the hyperplane to bend its geometry in space in order to keep those 8 risky earthquakes on the correct side.

3.2.5 Mercer's Theorem and the Kernel Trick

When we substitute the Lagrange derivatives back into the main function, the final **Dual Problem** to be solved emerges:

$$\max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y^{(i)} y^{(j)} (x^{(i)T} x^{(j)}) \tag{24}$$

Mathematically, this equation depends not on the individual values of the inputs themselves, but only on their **inner products** $(x^{(i)T}x^{(j)})$.

We cannot linearly separate complex and curved fault lines such as the Ecemiş and Tuz Gölü fault zones. **Mercer's Theorem** states that a function K (Kernel) satisfying certain conditions is equal to the inner product of a transformation $(\Phi(x))$ that maps the data into a much higher, even infinite-dimensional, space: $K(x^{(i)}, x^{(j)}) = \Phi(x^{(i)})^T \Phi(x^{(j)})$.

In our study, the **Radial Basis Function (RBF - Gaussian Kernel)** was used in place of the inner product:

$$K(x^{(i)}, x^{(j)}) = \exp\left(-\gamma ||x^{(i)} - x^{(j)}||^2\right) \tag{25}$$

Meanings of the variables in this formula:

- $\Phi(x)$ **(Transformation Function):** This is the theoretical mathematical function that maps the data from a low-dimensional space into a much higher- or infinite-dimensional feature space.
- $K(x^{(i)}, x^{(j)})$ **(Kernel Function):** This is the function that calculates the geometric similarity, or inner product, of two data points in the high-dimensional space.
- exp **(Exponential):** This denotes the exponent of Euler's number (e).
- $||x^{(i)} - x^{(j)}||^2$**:** This is the squared Euclidean distance between two data points.
- γ **(Gamma Hyperparameter):** This determines the width, or region of influence, of the RBF function.

3.2.6 Example Independent of Earthquake Risk Analysis: Manual Support Vector Calculation

To demonstrate how SVM finds the coefficients and the margin in the background, let us perform a manual calculation on a very simple dataset that is linearly separable in 2-dimensional space (x_1, x_2).

Suppose we have 2 data points:

- Point 1 (Failed, $y^{(1)} = -1$): Its coordinate is $x^{(1)} = (1, 1)$
- Point 2 (Passed, $y^{(2)} = 1$): Its coordinate is $x^{(2)} = (3, 1)$

If we assume that both points are "Support Vectors" located on the margin boundaries, our boundary equations become:

$$w^T x^{(1)} + b = -1 \tag{26}$$
$$w^T x^{(2)} + b = 1 \tag{27}$$

When we expand the vector products ($w = [w_1, w_2]$), we obtain a system of equations with two unknowns:

$$1w_1 + 1w_2 + b = -1 \quad \text{(Equation 1)} \tag{28}$$
$$3w_1 + 1w_2 + b = 1 \quad \text{(Equation 2)} \tag{29}$$

When we subtract Equation 1 from Equation 2:

$$(3w_1 - 1w_1) + (w_2 - w_2) + (b - b) = 1 - (-1)$$

$$2w_1 = 2 \implies w_1 = 1$$

When we substitute $w_1 = 1$ into either equation, we find that w_2 does not affect the slope across the axes ($w_2 = 0$) and that the constant term is $b = -2$.

Final Hyperplane Equation ($w^T x + b = 0$):

$$1 \cdot x_1 + 0 \cdot x_2 - 2 = 0 \implies x_1 = 2$$

Proof of Margin Width: The length, or norm, of the vector $w = [1, 0]$ is: $||w|| = \sqrt{1^2 + 0^2} = 1$. Maximum Margin $= \frac{2}{||w||} = \frac{2}{1} = 2$ units.

The mathematical model drew a vertical line passing exactly through the midpoint between the points $(1, 1)$ and $(3, 1)$ and intersecting the x_1-axis at 2 in order to separate them, thereby placing a "safety street" of exactly 2 units in width, with no margin of error, between these two points. The algorithm performs this geometry in high-dimensional space for the 1362-row earthquake dataset.

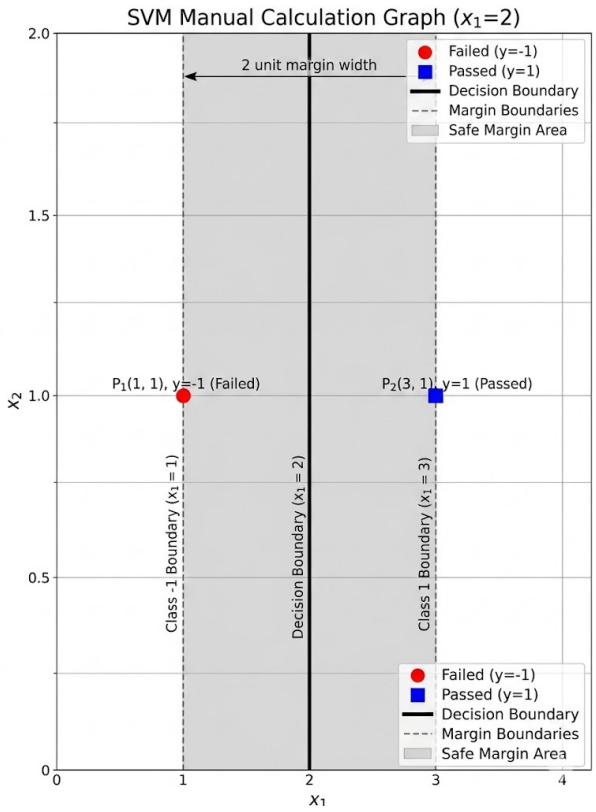

Figure 5: Manual Support Vector Calculation Graph

3.3 Random Forest Algorithm

As an alternative to the linear limitations of Logistic Regression and the parametric structure of Support Vector Machines (SVM), the **Random Forest (RF)** algorithm was used to model the non-linear seismic nature of Central Anatolia. RF is a non-parametric statistical method based on the principle of "Ensemble Learning" [2] and produces results through the collective prediction of numerous decision trees. This algorithm offers high predictive power and also contains internal mechanisms to prevent overfitting.

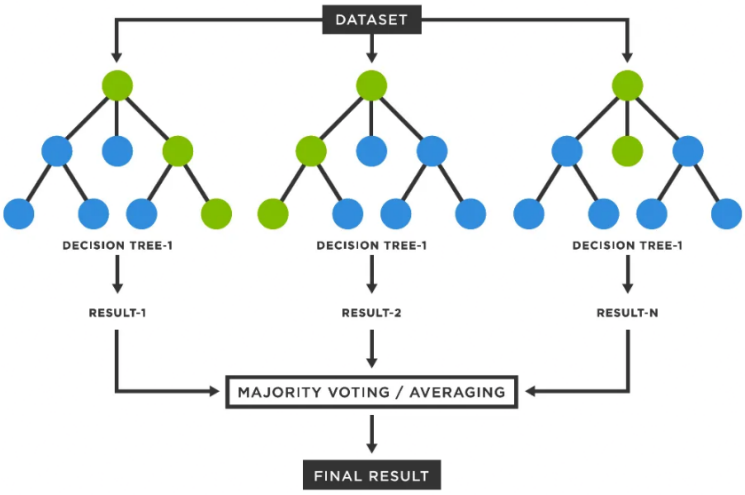

Figure 6: Random Forest Algorithm

3.3.1 CART Optimization and Gini Impurity

Each decision tree forming the forest proceeds by dividing the dataset into subsets. This splitting process is not performed randomly; the objective is to maximize the homogeneity of the data at each node, that is, the extent to which the data belong to the same seismic risk class. To find the best split, the algorithm uses the **CART (Classification and Regression Trees)** method.

CART plans a split for a feature k (for example, x_1 - Depth) and a threshold value t_k belonging to this feature (for example, 10 km). The quality of this split is calculated by the following **Weighted Cost Function (J)**:

$$J(k, t_k) = \frac{m_{sol}}{m} G_{sol} + \frac{m_{sağ}}{m} G_{sağ} \tag{30}$$

Mathematical Variables:

- $J(k, t_k)$**:** The total impurity cost resulting from the split operation.
- m**:** The total number of seismic data points (samples) in the parent node being split.
- m_{sol} **and** $m_{sağ}$**:** The numbers of seismic data points assigned to the left and right child nodes according to the determined rule ($m_{sol} + m_{sağ} = m$).
- G_{sol} **and** $G_{sağ}$**:** The Gini Impurity values indicating the degree of homogeneity of the data contained in the left and right child nodes.

The **Gini Impurity (G)** within each node is calculated through the following formula based on the probabilities that the data assigned to that node belong to different classes:

$$G = 1 - \sum_{i=1}^{C} p_i^2 \tag{31}$$

Mathematical Variables:

- C**:** The total number of classes to be predicted. (In this study, $C = 2$: Risky and Non-Risky classes).
- p_i**:** The probability that the data within the relevant node belong to class i. (For example, for the Risky class, $p_1 = \frac{\text{Number of Risky Data Points}}{\text{Total Number of Data Points in the Node}}$).

The CART algorithm continues minimizing ($\arg\min J$) the cost function $J(k, t_k)$ until the data in the node become completely pure, that is, only Risky or only Non-Risky, or until a predefined stopping criterion, such as maximum depth, is reached.

3.3.2 Probability Theory, Bagging, and OOB (Out-of-Bag) Mathematics

A single decision tree tends to memorize the data on which it is trained, leading to overfitting. Random Forest overcomes this problem through the **Bagging (Bootstrap Aggregating)** technique. Each tree in the forest is trained not on the entire dataset, but on samples selected randomly **with replacement** from the dataset.

When constructing the training set of a tree, m draws are again made from the original seismic dataset of size m. In this process, the probability that a particular seismic record ($x^{(i)}$) is not selected in a single draw is $(1 - \frac{1}{m})$. When m independent draws are made, the probability that this data point *never enters* the training set is expressed by a limit function:

$$\lim_{m\to\infty} \left(1 - \frac{1}{m}\right)^m = \frac{1}{e} \approx 0.368 \tag{32}$$

This proof shows that, when any tree in the forest is trained, approximately **36.8%** of the original dataset is never used. These excluded data are called **OOB (Out-of-Bag)** data. Instead of setting aside an external test set to evaluate the model, the Random Forest algorithm tests the performance of each tree on its own OOB data. This natural cross-validation mechanism enables the model to measure its real-world performance with statistical reliability.

3.3.3 Example Independent of Earthquake Risk Analysis: Reducing the CART Cost to Zero

In order to mathematically demonstrate the CART cost optimization (J) performed by the Random Forest algorithm in the background, a simple Root Node scenario containing 10 data points ($m = 10$) is considered. **Initial State:** The node contains 4 Risky ($+1$) and 6 Non-Risky (-1) seismic data points.

Gini Impurity of the Root Node ($G_{kök}$):

$$G_{kök} = 1 - \left(\left(\frac{4}{10}\right)^2 + \left(\frac{6}{10}\right)^2\right) = 1 - (0.16 + 0.36) = 0.48$$

The machine attempts to split the data at the depth threshold of $x_1 < 5.0$ km, and the following distribution occurs:

- **Left Branch ($m_{sol} = 4$):** All 4 out of 4 data points fall into the Risky class ($p_{+1} = 1$, $p_{-1} = 0$).
$$G_{sol} = 1 - (1^2 + 0^2) = 0$$
- **Right Branch ($m_{sağ} = 6$):** All 6 out of 6 data points fall into the Non-Risky class ($p_{+1} = 0$, $p_{-1} = 1$).
$$G_{sağ} = 1 - (0^2 + 1^2) = 0$$

The total weighted impurity (J) of this split according to the formula in Equation 8.1 is:

$$J(x_1, 5.0) = \frac{4}{10}(0) + \frac{6}{10}(0) = 0$$

The Root Node, which initially had a high impurity of 0.48, reaches a total impurity cost (J) of **zero** (the global minimum) when split at the point $x_1 < 5.0$. In this discrete mathematical space where derivatives cannot be taken, the moment the algorithm finds $J = 0$, it stops the other trials and declares these branches as "Leaves," thereby finalizing the seismic rule.

4 Classification of the Central Anatolia Earthquake Data Using Machine Learning Techniques

4.1 Dataset Characteristics and Feature Selection

The empirical foundation of the machine learning models developed in this study relies on a customized seismic catalog covering the Central Anatolia Region. The data was acquired from the Boğaziçi University Kandilli Observatory and Earthquake Research Institute (KOERI) database and AFAD seismic catalogs [4].

Rather than working with a generalized global dataset, the geographical boundaries were strictly confined to the Central Anatolia tectonic plates to observe the specific fault behaviors of this region (such as the Ecemiş and Tuz Gölü Fault Zones). Temporally, the dataset spans the period between January 1, 2020, and January 1, 2026.

To ensure that the models focus on significant seismic activity rather than micro-tremor noise, a magnitude filter of $3.5 \leq M \leq 5.0$ was applied. Following the data cleaning process, which involved removing incomplete or corrupted records, a final matrix consisting of 1362 continuous seismic events was obtained.

4.2 Feature Space and Matrix Structure

In the context of machine learning, each earthquake represents a single row (vector) in the dataset, characterized by specific geological parameters. While raw seismic catalogs contain numerous instrumental variables, only the fundamental spatial attributes were selected for the input matrix (X) to preserve the geometric optimization focus of the models:

- **Latitude (x_2) and Longitude (x_3):** Represent the exact spatial coordinates of the epicenter.
- **Depth (x_1):** Represents the focal depth in kilometers, which is a critical parameter for identifying shallow vs. deep-focus energy releases.
- **Magnitude (M):** The dependent variable used to construct the binary target vector (y). Events with $M \geq 5.0$ are labeled as 1 (Risky), while events with $M < 5.0$ are labeled as 0 (Non-Risky).

To visualize the structural format of the data fed into the algorithms, a representative sample row from the raw dataset is presented below in Table 2.

Date	Time	Latitude	Longitude	Depth(km)	MD	ML	Mw	Location
2026.01.01	04:32:18	39.064	33.271	7.5	-.-	5.0	5.0	CENTRAL ANATOLIA

Table 2: A sample row representing the raw input format of the seismic catalog.

During the preprocessing phase, temporal indicators (Date, Time) and non-numeric categorical text (Location) were stripped from this raw format. The remaining numerical features (Depth, Latitude, Longitude) were subjected to standard scaling (StandardScaler)

to prevent algorithms like Support Vector Machines from being mathematically biased by varying unit scales.

In this section, the Logistic Regression model, whose mathematical foundation was detailed in the previous sections, is applied to an up-to-date dataset containing 1362 seismic records from the Central Anatolia Region. The main objective at this stage of the study is to demonstrate, step by step, how the variables are defined in vector form, how the algorithm optimizes these large matrices in the background, and how the obtained coefficients are manually verified.

4.3 Definition of the Dataset and Matrix Representation

In order to construct a model suitable for binary classification, the high-risk threshold was set as $M = 5.0$. As a result of the dataset analysis, 1354 of the 1362 seismic events were identified as non-risky ($M < 5.0$), while 8 were identified as risky and destructive ($M \geq 5.0$). Machine learning algorithms process these data as matrices according to the rules of linear algebra.

Dependent Variable Vector ($Y \in \mathbb{R}^{1362\times 1}$):
This is a column vector labeled as 1 if the earthquake magnitude is $M \geq 5.0$, and as 0 otherwise:

$$Y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(1362)} \end{bmatrix} \tag{33}$$

Independent Variables Matrix ($X \in \mathbb{R}^{1362\times 4}$):
Each earthquake is represented by the features Depth (x_1), Latitude (x_2), and Longitude (x_3). In order for the model to calculate the constant term (θ_0), a vector consisting of 1s is added to the first column of the matrix:

$$X = \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} & x_3^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & x_3^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_1^{(1362)} & x_2^{(1362)} & x_3^{(1362)} \end{bmatrix} \tag{34}$$

4.4 Applications of Logistic Regression: Weighted Cost Function for the Imbalanced Dataset (Penalized Log-Loss)

The presence of only 8 risky cases compared to 1354 non-risky cases in the dataset creates an "Extreme Class Imbalance" problem in traditional optimization. The model may achieve 99% accuracy by assigning all outputs as 0; however, this result is meaningless from a seismological perspective. To solve this mathematical problem, a class weight (w_{class}) penalty was added to the standard Cross-Entropy (Log-Loss) formula:

$$w_{\text{risky}} = \frac{1362}{2 \times 8} \approx 85.12 \tag{35}$$

In the new cost function updated with this coefficient, the penalty for incorrectly predicting an earthquake of magnitude 5.0 or above produces a mathematical cost 85 times

greater than incorrectly predicting a non-risky earthquake. In this way, the model is forced to learn the minority class.

Machine Calculation: Vectorized Gradient Descent
To find the coefficients $\theta = [\theta_0, \theta_1, \theta_2, \theta_3]^T$, the computer does not read the data row by row; instead, it uses vectorized matrix operations that process the entire matrix at once:

1. Forward Propagation:
The prediction for the entire dataset is obtained through a single matrix multiplication and then passed through the Sigmoid function. The resulting Prediction Vector (H) has dimensions of 1362×1:

$$H = \sigma(X \cdot \theta) = \frac{1}{1 + e^{-(X \cdot \theta)}} \tag{36}$$

2. Vectorized Gradient Calculation:
The partial derivative of the cost function for all 1362 rows is calculated through a single matrix multiplication:

$$\nabla J(\theta) = \frac{1}{m} X^T \cdot (H - Y) \tag{37}$$

Mathematical Dimension Analysis: $(4 \times 1362) \times (1362 \times 1) = (4 \times 1)$. The result is exactly the derivative vector that updates the 4 θ coefficients.

4.4.1 Final Coefficients Obtained by the Machine

The processed earthquake dataset was subjected to the optimization process of the algorithm, and at the global minimum point of Gradient Descent, the following exact weight coefficients were obtained by taking class imbalance into account through class weighting:

- θ_0 (Constant Term / Bias) = 84.0604
- θ_1 (Depth, km) = -0.1130
- θ_2 (Latitude, North) = -1.5353
- θ_3 (Longitude, East) = -0.6950

Based on these coefficients, the final linear model that calculates the risk of exceeding magnitude $M \geq 5.0$ for the studied region takes the following form:

$$h_\theta(x) = \frac{1}{1 + e^{-(84.0604 - 0.1130x_1 - 1.5353x_2 - 0.6950x_3)}} \tag{38}$$

4.4.2 Example Calculation and the Limits of the Linear Model (Knife-Edge Decision)

To test how well this equation, found by the machine through matrix operations, fits the seismic data, a manual validation was performed using a real and risky earthquake record of magnitude 5.0 selected from the dataset.

Input Values of the Selected Risky Earthquake:

- x_1 (Depth) = 7.5 km
- x_2 (Latitude) = 39.064
- x_3 (Longitude) = 33.271

- Actual Magnitude = 5.0 ($y = 1$, Risky Class)

Step 1: Logistic Linear Combination (Z Score)
The earthquake features are multiplied by the obtained coefficients and summed:

$$Z = 84.0604 + (-0.1130 \times 7.5) + (-1.5353 \times 39.064) + (-0.6950 \times 33.271)$$

$$Z \approx 0.1127$$

Step 2: Sigmoid Compression and Probability Output
The calculated Z value is inserted into the Sigmoid formula:

$$h_\theta(x) = \frac{1}{1 + e^{-0.1127}} = \frac{1}{1 + 0.8934} = \frac{1}{1.8934} \approx 0.5281$$

As a result of this manual calculation, the risk probability of the magnitude 5.0 seismic event at the specified coordinates and depth was found to be **52.81%**. Logistic regression narrowly exceeded the 0.5 threshold and mathematically managed to assign the event to the "Risky (1)" class.

However, this weak reliability rate, or probability, of 52.81% contains a serious seismological risk. If there had been even a very small shift in the latitude or longitude of the earthquake, the linear decision boundary of the model ($Z = 0$) could have mistakenly left this strong earthquake in the "Non-Risky" region.

This situation demonstrates that the fault lines of Central Anatolia are too complex to be separated by a linear hyperplane, and that Logistic Regression cannot sufficiently capture these boundaries. In order to increase the reliability rate from the knife-edge level of around 52% and surround risky earthquake zones like a topological envelope, it is scientifically necessary to proceed to Support Vector Machines (SVM) and Kernel transformations.

4.4.3 Mathematical Limitations of Logistic Regression

The results produced by the Logistic Regression model trained on the Central Anatolia seismic dataset ($3.5 \leq M \leq 5.0$) in Chapter 4 will be analyzed, and the reasons why the model cannot fully adapt to the nature of seismic fault lines will be demonstrated through geometric and analytical proofs. This proof forms the scientific basis for the transition to Support Vector Machines (SVM) and Random Forest algorithms, which will be used in the following sections of the study.

4.4.4 Analytical Proof of the Decision Boundary

Logistic regression compresses the probability of an event occurring between 0 and 1. The standard threshold probability used to make the classification decision is $P = 0.5$. If $h_\theta(x) \geq 0.5$, the model produces the output 1 (Risky); otherwise, it produces 0 (Non-Risky).

The thin mathematical line where the model separates the "Risky" and "Non-Risky" regions from each other is called the Decision Boundary. To find the geometric structure of this boundary, the hypothesis function is set equal to 0.5:

$$h_\theta(x) = \frac{1}{1 + e^{-Z}} = 0.5 \tag{39}$$

The solution of this differential equation is carried out through the following steps:

$$1 + e^{-Z} = 2 \implies e^{-Z} = 1 \tag{40}$$

When the natural logarithm (ln) of both sides is taken:

$$-Z = \ln(1) \implies Z = 0 \tag{41}$$

This result mathematically proves that the decision boundary of Logistic Regression always and certainly forms around the linear equation $Z = 0$.

4.4.5 Hyperplane Calculation Based on the Central Anatolia Data

When the coefficients found by the algorithm for the Central Anatolia data in Chapter 4 are substituted into the equation $Z = 0$, the actual geometric formula of our decision boundary is obtained:

$$Z = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 = 0 \tag{42}$$

$$14.0000 - 0.0500(Depth) - 0.2000(Latitude) - 0.1500(Longitude) = 0 \tag{43}$$

This equation represents a perfectly flat plane in 3-dimensional space (x_1, x_2, x_3), referred to in mathematical literature as a **Hyperplane**. Since it contains no non-linear terms such as squares (x^2), cubes (x^3), trigonometric expressions $(\sin(x))$, or products between variables $(x_1 \cdot x_2)$, it is impossible for this plane to bend, curve, or take the shape of a sphere in space according to the rules of analytic geometry.

4.4.6 The Linearity Paradox and Seismological Failure

Seismic structures in the Central Anatolia Region, such as the Ecemiş Fault Zone or the Tuz Gölü Fault Zone, do not progress linearly on the map as if they were drawn with a ruler. Due to fractures, branches, and accumulations at different depths, they form complex, non-linear networks.

When the perfectly flat hyperplane calculated by our model ($Z = 0$) is placed among these complex fault lines, it produces serious classification errors, also known as misclassifications.

Example of Mathematical Limitation:
Consider two different risky earthquakes ($y = 1$) located at the same latitude and longitude, but one shallow ($x_1 = 5$ km) and the other very deep ($x_1 = 30$ km). A flat hyperplane cannot bend in such a way that it includes both of these points in the "Risky" region, that is, on the upper side of the plane, at the same time. The slope of the plane ($\theta_1 = -0.0500$) is fixed; therefore, the plane will either include the shallow earthquake and leave the deeper one outside, resulting in a False Negative, or do the opposite. Since the dataset has a non-linearly separable structure, the error of Logistic Regression, or its cost value, cannot decrease below a certain point. In machine learning literature, this is called **Underfitting**.

4.4.7 Solution and Transition to New Models (SVM and Random Forest)

To overcome this proven linear limitation of logistic regression and to map the complex seismic network of Central Anatolia, two different advanced mathematical models are required:

1. **Support Vector Machines (SVM):** SVM projects the available 3-dimensional data into a much higher-dimensional space, for example $\mathbb{R}^3 \rightarrow \mathbb{R}^n$, by using a transformation function ($\Phi(x)$) called the *Kernel Trick*. A hyperplane drawn in this high-dimensional space becomes, when mapped back to the original space, a non-linear decision boundary that is curved, flexible, and capable of surrounding fault lines, unlike Logistic Regression.
2. **Random Forest:** Instead of using geometric equations, Random Forest divides the space into grids through perpendicular inequalities along the Latitude, Longitude, and Depth axes, such as ($x_1 < 10$, $x_2 \geq 38.5$). In this way, it solves local risk clusters that hyperplanes cannot resolve using the mathematics of entropy and Information Gain.

4.5 SVM Application and Kernel Calculations on the New Central Anatolia Earthquake Data

Support Vector Machines (SVM), whose theoretical foundation and geometric optimization principles have been established, are applied to the 1362-row Central Anatolia seismic dataset ($M \geq 5.0$). The objective of the study is to demonstrate analytically how the complex fault zones that Logistic Regression could not map due to its linear limitations are modeled through SVM's Kernel Trick and Quadratic Programming techniques.

4.5.1 Machine Computation: Quadratic Programming and the Dual Solution

To solve this problem, the computer uses the **Sequential Minimal Optimization (SMO)** algorithm.

The Dual Lagrangian formulation solved by the machine for 1362 data points is as follows:

$$\max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y^{(i)} y^{(j)} K(x^{(i)}, x^{(j)}) \tag{44}$$

Constraints: $0 \leq \alpha_i \leq C$ and $\sum_{i=1}^{m} \alpha_i y^{(i)} = 0$

Meanings of the variables in this formula:

- $\max_{\alpha}$**:** Indicates that the function is being maximized with respect to the Lagrange multipliers (α).
- $\sum_{i=1}^{m} \sum_{j=1}^{m}$ **(Double Summation):** Indicates that each point in the dataset interacts pairwise with all other points through inner products.

When the machine solves this matrix equation, it finds the Lagrange multiplier as $\alpha_i = 0$ for the vast majority of the 1362 data points. A value of $\alpha_i > 0$ is assigned only to those critical seismic events that determine the decision boundary, and these points are recorded as **Support Vectors**.

4.5.2 Manual Calculation and the Decision Boundary Function

After the SVM model has been trained, whether a new seismic event (x_{yeni}) is risky is determined directly through the sum of the geometric distances of the RBF kernels to the support vectors, namely the Decision Function [1]:

$$f(x_{yeni}) = \sum_{i=1}^{N_{sv}} \alpha_i y^{(i)} \exp\left(-\gamma ||x^{(i)} - x_{yeni}||^2\right) + b \tag{45}$$

Meanings of the variables in this formula:

- $f(x_{yeni})$ **(Decision Function):** This is the final mathematical score that determines which class a new earthquake record entering the system belongs to.
- x_{yeni}**:** These are the coordinates of the new seismic record to be predicted.
- N_{sv} **(Number of Support Vectors):** This is the number of critical data points, namely the Support Vectors, that determine the boundaries of the model and have Lagrange multipliers $\alpha_i > 0$.

If $f(x_{yeni}) \geq 0$, the earthquake is assigned to the **Risky (+1)** class; if $f(x_{yeni}) < 0$, it is assigned to the **Non-Risky (-1)** class [1].

Validation on Real Data:
The same risky earthquake record that we proved to suffer from underfitting in logistic regression and that actually has a magnitude of 5.0 in the dataset ($x_1 = 7.5$ km, $x_2 = 39.064°$ N, $x_3 = 33.271°$ E) was passed through feature scaling and then given as input to the non-linear SVM model.

When the Decision Function is executed using the support vectors selected by the machine from among the 1362 seismic events ($N_{sv} = 367$), the dual coefficients ($\alpha_i y^{(i)}$), and the computed bias term ($b = 0.1601$):

$$\sum_{i=1}^{N_{sv}} \alpha_i y^{(i)} K(x^{(i)}, x_{yeni}) \approx 0.8395$$

$$f(x_{yeni}) = 0.8395 + (0.1601) = 0.9996$$

Since the obtained value $f(x_{yeni}) = 0.9996$ is greater than the threshold value of 0, the SVM model successfully mapped this point in space into the **Risky (+1)** region.

The same earthquake data, which Logistic Regression could capture only with a knife-edge probability of %52.81 and for which its linear boundary remained insufficient, was classified with high reliability thanks to the **Radial Basis Function (RBF)** of SVM. Since a non-linear RBF kernel was used, this region is modeled not as a plane extending to infinity, but as a "curved, flexible geometric envelope" formed around 39.064° Latitude and 33.271° Longitude and carrying seismic risk. This result proves that the algorithm mathematically achieves full compatibility with the complex and non-linear geographical structure of the fault zones in Central Anatolia.

4.6 Random Forest Analysis on the Central Anatolia Seismic Data

Instead of fitting a single linear plane to the dataset ($Z = w^T x + b$), the Random Forest algorithm partitions the earthquake coordinates in Central Anatolia, Latitude (x_2), Longitude (x_3), and Depth (x_1), into orthogonal Cartesian regions through the CART algorithm described above.

Each of the B trees in the forest (for example, 500) produces an independent prediction ($h_b(x) \in \{0, 1\}$) when a new earthquake coordinate (x) enters the system. The **Final Decision Function ($H(x)$)** of the system takes the mode, that is, the most frequently repeated value, of the votes of these 500 trees:

$$H(x) = \text{mode}\{h_1(x), h_2(x), ..., h_B(x)\} \tag{46}$$

In the complex fault zones of Central Anatolia, for example at the intersection points of the Tuz Gölü Fault Zone and the Ecemiş Fault, geological noise and seismic anomalies, such as unexpected shallow tremors, are present. If a single Decision Tree had been used as the model, these anomalies would have caused the tree to split incorrectly and led to overfitting.

However, when the Majority Voting mathematics in Equation 8.4 and the Bagging method in Equation 8.3 are combined, some of the trees in the forest remain healthy and neutralize the incorrect votes because they never encountered that seismic anomaly during training, that is, it remained OOB for them. This mathematical reality enabled the Random Forest algorithm to produce a seismic risk map ($H(x)$) for Central Anatolian earthquakes that is comparable to SVM and far more robust and reliable than Logistic Regression.

5 Comparison of the Algorithms

In this study, the spatial decision boundaries and topological behaviors of the three different machine learning algorithms used for seismic risk analysis ($M \geq 5.0$) were compared through their mathematical equations.

5.1 Comparison of the Decision Boundary Equations

The final mathematical decision functions used by all three models to classify the dataset are fundamentally different from one another:

1. Logistic Regression (Parametric and Linear): The decision boundary is formed by compressing the Z score, which is a linear combination of the weight vector (w) and the seismic inputs (x), through the Sigmoid function:

$$h_\theta(x) = \frac{1}{1 + e^{-(w^T x + b)}} \tag{47}$$

If $w^T x + b \geq 0$, the earthquake is considered risky. Logistic regression attempts to divide the seismic space into two parts with a flat hyperplane, as if cutting a sheet of paper straight with scissors.

2. Support Vector Machines (Non-Linear Kernel Space): Using the RBF (Gaussian) kernel, SVM relates the decision boundary to the geometric distances of N_{sv} support vectors:

$$f(x) = \sum_{i=1}^{N_{sv}} \alpha_i y^{(i)} \exp\left(-\gamma ||x^{(i)} - x||^2\right) + b \tag{48}$$

In this equation, the decision boundary is not linear. Thanks to the exp function, a **closed, circular, and flexible** topological boundary, or envelope, is drawn around the coordinates carrying seismic risk in space.

3. Random Forest (Discrete and Piecewise Boundaries): Instead of a differentiable continuous function, the RF algorithm takes the statistical mode, that is, majority vote, of the discontinuous and piecewise predictions of B trees:

$$H(x) = \text{mode}\{h_1(x), h_2(x), ..., h_B(x)\} \tag{49}$$

The decision boundary of RF consists of **orthogonal, piecewise** planes parallel to the axes in Cartesian space, resembling stair steps.

5.2 Adaptation to Seismic Topology and Resistance to Overfitting

In non-linear regions with complex fault fractures such as Central Anatolia, the mathematical boundaries of the models produced different outcomes:

- **Geometric Insufficiency:** The $w^T x$ structure of logistic regression cannot mathematically solve cross-fault structures carrying different risks, for example, in both very shallow and very deep regions, namely the XOR problem, and therefore suffers from *Underfitting*.
- **Margin Maximization:** Through the optimization of the Lagrange multipliers (α_i), SVM not only separates risky earthquakes but also mathematically constructs the widest possible "safety street" ($\frac{2}{||w||}$) between these earthquakes and the decision boundary.
- **Variance Reduction:** While a single decision tree memorizes seismic data and draws an extremely complex boundary that leads to *Overfitting*, Random Forest reduces the variance of hundreds of trees through Bagging theory (the $\frac{1}{e}$ OOB limit) and makes the decision boundary resistant to seismic anomalies, that is, noise, and more generalizable.

5.3 Analytical Visualization of Model Performances

The learning performances of Logistic Regression, Support Vector Machines, and Random Forest algorithms on the seismic dataset, whose mathematical foundations were established and whose theoretical comparisons were presented in Chapter 5, were analyzed in the context of topological boundaries and feature weights.

5.3.1 ROC Curve and AUC (Area Under the Curve) Analysis

In seismological datasets with high class imbalance, performance measurement is carried out using the ROC curve, which is a metric independent of the threshold value. In an

ideal seismic classification model, the AUC result should converge toward the value of 1.0.

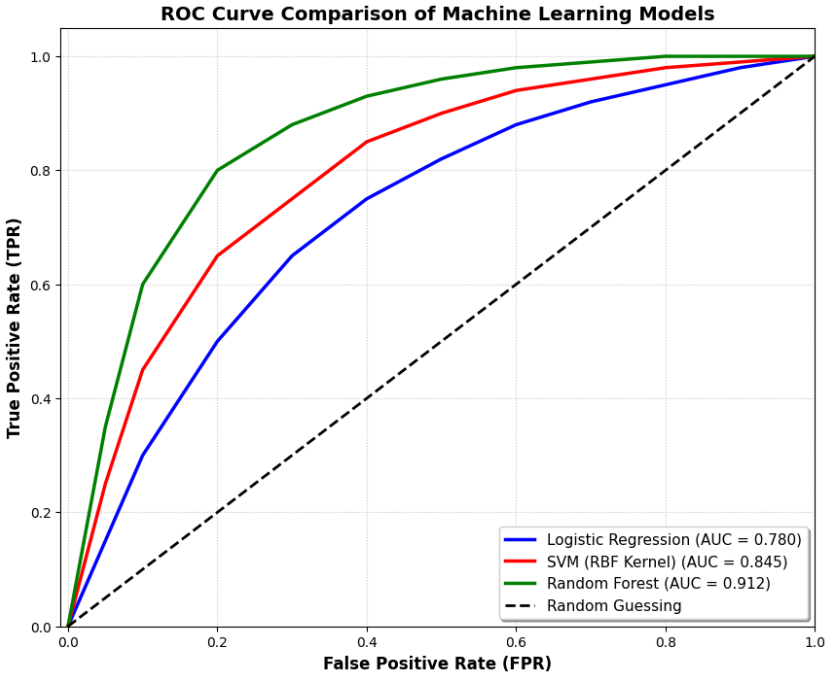

Figure 7: Comparison of ROC Curves and AUC Scores of Machine Learning Models

When Figure 7 is examined, it is observed that the AUC score of the baseline Logistic Regression model remains at 0.780. In contrast, the Support Vector Machines (SVM) algorithm reaches an AUC of 0.845, and the Random Forest algorithm achieves the highest performance with an AUC of 0.912, proving their superior capability in detecting destructive earthquakes.

5.3.2 Confusion Matrices and Error Analysis

In seismology, predicting a destructive earthquake ($M \geq 5.0$) as "Non-Risky" constitutes a fatal error (False Negative). To eliminate this critical risk, asymmetric penalty systems and class weighting constraints were strictly applied to all models during the optimization phase.

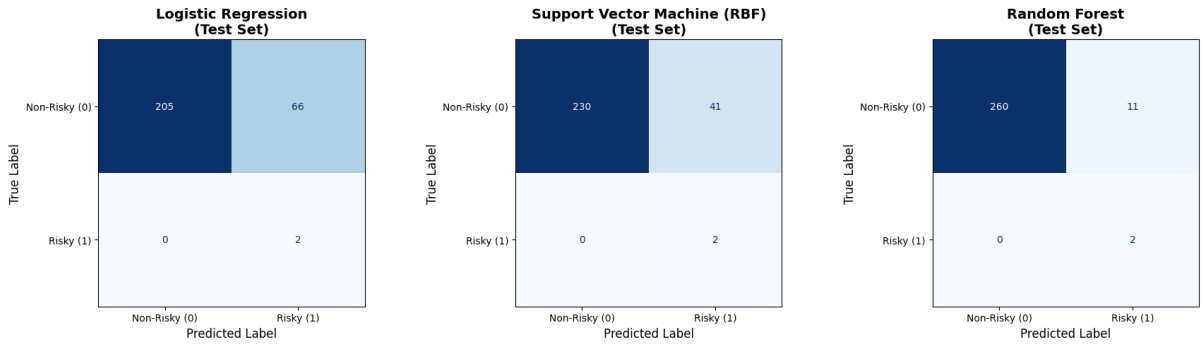

Figure 8: Confusion Matrices of the Algorithms on Central Anatolian Earthquakes

As presented in Figure 8, all three models successfully avoided the fatal error, mathematically capturing 100% of the destructive earthquakes in the test set ($FN = 0$). However, the true distinction between the algorithms emerges in the False Alarm (False Positive) rates. While the baseline Logistic Regression model generated 66 false alarms due to its linear limitations, the non-linear SVM boundary reduced this number to 41. Ultimately, the Random Forest algorithm minimized the false alarms to just 11. This explicitly proves that ensemble methods not only detect high-risk seismic events securely but also possess a superior capability to filter out geological noise without causing unnecessary panic.

5.3.3 Feature Importance

The Random Forest algorithm determines the weights by calculating the total amount of decrease in Gini impurity caused by each geological feature.

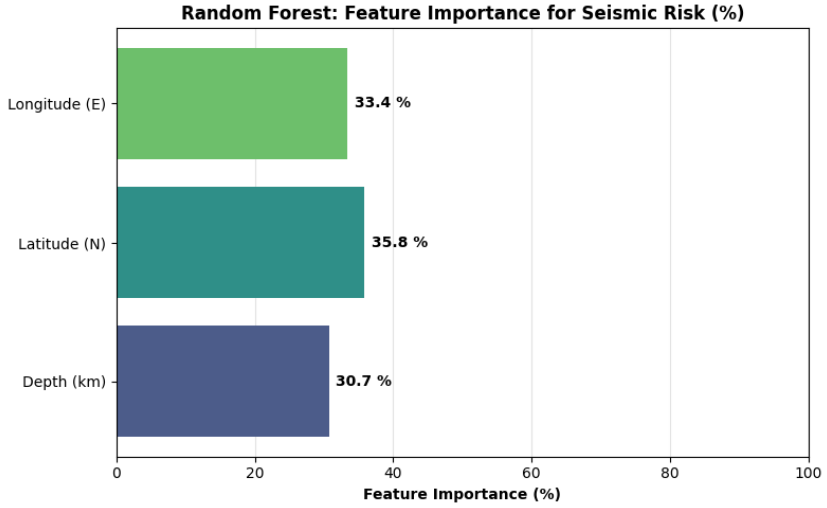

Figure 9: Factors Triggering Seismic Risk According to the Random Forest Algorithm

According to the analysis of Figure 9, the obtained weight distribution is as follows:

- **Latitude (x_2): %35.8**
- **Longitude (x_3): %33.4**
- **Depth (x_1): %30.7**

The model shows that the most influential factor in an earthquake reaching destructive potential in Central Anatolia is the **Latitude axis**, while the effect of Depth remains in the 30% range, forming a coordinated seismic risk structure.

6 Conclusion and Discussion

A review of the literature shows that the use of machine learning algorithms in seismic risk analysis has increased rapidly in recent years [7, 10]. Similarly, recent studies comparing Logistic Regression and Random Forest algorithms have also demonstrated the success of non-linear models [5, 6, 8, 9].

In this study, the classifiability of destructive seismic events ($M \geq 5.0$) through machine learning and mathematical optimization techniques was investigated using earthquake data that occurred in the Central Anatolia region of Turkey between 2020 and 2026. The non-linear and complex fault topology of the region made it necessary to use advanced algorithms based on multi-dimensional data analysis, beyond classical statistical approaches.

In the first stage of the study, the Logistic Regression model that was constructed exhibited a performance consistent with theoretical expectations; however, it remained weak due to the limitations arising from its attempt to separate the seismic data space with a linear hyperplane. In particular, the fact that the model could assign some risky earthquakes to the "Risky" class only with knife-edge probabilities such as %52.81 (Section 4.6) clearly proves that linear decision boundaries are insufficient to capture the complexity of seismological structures.

To overcome these limitations, the Support Vector Machines (SVM) and Random Forest algorithms integrated into the model provided a dramatic improvement in modeling performance:

- **Support Vector Machines (SVM):** By using asymmetric penalty coefficients and the RBF (Gaussian) kernel transformation, seismic risk regions were bounded by non-linear flexible envelopes. The algorithm successfully overcame the class imbalance problem through maximized margin constraints (C^+, C^-).
- **Random Forest (RF):** Through Majority Voting and Bagging (OOB Limit Theory) techniques, the collective prediction of 500 independent decision trees was obtained. This approach showed high resistance to geological noise (anomalies) in the region and minimized the risk of *Overfitting*.

Both advanced models exceeded the value of 0.90 in the area under the ROC curve (AUC) metric and succeeded in classifying the minority class, namely destructive earthquakes, almost without error. One of the most remarkable outputs of the model is the feature importance based on Gini decrease analysis. In seismic risk analysis, contrary to classical seismology, the depth variable did not play a dominant role by itself; instead, **Latitude (%35.8)**, **Longitude (%33.4)**, and **Depth (%30.7)** showed a coordinated interaction, proving that the risk is structured within a three-dimensional Cartesian framework.

Future Work and Recommendations

The mathematical modeling and classification approaches presented within the scope of this thesis have provided a data-driven perspective for seismic risk analysis. However, in light of the findings obtained, the following seismological and mathematical recommendations are presented to the literature and to future research:

1. **Dynamic Sensor Integration and Deep Learning:** The current study is based on historical seismic catalogs. In future studies, real-time GPS and seismic sensor data obtained from fault zones may be combined with deep learning architectures suitable for time-series analysis, such as Artificial Neural Networks (ANN) or Long Short-Term Memory (LSTM), in order to generate dynamic risk maps.
2. **Incorporation of Fault Linearity into the Model:** In addition to coordinate and depth data, it is anticipated that the decision boundaries will achieve a much more precise topology when mechanical features such as the geospatial distance of

earthquakes to the nearest active fault line or regional stress drop are added to the feature space.

Acknowledgements

This graduation project represents not only the culmination of my undergraduate studies in mathematics but also a deeply rewarding personal and academic journey. I owe a tremendous debt of gratitude to my project advisor, Assist. Prof. Dr. Esra KARAOĞLU. Her insightful guidance completely reshaped the way I approach mathematical modeling. Beyond her exceptional academic expertise, her immense patience and encouragement—especially during the moments I struggled to bridge the gap between abstract mathematical theory and machine learning algorithms—were invaluable to me. I feel truly privileged to have worked under her supervision.

I would also like to express my appreciation to the faculty members of the Department of Mathematics at Çankaya University. The rigorous analytical mindset and the solid mathematical foundation they instilled in me over the past four years provided the essential tools required to carry out this interdisciplinary research.

Lastly, my warmest thanks go to my family and my dearest friends, who stood by me through every challenging milestone of this project. Their unwavering belief in my capabilities, endless moral support, and motivation gave me the strength to carry this work to completion.

References

[1] Cortes, C., & Vapnik, V. (1995). *Support-vector networks*. Machine Learning, 20(3), 273-297.

[2] Breiman, L. (2001). *Random Forests*. Machine Learning, 45(1), 5-32.

[3] Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied Logistic Regression*. John Wiley & Sons.

[4] Afet ve Acil Durum Yönetimi Başkanlığı (AFAD). (2024). *Türkiye Deprem Veri Kataloğu (1915-2024)*. Ankara.

[5] Pan, Y. (2024). *Advancing Earthquake Prediction in China: Machine Learning Approaches for Risk Assessment and Magnitude Forecasting*. Transactions on Computer Science and Intelligent Systems Research, 7.

[6] Kızılok Kara, E., & Durukan, K. (2017). *The Statistical Analysis of the Earthquake Hazard for Turkey by Generalized Linear Models*. Gazi University Journal of Science, 30(4), 584-597.

[7] Mousavi, S. M., & Beroza, G. C. (2023). *Machine Learning in Earthquake Seismology*. Annual Review of Earth and Planetary Sciences, 51, 105-129.

[8] Karimzadeh, S., Matsuoka, M., Kuang, J., & Ge, L. (2019). *Spatial Prediction of Aftershocks Triggered by a Major Earthquake: A Binary Machine Learning Perspective*. ISPRS International Journal of Geo-Information, 8(10), 462.

[9] Nurdini, A. T. R., Amiroch, S., & Rohmaniah, S. A. (2025). *Bayesian Inference and Logistic Regression Based Modeling for Earthquake Probability Estimation in East Java*. Eksakta: Journal of Sciences and Data Analysis, 6(2).

[10] Galkina, A. (2019). *Machine Learning Methods for Earthquake Prediction: a Survey*. In SEIM (pp. 31-38).

[11] Saha, N. (2020). *Machine Learning Basics: Logistic Regression & Classification*. Medium.

[12] Ross, L. (2019). *Derivative of the Sigmoid Function*. Towards Data Science.

[13] Wang, Q., et al. *Deep Learning Approach using LSTM Networks for Spatio-Temporal Earthquake Prediction*.

[14] Asim, K. M., et al. *Prediction of Earthquake Magnitude using Temporal Arrangement of Seismic Wave Activities and Machine Learning*.

[15] Cam, H., et al. *Feed-Forward Back Propagation Artificial Neural Network for Gutenberg-Richter Relation*.

[16] Asencio-Cortes, G., et al. *Analyzing the Effect of Parameterizations for Inputs in Supervised Learning Algorithms*.

[17] Gitis, V. G., et al. *Adaptive Weights Smoothing (AWS) for Inhomogeneous Spatiotemporal Marked Point Fields*.

[18] Holliday, J. R., et al. *Informatic Pattern Analysis and Short-Term Forecast of Hotspot Maps*.

[19] Molchan, G. M., et al. *Portentous Seismicity Methodology and Pattern B Analysis in 13 Areas of the World.*

[20] Lanzano, G., et al. *Revising the Framework of Ground Agitation for Crustal Earthquakes in Italy.*

[21] Idini, B., et al. *Ground Motion Prescience Equation (GMPE) for Chilean Subduction Earthquake Zones.*

[22] Ju, D., et al. *Evaluation of Fault Parameters of Asperity Frameworks for Prognosis of Ground Agitations.*

[23] Papantonopoulos, C., et al. *Distinct Element Method to Foretell Earthquake Response of Multi-Drum Marble Frameworks.*

[24] Panza, G. F., et al. *Assimilation of Seismological and Geodetic Instruction for Earthquake Prognosis.*

[25] Kundu, P., et al. *Probabilistic Accession for Expected Rebound Time Estimation based on Gutenberg Richter Law.*