İSTANBUL UNIVERSITY PRESS

Acta Infologica
acin 2023, 7 (2), 360–383
DOI: 10.26650/acin.1394019

Research Article / Araştırma Makalesi

The Efficiency of Regularization Method on Model Success in Issue Type Prediction Problem

Sorun Türü Tahmini Probleminde Düzenlileştirme Yönteminin Model Başarısı Üzerindeki Etkisi

Ali Alsaç[1], Mehmet Mutlu Yenisey[2], Murat Can Ganiz[3], Mustafa Dağtekin[4], Taner Ulusinan[5]

[1](M.Sc.), Istanbul University-Cerrahpasa, Institute of Graduate Studies, Department of Industrial Engineering, Istanbul, Turkiye
[2](Prof. Dr.), Istanbul University-Cerrahpasa, Faculty of Engineering, Department of Industrial Engineering, Istanbul, Turkiye
[3](Assoc. Prof. Dr.), Marmara University, Faculty of Engineering, Department of Computer Engineering, Istanbul, Turkiye
[4](Dr.), Istanbul University-Cerrahpasa, Faculty of Engineering, Department of Computer Engineering, Istanbul, Turkiye
[5]Private company, Istanbul, Turkiye

Corresponding author : Ali ALSAÇ
E-mail : ali.alsac@ogr.iuc.edu.tr

Submitted : 22.11.2023
Revision Requested : 30.11.2023
Last Revision Received : 30.11.2023
Accepted : 01.12.2023
Published Online : 14.12.2023

ABSTRACT

Designing a prediction method with machine learning algorithms and increasing the prediction success is one of the most important research areas and aims of today. Models designed using classification algorithms are frequently used especially in problem types that require prediction. In this study, real life data is used to answer the question of which problem type should be included in the Information Technology Service Management (ITSM) system. An important step in the search for a solution is to examine the dataset with regularization methods. Experimental results have been obtained to establish the overfitting or underfitting balance of the dataset with L1 and L2 regularization methods. While the Root-Mean-Square Error (RMSE) value was approximately 0.13 in the regression model without regularization, this value was found to be approximately 0.083 after L1 regularization. With the regularized dataset, new results were obtained using Artificial Neural Network (ANN), Logistic Regression (LR), Support Vector Machine (SVM) classifier algorithms. SVM algorithm was the most successful model with a performance of approximately 0.73. It is followed by LR and ANN respectively. Accuracy, Precision, Recall and F1Score were used as evaluation metrics. It is seen that the use of regularization methods, especially in the preparation of real-life data for use in machine learning or other artificial intelligence research, will contribute to increasing the success level of the model.

Keywords: IT service management, regularization, prediction, classification

ÖZ

Matematik düzleminde bir tahmin yöntemi tasarlamak ve başarılı sonuçlarından faydalanmak günümüzün önemli araştırma alanlarından ve amaçlarından biri olarak öne çıkmaktadır. Sınıflandırma algoritmaları kullanılarak tasarlanan modeller özellikle tahmin gerektiren problem türlerinde sıklıkla kullanılmaktadır. Çalışmada gerçek hayat verileri kullanılarak bir gerçek hayat problemi olan müşteriden gelen çözüm talebinin Bilgi Teknolojisi Hizmet Yönetimi (BTHY) sistemi içinde hangi sorun tipine dahil edilmesi gerektiği sorusuna cevap aranmaktadır. Çözüm arayışının önemli bir aşamasında veri kümesinin Regülarizasyon yöntemleri ile incelenmesi yer almaktadır. L1 ve L2 regülarizasyon yöntemleri ile veri kümesinin overfitting ya da underfitting dengesinin kurulması için deneysel sonuçlar alınmıştır. Regülarizasyon uygulanmamış regresyon modelinde Kök Ortalama Kare Hatası (RMSE) değeri yaklaşık olarak 0,13 iken L1 regülarizasyonu sonucunda bu değer yaklaşık 0,083 olarak bulunmuştur. Düzenlileştirilmiş veri kümesi ile Yapay Sinir Ağları (YSA), Lojistik Regresyon (LR), Destek Vektör Makinaları (DVM) sınıflandırıcı algoritmaları kullanılarak yeni sonuçlar elde edilmiştir. DVM algoritması yaklaşık 0,73 başarım sonucu ile en başarılı model olmuştur. Sırasıyla LR ve YSA takip etmektedir. Değerlendirme metrikleri olarak Accuracy, Precision, Recall ve F1Score kullanılmıştır. Özellikle gerçek hayat verilerinin makina öğrenmesi ya da diğer yapay zeka araştırmalarında kullanımı için hazırlanması aşamasında Regülarizasyon yöntemlerinden faydalanmanın modelin başarı düzeyinin artmasında katkısı olacağı görülmektedir.

Anahtar Kelimeler: Bilgi işlem servis yönetimi, regülarizasyon, tahmin, sınıflandırma

1. INTRODUCTION

The way for businesses, public institutions and organizations that produce goods or services to fulfill their activities faster has been opened with the participation of computers in business management. Production through computer-aided systems has facilitated the control of costs and increased the efficiency of production management. The increase in the transaction speed of service-producing enterprises has enabled them to work with more customers and increased their transaction volumes. According to TUIK 2021 data, 43.1% of the enterprises active in 2021 were in the service sector and 36.5% in the trade sector, while the service sector accounted for 37.4% of total employment in employment, while the employment share of the industrial sector was 28.9% (YENİSU, 2021). Information and Communication Technologies, which is one of the leading service sector areas, is the main area analyzed in this article.

In the study titled "Information and Communication Technologies Sector 2021 Market Data" conducted by Delloitte for TUBISAD, the predictions for the future of the IT sector are as follows (Deloitte & TUBISAD, 2022);

- Geopolitical and economic uncertainties and supply chain disruptions will increase spending on flexible and agile solutions
- As the focus on digitalization increases, companies position digital transformation as a strategic priority
- Increasing interest in emerging technologies such as analytics, cloud computing, digital customer experience and security solutions
- Remote working is becoming permanent and more than 50% of total employees are expected to switch to remote working in 2024

As can be understood from the items listed above, the importance of customer service will increase in many sectors. The permanence of remote working, which has become widespread with the Covid-19 pandemic, has also enabled the IT sector to expand its sphere of influence. Information technologies realized a growth of approximately 25% in the 2020-2021 period (Deloitte & TUBISAD, 2022).

Information technologies consist of hardware, software and service applications. Service applications include the following components (Deloitte & TUBISAD, 2022).

- Outsourcing services
- Consultancy services
- Development, integration, installation and operation services
- Support, care and training services

In this paper, we examine regularization models used to improve the success level of methods and models that seek solutions to the problem of assigning customer requests to the most appropriate expert. Assigning customer requests to the most appropriate consultant with the help desk application enables the control or reduction of time-related costs. The core of the study is to weight the attributes that affect the performance of the assignment problem in the model with machine learning methods on the data set to be explained in the following sections, or to penalize the attributes that do not affect the solution or cause the model to overlearn.

Customer satisfaction will increase if the customer's request reaches the right consultant in the fastest way possible. Currently, the customer service approach tries to increase service quality by categorizing customers in classical business models according to transaction volume, business nature, frequency or direct strategic importance for the business. However, this method of work allocation does not meet the quest to increase the speed and quality of service provided by finding the most appropriate consultant for the incoming demand.

The overlearning of the above-mentioned model is generally defined as "overfitting" in the literature. The concept of model is the broadest definition that covers all the methods, definitions, evaluations and explanations used to address and solve the problem. The concept of hyperparameterization also stands out as a very important concept in machine learning models. The negativities caused by overlearning of the model and hyperparameters in this context are discussed in the second section.

In the first part of the study, the literature is presented. In the second section, the methodology of the study is discussed and in the third section, the real-life data of a company operating in the IT sector are evaluated in an experimental environment using the machine learning algorithms described in the methodology and the success of the model is examined. The fourth section includes a discussion section where the results obtained from the model are evaluated. The fifth and final section presents the results obtained within the subject integrity of the article. The paper concludes with an area where the authors express their gratitude to the institutions and individuals who have supported their scientific work, followed by a bibliography.

2. RELATED WORK

In the IT sector, the use of criteria such as education, work experience, availability in working hours, age, language skills, salary group, etc. in the assignment of experts to customers constitutes the essence of the model designed for the problem. In order to increase the success level of the model, regularization is used as an experimental method. In the literature, the two cases mentioned above, namely expert assignments in the IT sector and regularization methods, are examined in academic databases. In Web of Science and Google Scholar databases, the keywords "expert recommendation", "issue classification", "regularization", "help desk" are searched in relation to the keywords "Machine Learning" and "Artificial Intelligence" and sample studies on the topics studied in the article are selected. When a search was made by selecting All Fields, 1665 publication results were obtained. When Open Access was selected, the number decreased to 663. When WoS Index is selected as SCI-EXPANDED and CPCI-S, publication date between 2008-2023, the number of publications is 512. Citation Topics Meso; Artificial Intelligence and Machine Learning, Knowledge Engineering, Software Engineering, Numerical Methods were selected to increase the precision of the search and thus the number of publications was simplified to 138. Web of Science Categories; Computer Science Artificial Intelligence, Computer Science Information Systems, Mathematics, Computer Science Software Engineering were selected and the number of publications was simplified to 85. In this paper, the use of regularization techniques to improve the performance of algorithms used in classification problems will be evaluated with experimental results. Learning the notations of regularization methods and examining their applications will be sufficient to understand the technique.

Figure 1. *Sources selected for the article*

As a result of the literature review on the classification of demands from customers, which is shown in Figure 1, some of the articles that are suitable for the content of the article are described below.

This paper is a follow-up to a conference paper published in 2022, which used data from a company operating in the ITSM sector to estimate the time it takes to resolve tickets to an expert. 16970 data sets were used in this study and Linear Regression, Decision Tree Regression, Random Forest Regression, Support Vector Machines Regression and Multiple Regression algorithms were used. Various metrics such as MAE, MSE and MAPE were used to evaluate these supervised models. The results show varying levels of success with different supervised machine learning algorithms for this challenging task. Among the trained models, Decision Trees and Random Forest Regression were evaluated as the algorithms with the best results (Yildiz et al., 2022).

Jonsson et al. (2016) investigated different classifiers for fault assignment problems and discovered that utilizing Stacked Generalization to combine the best classifiers enhances performance. An ensemble approach called stacked generalization (DG) uses the output of several classifiers as the input for a final classifier that determines the final class. According to the authors, the DG model's classification accuracy of four mistake datasets from a telecoms business ranged from 57% to 89%, which was comparable to the human approach in use at the time. Ultimately, the study demonstrated that using a topic's non-textual fields yields more encouraging outcomes.

Helming et al. (2011) suggests in their paper a novel model-based method that takes into account the connections between task items and system properties when assigning. They contrast this method with other approaches that investigate both textual content and structure information. Every technique is used on many kinds of work items, such as tasks and bug reports. They look through the model repositories of three distinct projects, including historical data, for our assessment to see how well they perform using various techniques. In this paper, they present a new model-based

method for semi-automatically assigning task items and compare other machine learning strategies that are currently in use. Every method is used on a single, unified model that is built within the UNICASE tool.

The fault classification method proposed by Zibran (2016) is based on topic modeling. The method used is the Labeled Latent Dirichlet Discrimination (EGDA) algorithm, named after the German mathematician Johann Peter Gustav Lejeune Dirichlet. In this study, we investigate the effectiveness of labeled EGDA in automatically classifying error reports into a predefined set of categories.

Bhattacharya et al. (2012) use a probabilistic graph-based model in this paper, which they propose to be a model that makes highly accurate predictions. This is the first study to look at how several machine learning dimensions (features, training history, and classifiers) affect prediction accuracy in fault assignment, as well as how fault discard graphs affect it. Using data from Eclipse and Mozilla that spans 21 combined development years and 856,259 bug reports, they objectively assess their methodology. They demonstrate how their method can greatly shorten throwing pathways and achieve up to 86.09% prediction accuracy in fault assignment. They contend that the greatest results for their dataset come from combining a Naive Bayes classifier with scatter plots, incremental learning, and product-component characteristics. They highlight optimization strategies that shorten training and prediction times while delivering excellent prediction accuracy.

In recent years, research on regularization methods has increased significantly over time due to the need to develop more accurate and reliable prediction models. Papers that examine various clustering and optimization problems using regularization methods include evaluations based on cost function results.

Li and Zhou (2009) addressed the group weight finding problem by employing a cost function that combined hinge loss and L1 regularization. They utilized Quadratic programming to minimize this cost function, conducting experiments with Decision tree classifiers and UCI datasets. Additionally, they proposed a semi-supervised version and found that the Regularized Selective Ensemble Algorithm (RSE) could generate ensembles with strong generalization ability while maintaining a small size.

Zhang and Zhou (2011) tackled the weight finding problem, formulating three distinct cost functions: LP1, which utilized only Hinge loss; LP2, incorporating Hinge loss and L1 adjustment; and LP3, allowing negative weights. Linear programming was employed to minimize these cost functions, and the experiments featured the K-Nearest Neighbor (KNN) algorithm as base classifiers along with UCI datasets.

Goldberg and Eckstein (2012) approached the weight finding problem using the indicator loss function and L0 regularization. They considered this problem NP-hard in specific cases and provided various relaxation strategies and bounds for solving it. Importantly, their work was primarily theoretical in nature, distinguishing it from other practical implementations.

Tinoco et al. (2013) combined MLP and SVM algorithms for classifying remote sensing images, employing genetic algorithms to find the weights. An improved version of their work utilized hinge loss and L1 regularization, with linear programming employed to minimize the cost function. Both versions classified remote sensing images using an ensemble of MLP and SVM classifiers.

Hautamaki et al. (2013) explored sparse ensembles in the speaker verification domain, modeling ensemble weight finding with a cross entropy loss function and three regularization functions: L1, L2, and L1+L2. The Nelder-Mead method was used to minimize these cost functions, and logistic regression classifiers were employed in the experiments.

Yin et al. (2012) addressed ensemble weight finding, incorporating a cost function with squared loss, L1 regularization, and diversity terms based on Yule's Q statistic. Their experiments featured neural network classifiers on six UCI datasets, and the proposed cost function was initially minimized using genetic algorithms.

Şen and Erdogan (2013) modeled ensemble weight finding using a cost function that included hinge loss and two regularization functions: L1 and group sparsity. Convex optimization techniques were employed to minimize this cost function, and experiments involved comparing 13 classifiers on 12 UCI datasets and three other datasets using CVX Toolbox.

Mao et al. (2013) tackled ensemble weight finding using a cost function consisting solely of absolute loss, minimizing it through a 0-1 matrix decomposition. In a subsequent work, they proposed a cost function with squared loss and L1 regularization, minimized using a quadratic form approach. Decision tree weak classifiers and UCI datasets were used in both studies.

Özgür et al. (2018) introduced a sparsity-driven weighted ensemble classifier (SDWEC) to enhance classification accuracy and minimize the number of classifiers. SDWEC formed ensembles with pre-trained classifiers, and the assigned weights determined how base classifiers voted. Efficiency tests on 11 datasets showed that SDWEC outperformed or matched state-of-the-art classifier ensemble methods, achieving similar accuracy levels with fewer classifiers and reducing testing time for the ensemble.

3. APPROACH

It is important to note that supervised learning is utilized when we wish to anticipate a certain outcome based on a specific input and we have instances of input/output pairings. Our training set consists of these input/output pairs, from which we construct a machine learning model. Making precise forecasts for fresh, never-before-seen data is our aim. Building the training set for supervised learning frequently takes human labor, but once done, it automates and frequently accelerates a tedious or impractical activity.

Linear Models (Linear Regression, SVM, etc.) stand out as the most widely used Machine Learning algorithms. However, they have an important drawback, they are very prone to Overfitting. In its simplest form, as seen in the 2-dimensional plane shown in Figure 2, the best fitting line (or curve) segment to the data points is tried to be found.

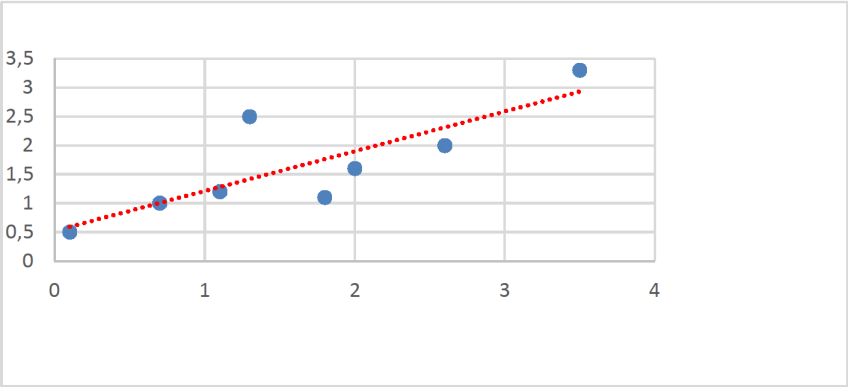

Figure 2. *Linear Regression*

The equation expressing linear regression models using coefficient weights is shown below.

$$y = w_0 + w_1 x_1 \tag{1}$$

As the number of variables increases, the coefficients also increase in number. Increasing the number of variables also means increasing the attributes added to the model. This situation can be explained with the following equation.

$$y = w_0 x_0 + w_1 x_1 + w_2 x_{2+} w_3 x_3 + w_4 x_4 + w_5 x_5 + \cdots + w_n x_n \tag{2}$$

Linear models are machine learning models that are prone to overfitting the training data. In case of overfitting the training data, the accuracy, reliability and generalization of the model in the testing phase are weakened. The concepts of overfitting, hyperparameters, variance and bias should also be mentioned.

Bias-variance decomposition is an important tool for understanding machine learning algorithms and its use in experimental studies has grown rapidly in recent years. The concepts of bias and variance help explain how very simple learners can outperform more complex learners and how groups of models can outperform single models (Domingos, 2000). In machine learning studies, bias is defined as the difference between the true value and the predicted value, and variance is defined as the amount by which the predictions deviate from the average prediction.

Every model has bias and variance error components. Bias and variance are inversely related; trying to reduce one component of the model will cause the other component to increase (Geman et al., 1992). Low bias and low variance are desirable characteristics in a model. The errors of the bias component are due to incorrect assumptions in the learning method. Figure 3 shows the relationship between bias and variance. At the point where the error is at its lowest, there is the necessary agreement between bias and variance for the model to obtain successful predictions. Complexity beyond this point means high variance. It is seen that the classification success will decrease.

In equation (2) above, as the coefficients $w_0, w_1, w_2, .., w_n$ increase, the variance will increase and a model that is difficult to generalize will emerge.

Since variance is sensitive to changes in the fit of the model, even a small change in the training data, it generates errors; therefore, high variance can lead to the problem of overfitting (Dangeti, 2017).

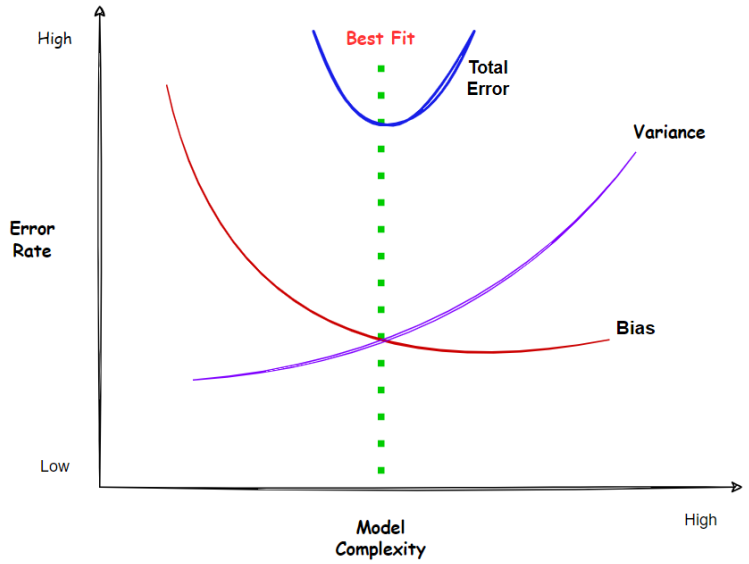

Figure 3. *Bias Variance Correlation*

The only measure of whether an algorithm will perform well on new data is the evaluation on the test set. However, intuitively, simple models are expected to generalize better when working with new data. Therefore, we always want to find the simplest model. Overfitting, or more widely, overfitting, is the process of creating a model that is too sophisticated for the data that we currently have. When a model is overfitted, it performs well on the training set but is unable to be applied to fresh data because it was fitted too tightly to the training set's features. However, if your model is very simplistic, you might not be able to fully capture all of the nuances and variations in the data. Underfitting or underlearning is the process of selecting a model that is overly simplistic (Muller & Guido, 2017). As shown in Figure 4, the model fits all the outcome points. The distribution of the dataset is also very influential on overfitting. For example, if the class distribution is 90% to 10% in a two-class data set, a prediction model run on this data set will have a 90% success rate in the training data. This will also lead to overlearning, i.e. the model will be memorized. On the test data set or a new data set, this success cannot be achieved.

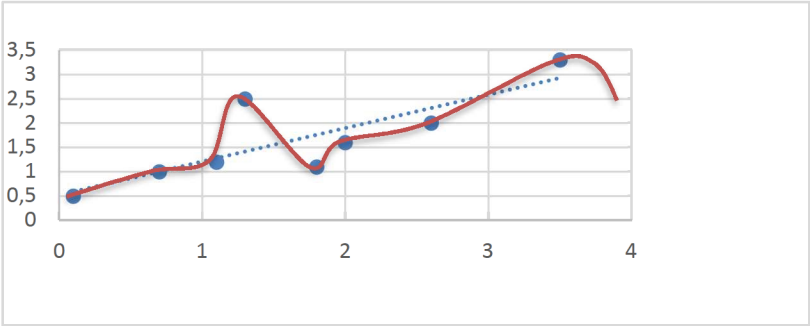

Figure 4. *Extreme learning curve in regression models*

The more the complexity of the model is allowed to increase, the better the training data is predicted. However, if the model becomes too complex, it starts to focus too much on each data point in the training set and the model cannot generalize well to new data. There is a sweet spot in between that will provide the best generalization performance. This is where the desired model tuning takes place. The variation in model performance between overfitting and underfitting is shown in Figure 5. Overfitting models have high variance and low bias. Where underfitting is observed, high bias is observed.

In order to prevent overfitting, multiple learning algorithms (ensemble), early stopping, cross-validation, feature engineering, expanding the volume of the dataset to create diversity and reducing the complexity of the model with the regularization method examined in this article are used. Linear models are prone to overfitting.

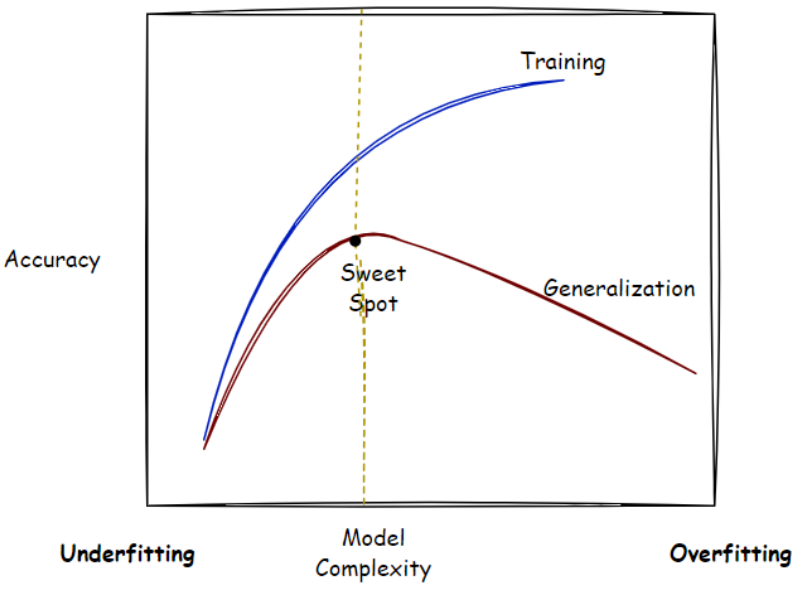

Figure 5. *Impact of model complexity on training and test accuracy*

The performance of a machine learning model can vary due to multiple factors, including data structure, size, the number of records related to classes, algorithms, performance verification methods, sampling techniques, and feature selection methods. Hyperparameters used in algorithms constitute a crucial factor influencing model performance, as they are parameters manipulated by the model developer. Consequently, hyperparameter tuning is a significant research area focused on optimizing these parameters to achieve the best possible model performance (Koçoğlu & Özcan, 2022).

Two types of parameters are encountered in machine learning models. These are called model parameters and hyperparameters. Model parameters are parameters that are included in the model and can be estimated from the data. They are not added to the model later by the expert during data analysis. Weights in a neural network, support vectors in a support vector machine, coefficients in linear regression or logistic regression are examples of model parameters (Tanyildizi & Demirtas, 2019).

Hyperparameters, unlike parameters, are not estimated from the data and depend on manual adjustment by the expert designing the model (Mantovani et al., 2017). Hyperparameters are adjustable parameters that can be selected by expert experience and trial and error methods. However, selecting the right hyperparameters requires an algorithmic process. Hyperparameter selection is seen as an optimization problem (Doğan, 2021).Kernel parameter (γ), epsilon value (ϵ) used in support vector machines; neighborhood value (k) in K-Nearest Neighbor algorithm; filter size, number of filters, number of neurons, number of layers, activation function, etc. used in deep neural network algorithms are among the examples of hyperparameters (Şipal et al., 2022).

Examples of hyperparameter solution methods considered as optimization problems are Grid Search, Random Search, Bayesian Optimization, Cross Validation and Hyperopt, Scikit Optimize and Optuna, which are included in the library of Python software language called alternative methods (Doğan, 2021).

3.1. Regularization

Overfitting is one of the most typical problems that any data scientist deals with. It is common for a machine learning model to perform well on training data, but not so well on testing data or new data sets. This suggests that the model is unable to predict the output or the target column of unseen data by introducing noise into the output. Noise is data points in the data set that do not really reflect the true qualities of our data but are there by chance (Kotsilieris et al., 2022).

Regularization greatly reduces the variance of the model without introducing a large bias. Consequently, the tuning parameter (α) used in regularization techniques limits the impact on variance and bias. As the value of (α) increases, the value of the coefficients decreases, reducing variance. This increase in (α) is useful to some extent because it only reduces variance without sacrificing any important features in the data, thus avoiding over-fitting. However, once a certain value is reached, the model starts to lose important features, leading to bias and Poor Fit. Consequently, the

value of (α) should be chosen carefully. It is a useful strategy to improve the accuracy of regression models (Friedrich et al., 2023).

Data collection and data preprocessing are the main causes of Overfitting. A data set with an uneven distribution of features, noises, random data fluctuations and variance can have an adverse effect on model training. The model learns these random errors and fluctuations so well during training that the accuracy of the training data model becomes extremely high, at which point the overfitting problem is encountered. A simple solution to overfitting is to update and penalize the weights. Table 1 shows the types of regularization and general approaches. In this paper, experimental studies with Ridge, Lasso and Elastic Net methods based on penalization type regularization are evaluated.

Table 1. *Overview of types of regularization, general approaches and methods* (Friedrich et al., 2023)

Regularization Type	Description	Solution Approach and Methods
Penalization	Add penalty term(s) to fitting criterion	Ridge regression, LASSO, elastic net
		Bayesian regularization priors
		Constraints for parameters
		Random effects
		Semiparametric regression
Early stopping	Early stopping of an iterative fitting procedure	Coefficient paths in penalization approaches
		Boosting
		Pruning of trees
		Learning rate in deep neural networks
Ensembling	Combine multiple base-procedures to an ensemble	Bagging
		Random forests
		(Bayesian) model averaging
		Boosting
Other approaches	-	Injecting noise
		Random probing in model selection
		Out-of-sample evaluation

In linear forecasting models, the Least Squares method aims to minimize the forecast error. The coefficients of the model tend to grow in (-) or (+) direction. On the other hand, regulatory extensions penalize the growth of the model's coefficients. Since the penalization prevents the model coefficients from growing, it prevents the model from producing extreme results. The sum of the squares of the error is the sum of the differences between the actual data point and the result point, i.e. the predicted value, formed by the data point taken into the function. Linear regression models are based on explaining this total value.

The cost (loss) function when applying regularization is shown in equation (3).

$$L_R = \Sigma(y_{gercek} - y_{tahmin})^2 + \alpha \Sigma w_i^2 \tag{3}$$

The equation $\alpha \Sigma w_i^2$ denotes the regularization part, while the coefficient α in the equation is the regularization coefficient and is a hyper parameter. As mentioned in section 2.c, the hyperparameters are given to the model externally. The hyper parameter α takes values between 0 and 1.

w is a parameter and is always a positive value. It is clear from this explanation that the model always wants to keep the parameters w small. The hyperparameter α is determined according to the parameter w. The operations performed for this purpose are called penalization (Tian & Zhang, 2022).

In this paper, experiments are carried out with Lasso, Ridge and Elastic Net regularization methods.

3.1.1. L1 Regularization (Lasso)

Lasso regularization has an approach that forces the coefficients to converge to zero (Emmert-Streib & Dehmer, 2019).

$$L(w) = \Sigma_{i=1}^{n}(y_i - \hat{y}_i)^2 + \alpha \Sigma_{j=1}^{p} |w_j^1| \tag{4}$$

When the equation is examined, it is seen that the exponent of the parameter w is 1. The Lasso regularization is called L1, referring to the exponent 1 of the parameter w (Bharambe et al., 2022). The parameters and hyperparameters in equation (4) are defined below:

α = penalty term (between 0 and 1)
$|w_j^1|$ = absolute value of the coefficients (slope of the curve)
y_i = actual result
$\hat{y}_i$ = prediction result

3.1.2. L2 Regularization (Ridge)

Ridge regression reduces the size of the regression coefficients so that the coefficients of the variables are close to zero. The penalty term L2, the sum of the squared coefficients, is used to penalize the regression model that causes the coefficients to shrink. The alpha (α) constant, the hyper parameter, is used to fine-tune the amount of penalty. It is very important to choose a perfect value for α. When α is set to 0, the penalty component has no effect and the OLS coefficients are calculated using Ridge regression. However, when α approaches infinity, the shrinkage penalty becomes more significant and the Ridge regression coefficients approach zero (Golam Kibria & Banik, 2016).

$$L(w) = \Sigma_{i=1}^{n}(y_i - \hat{y}_i)^2 + \alpha\Sigma_{j=1}^{p} w_j^2 \tag{5}$$

When the equation is examined, it is seen that the exponent of the parameter w is 2. Ridge regularization is called L2, referring to the exponent 2 of the parameter w, just as in the Lasso method (Bharambe et al., 2022).

3.1.3. Elastic Net Regularization

Elastic Net emerged in reaction to criticism of Lasso, which relies heavily on data for variable selection, making it unstable. Ridge regression and Lasso's penalties are combined to get the best of both approaches (Paper, 2019).

$$L(w) = \Sigma_{i=1}^{n}(y_i - \hat{y}_i)^2 + \alpha\Sigma_{j=1}^{p}|w_j^1| + \alpha_2\Sigma_{j=1}^{p} w_j^2 \tag{6}$$

In the Elastic Net method, additional cost is added to the loss, cost function to form a hybrid of the L1 and L2 methods (Bharambe et al., 2022).

3.2. Support Vector Machines

The support vector machine can classify data into two or more classes with linear separation mechanisms in two-dimensional space, planar separation mechanisms in three-dimensional space and hyperplane separation mechanisms in multi-dimensional space (ÇELİK et al., 2021).

The case where a group of data can be separated by a line is the case where the group can be separated linearly. The idea is that the object separating the two classes is a corridor rather than a line, and that the width of this corridor is determined by some data vectors and is as wide as possible (Cortes & Vapnik, 1995).

In SVM literature, an attribute is termed a predictor variable, and a feature denotes a transformed symbol used to describe the hyperplane. Feature selection involves the task of choosing the most appropriate representation. A collection of features describing a case, such as a row of predictor values, is referred to as a vector. Therefore, the objective of SVM modeling is to identify the optimal hyperplane that separates sets of vectors, placing one-category cases of the target variable on one side of the plane and the other-category cases on the opposite side (Witten et al., 2016). The purpose of SVM in linear problems is to find a hyperplane passing through the features. This hyperplane consists of two lines where the features belonging to the classes are the furthest apart. Figure 6 shows the lines on this hyperplane.

In a non-linear dataset, SVMs cannot draw a linear hyperplane. Therefore, Kernel is used. The Kernel method greatly improves machine learning on nonlinear data. The operation of the SVM estimator (y) is expressed as follows (ARSLAN et al., 2020).

$$y = (K_{x_i} W_{j_k}) + b \tag{7}$$

The kernel function K_{x_i} is the bias term of the SVM network " b " and W_{j_k} is the weight vector. K and W denote Lagrange multipliers.

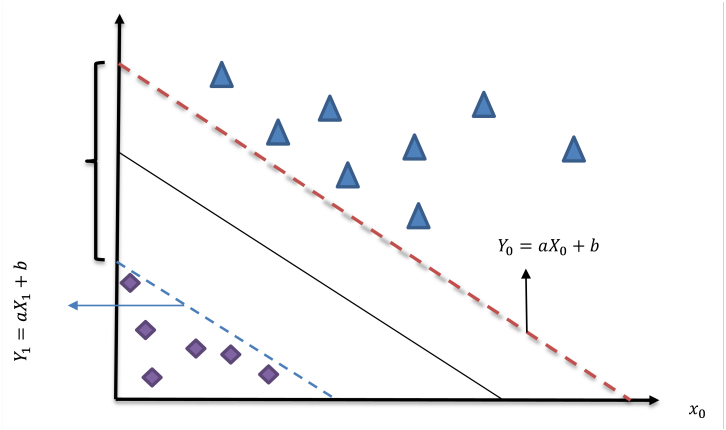

Figure 6. *Hyperplane and support vectors*

3.3. Logistic Regression

Logistic regression analysis is named after the logit transformation applied to the dependent variable (Hair et al., 2010). Logistic regression analysis is divided into three according to the type of scale on which the dependent variable is measured and the number of options of the dependent variable. If the dependent variable is a categorical variable with two options, it is called "Binary Logistic Regression Analysis". For example, binary logistic regression is applied when students are classified as successful and unsuccessful according to their completion of an academic program. If the dependent variable is a variable with more than two categories (levels), it is called "Multinominal Logistic Regression Analysis". For example, if there is a dependent variable consisting of students studying in three different academic programs, multinominal nominal logistic regression is applied. If the dependent variable is obtained with an ordinal scale, then "Ordinal Logistic Regression Analysis" is used. For example, ordinal logistic regression is applied when students' achievement in the academic program they are studying is grouped as "low", "medium" and "high" (Cook et al., 2001).

In logistic regression analysis, logit transformation is applied to the dependent variable and the logit of the dependent variable is estimated with the help of the independent variable. Logistic regression analysis, also called logit model, is a method used to determine the cause-and-effect relationship between independent variables and the dependent variable when the dependent variable has two, three or multiple categories and explains the effects of independent variables on the dependent variable with odds ratio (TAZEGÜL et al., 2016). Odds ratio is also called betting odds.

$$Odds\ Ratio = \frac{P_i}{1 - P_i} \tag{8}$$

According to Equation 8, P_i represents the probability of occurrence of an observed situation (i=1,2,3,. . . ,n) and 1-P_i represents the probability of non-occurrence of an observed situation. In this case, the dependent variable takes the value 1 for P_i ($Y_i = 1$) and 0 for 1-P_i ($Y_i = 0$), making it bicategorical. Moreover, independent variables can be continuous, categorical or both. Odds ratio is defined as the ratio of the probability that a situation will occur to the probability that it will not occur.

The odds ratio ensures that the probability estimation takes a value between 0 and 1. However, in order to prevent the odds ratio from taking a value below zero, the logit value should be calculated by taking the natural logarithm of the value obtained with the odds ratio. As a result of the calculation of the logit value, a metric variable that can be converted into a probability between 0-1 is obtained (ŞENEL & ALATLI, 2014).

The model in which the logit value is obtained by taking the natural logarithm after the odds ratio is calculated is shown in Equation 9.

$$logit(Y) = ln_e\left[\frac{P_i}{1 - P_i}\right] = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n \tag{9}$$

If the odds ratio is less than 1, the logit value takes a negative value, while if it is greater than 1, it takes a positive value. As a result of logistic regression analysis, the intended model, which is a non-linear logarithmic function, is obtained and the model coefficients are shown as logarithmic values. In order to eliminate this situation that makes the

interpretation of the model coefficients difficult, the exponential logistic coefficient value obtained by taking the anti-logarithms of the coefficients and denoted by the symbol Exp (β) is used. The model coefficients provide information about the direction of the relationship, while the exponential logistic coefficient provides information about how many times the change in the independent variable will decrease or increase the likelihood value.

3.4. Artificial Neural Network

ANN is a polycentric, parallel computational or rather modeling method inspired by the nerve cells in the human brain, first named as "neurons" by the German scientist Heinrich Wilhelm Gottfried von Waldeyer-Hartz in 1890 (Anderson D, 1992). The first concrete modeling technique based on neural cells was introduced by Frank Rosenblatt in 1958 as a simple perceptron (Yoon, 1989). Figure 7 shows the structure of an artificial neural network.

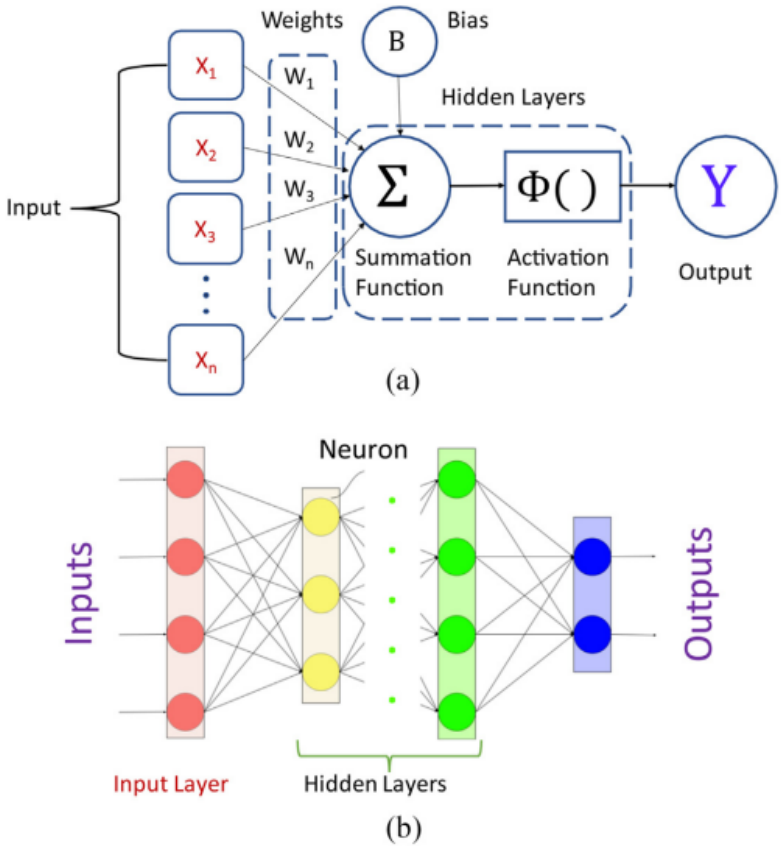

Figure 7. *(a) An artificial neuron, and (b) Structure of an artificial neural network* (Sinha et al., 2023)

The optimal architecture for a neural network should be sufficiently large to learn the problem yet small enough to generalize effectively. A network smaller than the optimal architecture struggles to learn the problem, while a larger network may overlearn the training data, resulting in poor generalization. Two primary approaches guide the determination of network structure: growing/constructive and pruning/destructive. The choice depends on whether the network's structure starts small and grows during learning (constructive) or begins large and shrinks during learning (destructive) (Aran et al., 2009).

There are basically three main layers in ANN, which are inspired by the information processing process of the brain. The names of these layers are input layer, hidden layer and output layer. The number of hidden layers can be one or more. The hidden layer between the input and output layer consists of structures called neurons. Each neuron in the layer is connected to all the neurons in the layer after it, but not to the neurons in the current layer. The input layer contains the parameters related to the state to be classified in the output layer. The hidden layer performs information processing and the output layer produces the class label or estimates the continuous time value (ÖZBİLGİN & KURNAZ, 2023).

Each relation between layers is assigned a weight value. As given in Equation 10, the values in the input layer are multiplied by the weights and given to a non-linear function by adding bias.

$$h_j = f(\Sigma_i w_{ij} x_i + b_j^1) \tag{10}$$

In the equation 10, x_i is the input parameters and w_{ij} is the weight value connecting input i to hidden neuron j. h_j is the output of hidden neuron j. b_j^1 bias and f is the activation function.

The mathematical expression of the sigmoid activation function, which is also widely used as a function, is as given in Equation 11.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{11}$$

The mathematical expression of the output y_j in the output layer is as in the following equations 12 and 13.

$$y_j = \text{soft max}(z_j) = \frac{e^{z_j}}{\Sigma_j e^{z_j}} \tag{12}$$

$$z_j = \Sigma_i a_{ij} h_i + b_j^2 \tag{13}$$

Here, z_j represents the output units, b_j^2 the bias and a_{ij} the weight value between the jth output neuron and the ith hidden neuron.

Artificial Neural Networks (ANN) are commonly used in regression and clustering problems as well as in classification.

4. EXPERIMENT SETUP

4.1. Dataset

In this study, research is conducted on a problem of classification type. Classification is a technique that decomposes data in accordance with predetermined outputs. Since the outputs are known in advance, classification learns the dataset supervised (Giudici & Jiang, 2006).

Table 2. *Jira Data*

Jira Data			
1	ISSUEID	21	ORIGINAL_REPORTER
2	PKEY	22	ASSIGNEE1
3	ISSUE	23	ASSIGNEE2
4	ISSUETYPE	24	ASSIGNEE3
5	ISSUE_TYPE	25	ASSIGNEE4
6	PRIORITY	26	ASSIGNEE5
7	COMPONENT	27	WORKLOG_ASSIGNEE1
8	LABEL	28	WORKLOG_ASSIGNEE2
9	URGENCY	29	WORKLOG_ASSIGNEE3
10	IMPACT	30	WORKLOG_ASSIGNEE4
11	PLATFORM	31	WORKLOG_ASSIGNEE5
12	ISSUE_CATEGORY	32	LOG_HOURS_ASSIGNEE1
13	ISSUE_SUB_CATEGORY	33	LOG_HOURS_ASSIGNEE2
14	SUMMARY	34	LOG_HOURS_ASSIGNEE3
15	DESCRIPTION	35	LOG_HOURS_ASSIGNEE4
16	ASSIGNEE	36	LOG_HOURS_ASSIGNEE5
17	REPORTER	37	COMMENTOR_COUNT
18	CREATED	38	COMMENT_COUNT
19	RESOLUTIONDATE	39	WAITING_HOURS_AT_CUSTOMER
20	DUEDATE		

In our problem, it is aimed to assign an ITSM company to the expert who will answer the requests from customers in the fastest and most accurate way according to the characteristics of the request. In our study, which is supported

by experimental results to evaluate the effects of the regularization method on the solution sought with classification algorithms, the data of the Help Desk services provided through Oracle platforms in Table 3 and the personnel data of the human resources department of the company in Table 4 will be used. The data belongs to the year 2022.

Table 3. *HR Data*

HR Data			
1	PERSON_ID	9	IS_BILGISI
2	FIRST_NAME	10	KADEME
3	LAST_NAME	11	PERSONEL_TIPI
4	USER_PERSON_TYPE	12	EMAIL_ADDRESS
5	ORG_NAME	13	DATE_OF_BIRTH
6	TAKIM	14	CINSIYET
7	POZISYON	15	EFFECTIVE_START_DATE
8	SORUMLULUK	16	EFFECTIVE_END_DATE

The two datasets are combined into a single dataset based on the assignment problem that forms the core of the study. Experimental results are performed on the new merged dataset. There are 39 variables in the Jira dataset and 16 variables in the HR dataset.

4.2. Preprocessing

Since the "assignee" in Table 3 and the "email address" in Table 4 contain the same information, these two tables are used as reference for merging. What is meant by table merging is the merging of data sets. The merged data sets are then used as a single data set in the experimental processes. The model prepared for the defined problem will be studied with this new data set. The results of the combined data set in the evaluations are shown in Table 5.

Table 4. *Attributes and Discriptions*

	Feature ID	Description		Feature ID	Description
1	ISSUEID	System Registration Number of Customer Demand	28	WORKLOG_ASSIGNEE2	Second Expert to Resolve the Ticket
2	PKEY	Customer Code	29	WORKLOG_ASSIGNEE3	Third Expert to Resolve the Ticket
3	ISSUE	Name And Code of Customer Submitting Ticket	30	WORKLOG_ASSIGNEE4	Fourth Expert to Resolve the Ticket
4	ISSUETYPE	Demand Code	31	WORKLOG_ASSIGNEE5	Fifth Expert to Resolve the Ticket
5	ISSUE_TYPE	Demand Type	32	LOG_HOURS_ASSIGNEE1	Time Spent By the First Expert On Ticket
6	PRIORITY	Priority of Demand	33	LOG_HOURS_ASSIGNEE2	Time Spent By the Second Expert On Ticket
7	COMPONENT	Components From Which Demand Originstes	34	LOG_HOURS_ASSIGNEE3	Time Spent By the third Expert On Ticket
8	LABEL	The Unit to Which Ticket Will Be Associated	35	LOG_HOURS_ASSIGNEE4	Time Spent By the Fourth Expert On Ticket
9	URGENCY	The Urgency of Ticket	36	LOG_HOURS_ASSIGNEE5	Time Spent By the Fifth Expert On Ticket
10	IMPACT	The Effed After Demand Intervention	37	COMMENTOR_COUNT	Number of Commenters for Ticket
11	PLATFORM	ExpertPlatform Where Ticket is Studied	38	COMMENT_COUNT	Number of Comments On the Ticket
12	ISSUE_CATEGORY	The Service Category In Which Ticket is Listed	39	WAITING_HOURS_AT_CUSTOMER	Waiting time for customer approval of the resolved demand
13	ISSUE_SUB_CATEGORY	Sub-Service Category of Ticet	40	PERSON_ID	The Experts Code Within The Company
14	SUMMARY	Textual Date of Ticet	41	FIRST_NAME	Name of The Expert
15	DESCRIPTION	Description of The Tidtet Submitted By The Cus tomer	42	LAST_NAME	Last Name of The Expert
16	ASSIGNEE	Expert Assigned to Ticket	43	USER_PERSON_TYPE	Job Status of Expert
17	REPORTER	The Person Reporting Ticket	44	ORG_NAME	The Department to which expert reports
18	CREATED	Date of Creation of Ticket	45	TAKIM	Departmental Teams
19	RESOLUTIONDATE	Resolution date of Ticket demand	46	POZISYON	Title of The Employee
20	DUEDATE	Deadline For Processing demand	47	SORUMLULUK	Responsibility of the experts
21	ORIGINAL_REPORTER	The main responsible to Report	48	IS_BILGISI	Personnel's Degree of Responsibility
22	ASSIGNEE1	The First Responsible expert to demand	49	KADEME	Seniority of the experts
23	ASSIGNEE2	The second responsible expert to demand	50	PERSONEL_TIPI	Department Source of the experts
24	ASSIGNEE3	The third responsible expert to demand	51	EMAIL_ADDRESS	E-Mail Address
25	ASSIGNEE4	The fourth responsible expert to demand	52	DATE_OF_BIRTH	Expert's Date of Birth
26	ASSIGNEE5	The fifth responsible expert to demand	53	CINSIYET	Gender of Pers onnel
27	WORKLOG_ASSIGNEE1	First Expert to Resolve the Ticket	54	EFFECTIVE_START_DATE	Start Date
			55	EFFECTIVE_END_DATE	Termination Date

After the mentioned stages, the preparation process of the dataset is completed and it is made suitable for working with machine learning algorithms. By combining the 39-variable Jira dataset and the 16-variable HR dataset, a new 55-variable dataset was organized. Some of the variables in the expanded dataset, such as ISSUEID, may erroneously affect the experimental results of the model positively or negatively. ISSUEID only shows the registration order of the incoming request in the JIRA system. Therefore, it will not be useful to include it in the model. In the process of organizing the data set for the model, we tried to remove similar variables from the data set. As a result of the editing

Table 5. *The data for classification analysis*

	Column	Dtype
0	PRIORITY	object
1	URGENCY	object
2	IMPACT	object
3	ISSUE_CATEGORY	object
4	IS_BILGISI	object
5	Total_Assignee	int64
6	Total_Worklog_Assginee	int64
7	Total_Log_Hours_Assignee	int64
8	COMMENTOR_COUNT	int64
9	COMMENT_COUNT	int64
10	WAITING_HOURS_AT_CUSTOMER	int64
11	ISSUE_TYPE	object
12	Complition_Time	int64

process, a data set with 12 variables was obtained as shown in Table 6. In the next stages, data analysis is performed on the data set with Python programming language version 3.11.3.

As can be seen in Table 6, there are 7 properties of data type int64 and 6 properties of data type object. Int data type represents integer data. Object data type represents all other data types except specific data types. It is frequently encountered in data where objects such as letters, numbers and signs are used together. The dataset uses more than 6.4 MB of memory.

Another important point to consider when preparing the dataset is the distribution of the data. If the data input is concentrated in one data point, this will confuse the forecasting model. New data is predicted in the same way as it would be in the same data set. This is an example of overlearning and in this problem, we try to influence this situation with regularization models. The feature column named position was removed from the dataset with the drop function because it was skewed to a single class exceeding 90

Table 6. *Data distributions for categorical data*

ISSUE_CATEGORY				PRIORITY		IS_BILGISI	
IS1	0,300	WIP	0,002323	Major	0,62	IS2	0,27
PO	0,110	BI	0,002231	Minor	0,26	Junior	0,16
Custom	0,080	QA	0,001677	Critical	0,05	KD	0,14
HR	0,080	XTR	0,001600	Trivial	0,04	Senior	0,14
Salesforce	0,078	OIE	0,001569	Blocker	0,02	UZY	0,09
AP	0,075	PIM	0,000877			DU	0,09
INV	0,054	LINUX	0,000569	URGENCY		Uzman	0,03
GL	0,054	IPROC	0,000354	Medium	0,46	Consultant	0,02
Database	0,036	Training	0,000323	Low	0,43	Principal	0,02
AR	0,028	CE	0,000308	High	0,11	BİY	0,01
Sysadmin	0,025	OPMCosting	0,000308			Danışman	0,01
OE	0,020	IT	0,000292	IMPACT		DU	0,00
FA	0,014	Hyperion	0,000138	I1	0,58	Yönetici	0,00
Development	0,012	FAH	0,000062	SPSP	0,29	YAS	0,00
EAM	0,009	GRC	0,000046	OCWW	0,07	SY	0,00
PA	0,005	WMS	0,000046	NCA	0,04	Partner	0,00
ISUPPLIER	0,003	OrgPub	0,000031	ABSP	0,03	Müdür	0,00
CST	0,003					SUPC	0,00

By looking at the plot graph shared in Figure 8 for the columns containing *numerical* data, the distribution situations are conveyed. In the graphs, it is seen that the distributions of *Total Log Hours Assignee* and *Waiting Hours at Customer* are in an undesirable situation. It is natural to encounter such graphical shapes in real life data because the probabilistic state of life is reflected in the data. *Total Assignee* and *Completion Time* features show a favorable distribution.

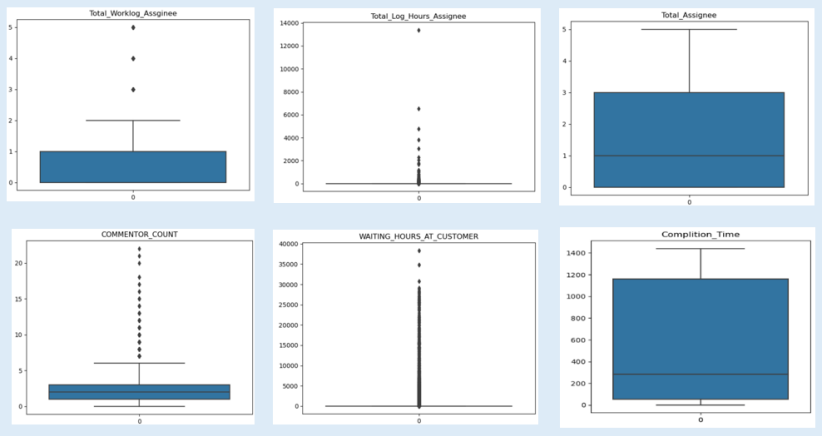

Figure 8. *Plot graphs of numerical properties*

Box Plot graphs of columns containing categorical data were analyzed according to Completion Time, which is the label class. Some of the inferences obtained from the graph shared in Figure 9 ;

- Tickets with Issue_type of Others and Incident take longer to complete
- Tickets with Is_information SY (Sales Manager) take longer to complete
- Tickets with GRC as Issue_category take longer to complete
- Tickets with Impact as OCWW take longer to complete
- Tickets with Priority as Trivial take longer to complete

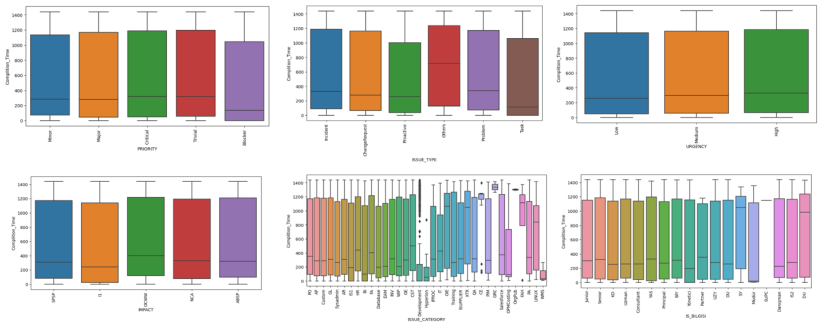

Figure 9. *Box Plot Graphs for categorical data*

A heat map was created to understand the correlation value between these variables. According to the heat map given in Figure 10;

- According to the heat map, the highest correlation is between Total_Log_Hours_Assignee, COMMENT_COUNT.
- The column most closely related to Complition_Time is COMMENTOR_COUNT.
- The weakest associated features/attribute is between Total_Log_Hours_Assignee and Complition Time.

Information about the dataset is described in this section, including data types, descriptions of real-life data and the steps involved in preparing the data for the algorithms. One hot encoding method was applied to categorical data with *get_dummies* function. This increased the number of columns, i.e. the number of features, to 77. After the outlier check, there were 64993 rows, which decreased to 64512 rows after this process, which means that 7 per thousand data was removed, which is an acceptable rate. Finally, the dataset was divided into 70% training 45158 and 30% test 19354 set.

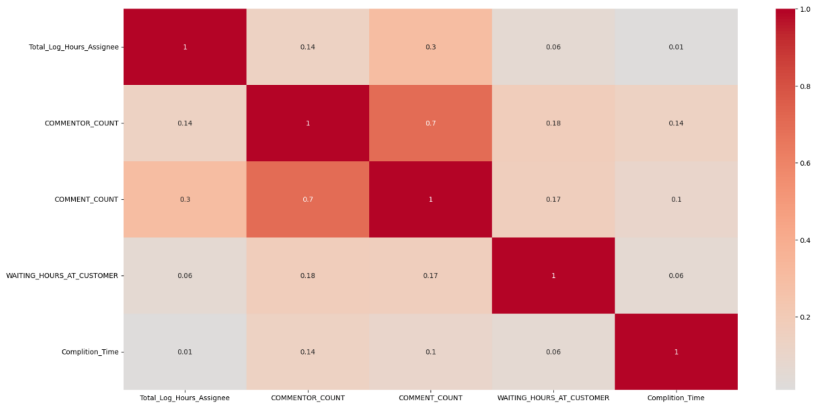

Figure 10. *Heatmap to understand the correlation value between these variables*

4.3. Evaluation Metrics

There are a number of evaluation metrics that are used to assess the models generated by classification algorithms and to compare which classification model yields better results. These metrics are usually based on a table structure called a Confusion Matrix. The Confusion Matrix is a table structure designed to illustrate how a classifier performs in machine learning and statistics classification problems (Ha et al., 2011).

There are four possible outcomes in dataset classification: a true positive (TP) when a truly positive example is correctly classified as positive, a false negative (FN) when a truly positive example is incorrectly classified as negative, a true negative (TN) when a truly negative example is correctly classified as negative, and a false positive (FP) when a truly negative example is incorrectly classified as positive (ALAN & KARABATAK, 2020).

Table 7. *Confusion Matrix*

		Actual Values	
Predicted Values		**Positive**	**Negative**
Positive	+	TP	FP
Negative	-	FN	TN

The performance metrics you use to measure how successful the model is are very important. If the evaluation is not done with the right metrics, a successful model can be characterized as unsuccessful and an unsuccessful model as successful. Machine Learning models are measured with the following metrics according to their types; Accuracy, Recall, Precision, F1 Score, ROC-AUC Curve, Log-Loss (Logarithmic Loss). The calculation methods of these criteria are given in the formulas below;

Accuracy is the rate at which the model created using the training set correctly classifies the data in the test set.

$$(Accuracy) = \frac{TP + TN}{TP + TN + FP + FN} \tag{14}$$

Recall is the rate at which the classifier correctly predicts data that belong to the positive class.

$$(Recall) = \frac{TP}{TP + FN} \tag{15}$$

Precision indicates the proportion of the positively predicted classifications that are correctly predicted.

$$Kesinlik(Precision) = \frac{TP}{TP + FP} \tag{16}$$

F Measure (F1 Score) is the weighted average of Precision and Recall. Therefore, it considers both FP (False Positive) and FN (False Negative) values.

$$(F1\ Score) = 2 * \frac{(Precision) * (Recall)}{(Precision) + (Recall)} \tag{17}$$

Apart from these criteria, another method used to evaluate classification performance is the ROC-AUC Curve (YETGINLER & ATACAK, 2020). It shows how successful the model is in separating the classes from each other on *Figure 11*. ROC stands for Receiver Operating Characteristic Curve or Probability Curve; AUC stands for Area Under the Curve or Area Under the Probability Curve. It takes a value between 0 and 1 and the closer it is to 1, the more successful the model is.

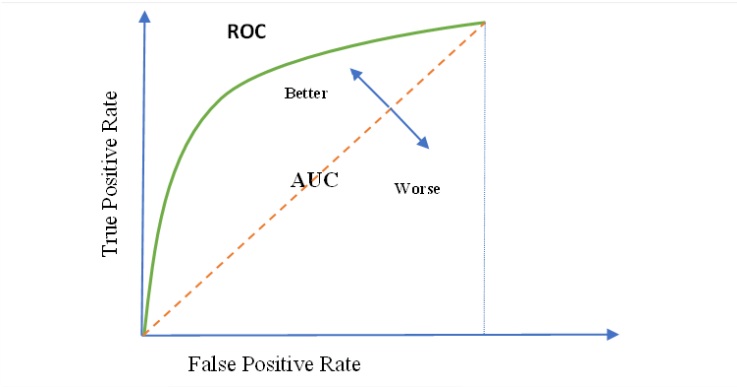

Figure 11. *ROC-AUC Curve*

In ROC curves, the x-axis is the FPO (False Positive Odds Ratio), while the y-axis is the TPO (True Positive Odds Ratio). For different threshold values, TPO and FPO values, i.e. sensitivity and precision values, are calculated. TPO and FPO pairs form the ROC curve. The ROC Curve is an increasing function between (0,0) and (1,1). After ROC analysis, it can be decided whether a test whose diagnostic success is evaluated is useless or a very good test (Orynbassar et al., 2022).

Root Mean Squared Error (RMSE) and r^2 also used to measure the performance of our model and k-fold cross validation is applied.

5. EXPERIMENTAL RESULTS AND DISCUSSION

The study adheres to the Cross Industry Standard Process for Data Mining (CRISP-DM) steps utilized in both Data Mining and Machine Learning. This process involves six steps: understanding the problem, understanding the data, preparing the data for analysis, developing models, evaluating model performances, and selecting the best model for application (Koçoğlu & Esnaf, 2022). In these stages, data analysis is performed on the data set with Python programming language version 3.11.3. The main libraries used are NumPy, SciPy, Matplotlib, Seaborn and SciKit-Learn.4

The data set was first fed into the linear regression model to find the prediction values of the completion time of the solution request from the customer called ticket. Then, the prediction success of the linear regression model was found and the effect of applying L2 (Ridge) and L1 (Lasso) regularization methods on this success value was observed. Figure 12 shows the mentioned processes on the diagram.

For the experimental application, the specified data set needs to be loaded and data preprocessing operations need to be performed. In data preprocessing, outlier detection and conversion of numerical data that may cause deviations such as year and month into string data are prioritized. Categorical data is corrected by encoding (LabelEncoder and OneHotEncoder). The next stage is Normalization, scaling and missing data management. Normalization rescales the data between 0 and 1. Feature scaling is based on the normal distribution and is calculated as the ratio of the distance of each value from the mean to the standard deviation. These are the preparations for using the dataset in experimental studies according to machine learning methods.

The problem is defined as correcting the success of the regression model established for predicting the completion time of a customer request with regularization and examining the classification success of the customer requests sent to the system according to the ISSUE TYPE class label through the classification method of the corrected model.

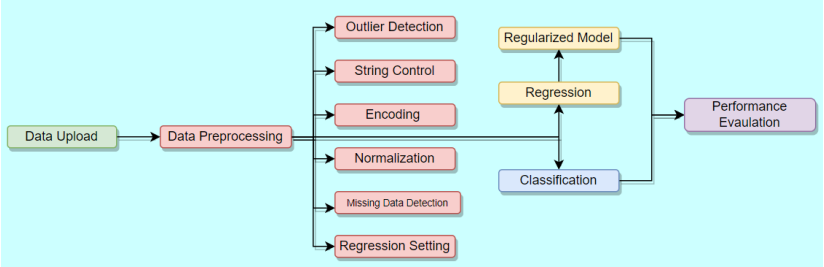

Figure 12. *Diagram of the proposed model*

In order to investigate the effect of regularization methods on the success of the model, an application is made on the dataset. The process starts with an unregularized OLS model and then a regularization model is constructed with L1 and L2 approaches.

The basic approach when reading the OLS table is to ensure that the p value is less than 0.05 *Confidence Interval*. Variables greater than this value do not have a significant effect on the result. Therefore, variables that are greater than this value are identified and removed from the model. Table 8 shows the results of the first iteration of the OLS method. According to the values under the p>ItI column in the list, variables such as PRIORITY_Blocker, 'IMPACT_NCA', 'ISSUE_CATEGORY_AP', etc. are dropped from the model with the code block 'X_train.drop('PRIORITY_Blocker',axis=1, inplace=True)'. The same process is continued until there is no variable with a p value greater than 0.005 in the list. In the model, this conclusion is reached at the end of the 3rd iteration. Score was calculated as R-squared: 0.753

Table 8. *OLS Regression results for first iteration*

	coef	std err	t	P>|t|	[0.025	0.975]		coef	std err	t	P>|t|	[0.025	0.975]
Total_Assignee	392.672	3.032	12.951	0.000	33.324	45.210	ISSUE_CATEGORY_IT	-425.168	130.730	-0.325	0.745	-298.750	213.716
constant	2.041.672	24.292	8.405	0.000	156.555	251.780	ISSUE_CATEGORY_LINUX	2.148.940	107.443	2.000	0.045	4.304	425.484
Total_Worklog_Assginee	-108.761	4.396	-2.474	0.013	-19.492	-2.260	ISSUE_CATEGORY_OE	156.124	27.505	0.568	0.570	-38.298	69.523
Total_Log_Hours_Assignee	-0.0726	0.033	-2.227	0.026	-0.136	-0.009	ISSUE_CATEGORY_OIE	1.703.153	62.245	2.736	0.006	48.315	292.316
COMMENTOR_COUNT	520.994	2.619	19.890	0.000	46.965	57.233	ISSUE_CATEGORY_OPMCosting	-2.101.062	134.870	-1.558	0.119	-474.453	54.241
COMMENT_COUNT	-28.274	0.494	-5.720	0.000	-3.796	-1.859	ISSUE_CATEGORY_OrgPub	1,26E-09	6.16e-13	2.046	0.041	5.28e-14	2.47e-12
WAITING_HOURS_AT_CUSTOMER	0.0039	0.002	2.057	0.040	0.000	0.008	ISSUE_CATEGORY_PA	-260.093	41.579	-0.626	0.532	-107.505	55.486
PRIORITY_Blocker	-0.5575	17.609	-0.032	0.975	-35.071	33.956	ISSUE_CATEGORY_PIM	0.8541	82.525	0.010	0.992	-160.896	162.604
PRIORITY_Critical	595.563	12.322	4.833	0.000	35.405	83.708	ISSUE_CATEGORY_PO	571.395	22.718	2.515	0.012	12.612	101.667
PRIORITY_Major	525.489	8.352	6.292	0.000	36.179	68.918	ISSUE_CATEGORY_QA	232.251	64.974	0.357	0.721	-104.124	150.574
PRIORITY_Minor	354.272	8.918	3.973	0.000	17.948	52.906	ISSUE_CATEGORY_Salesforce	1.280.628	23.673	5.410	0.000	81.664	174.462
PRIORITY_Trivial	571.923	17.623	3.245	0.001	22.650	91.734	ISSUE_CATEGORY_Sysadmin	-225.475	26.330	-0.856	0.392	-74.154	29.059
URGENCY_High	657.044	10.047	6.540	0.000	46.012	85.396	ISSUE_CATEGORY_Training	1.332.519	139.459	0.955	0.339	-140.091	406.595
URGENCY_Low	557.391	9.455	5.895	0.000	37.208	74.271	ISSUE_CATEGORY_WIP	-950.147	54.681	-1.738	0.082	-202.191	12.162
URGENCY_Medium	827.236	9.076	9.115	0.000	64.934	100.513	ISSUE_CATEGORY_WMS	-6.046.066	365.248	-1.655	0.098	-1.320.499	111.286
IMPACT_ABSP	358.105	15.959	2.244	0.025	4.530	67.091	ISSUE_CATEGORY_XTR	2.317.465	65.252	3.552	0.000	103.851	359.642
IMPACT_I1	519.183	16.278	3.189	0.001	20.013	83.824	IS_BILGISI_BİY	32.622	44.163	0.074	0.941	-83.297	89.821
IMPACT_NCA	252.520	18.489	1.366	0.172	-10.987	61.491	IS_BILGISI_Consultant	-182.579	41.968	-0.435	0.664	-100.515	63.999
IMPACT_OCWW	659.721	12.105	5.450	0.000	42.246	89.698	IS_BILGISI_DU	2.007.261	63.014	3.185	0.001	77.217	324.235
IMPACT_SPSP	252.143	9.700	2.599	0.009	6.202	44.226	IS_BILGISI_Danışman	-364.473	45.141	-0.807	0.419	-124.925	52.030
ISSUE_CATEGORY_AP	-24.818	23.137	-0.107	0.915	-47.830	42.867	IS_BILGISI_IS2	53.188	39.332	0.135	0.892	-71.773	82.411
ISSUE_CATEGORY_AR	245.116	25.860	0.948	0.343	-26.175	75.198	IS_BILGISI_Junior	60.753	39.517	0.154	0.878	-71.378	83.529
ISSUE_CATEGORY_BI	-798.816	55.688	-1.434	0.151	-189.032	29.269	IS_BILGISI_KD	-172.307	39.568	-0.435	0.663	-94.785	60.324
ISSUE_CATEGORY_CE	4.459.041	140.326	3.178	0.001	170.862	720.946	IS_BILGISI_Müdür	-6.798.101	356.890	-1.905	0.057	-1.379.321	19.701
ISSUE_CATEGORY_CST	628.408	47.624	1.320	0.187	-30.503	156.185	IS_BILGISI_Partner	-273.108	253.948	-0.108	0.914	-525.053	470.431
ISSUE_CATEGORY_Custom	143.265	23.131	0.619	0.536	-31.011	59.664	IS_BILGISI_Principal	147.473	42.809	0.344	0.730	-69.158	98.653
ISSUE_CATEGORY_Database	-428.091	24.977	-1.714	0.087	-91.765	6.147	IS_BILGISI_SUPC	4.731.995	503.157	0.940	0.347	-512.996	1.459.395
ISSUE_CATEGORY_Development	-2.228.992	31.378	-7.104	0.000	-284.400	-161.399	IS_BILGISI_SY	1.493.105	102.654	1.455	0.146	-51.893	350.514
ISSUE_CATEGORY_EAM	-42.952	33.791	-0.127	0.899	-70.527	61.936	IS_BILGISI_Senior	586.655	39.561	1.483	0.138	-18.875	136.206
ISSUE_CATEGORY_FA	523.978	29.420	1.781	0.075	-5.266	110.062	IS_BILGISI_UZY	91.607	39.863	0.230	0.818	-68.972	87.293
ISSUE_CATEGORY_FAH	2.702.357	258.644	1.045	0.296	-236.711	777.182	IS_BILGISI_Uzman	-80.007	41.051	-0.195	0.845	-88.462	72.460
ISSUE_CATEGORY_GL	475.042	23.890	1.988	0.047	0.679	94.329	IS_BILGISI_YAS	1.339.403	99.567	1.345	0.179	-61.214	329.094
ISSUE_CATEGORY_GRC	893.393	365.407	0.244	0.807	-626.864	805.543	IS_BILGISI_Yönetici	-539.267	67.779	-0.796	0.426	-186.774	78.921
ISSUE_CATEGORY_HR	1.310.836	23.211	5.647	0.000	85.590	176.578	ISSUE_TYPE_ChangeRequest	-264.431	10.036	-2.635	0.008	-46.113	-6.773
ISSUE_CATEGORY_Hyperion	-5.930.275	211.631	-2.802	0.005	-1.007.828	-178.227	ISSUE_TYPE_Incident	39.533	17.168	0.230	0.818	-29.697	37.603
ISSUE_CATEGORY_INV	262.815	23.800	1.104	0.269	-20.367	72.930	ISSUE_TYPE_Others	2.811.207	13.729	20.477	0.000	254.212	308.029
ISSUE_CATEGORY_IPROC	187.429	139.665	0.134	0.893	-255.003	292.489	ISSUE_TYPE_Proactive	-545.683	11.530	-4.733	0.000	-77.167	-31.969
ISSUE_CATEGORY_IS1	213.409	25.708	0.830	0.406	-29.047	71.728	ISSUE_TYPE_Problem	250.827	14.633	1.714	0.087	-3.598	53.764
ISSUE_CATEGORY_ISUPPLIER	-292.476	47.796	-0.612	0.541	-122.929	64.434	ISSUE_TYPE_Task	-249.781	11.034	-2.264	0.024	-46.605	-3.352

We examine the results obtained for the OLS regression by applying L1 and L2 regularization methods. The results of the regularized model are compared. The purpose of regularization models is to add a penalty term to Linear Regression to prevent the coefficients from growing too large. But they work a little differently; Ridge penalizes high coefficient values but does not force them to zero. Lasso forces as many coefficients as possible to zero.

In the Ridge method model, the model is built separately with all combinations for the hyperparameters and their values to be tested and the most successful hyperparameter set is determined according to the specified metric. GridSearchCV method is used for this in the model. Alpha parameter values for GridSearchCV are 0.0001, 0.001, 0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 20, 50, 100, 500, 1000. As a result of the experiments, the most appropriate param value was found to be 6 and cross-validation was defined as 10-fold.

R2 Score (Train): 0.75054674060912334
R2 Score (Test): 0.74791859089008832
RMSE (Train): 0.083440
RMSE (Test): 0.084280

In Lasso regularization, like Ridge, the initial value of the alpha parameter is given as 0.0001. An important point at this stage is that 10-fold fitting is performed for each of the 28 candidates, for a total of 280 fits.

R2 Score (Train): 0.759693391808405374
R2 Score (Test): 0.748265086687702795
RMSE (Train): 0.082810
RMSE (Test): 0.082721

Table 9 is provided to evaluate the results of the Ridge and Lasso methods collectively. Since both models have the same R2 score around 0.75 and RMSE score around 0.08, it is better to choose the simpler model. In this respect, the Lasso model does a better job as it does feature selection resulting in 45 features while the Ridge model has 76 features which is 31 features more than the Lasso model.

Table 9. *Ridge and Lasso results*

	Metric	Ridge	Lasso
0	R2 Score (Train)	0.750547	0.759693
1	R2 Score (Test)	0.747919	0.748265
2	RMSE (Train)	0.083440	0.082810
3	RMSE (Test)	0.084280	0.082721

The mean RMSE obtained in the regression analysis before regularization was calculated as 0.126. After regularization, the RMSE error metric is around 0.08. It is stated that the Lasso, i.e. L1 regularization obtained a more successful result. Therefore, Lasso has actually done Feature Elimination (Variable Reduction). One of the most common uses of Lasso is this Feature Elimination process.

Also, all the above features have a positive correlation with the ticket's completion time data. A zero coefficient indicates that the variable indicated by that coefficient is insignificant for the outcome. Therefore, the model is simplified by reducing the number of variables. Simplicity is a good thing for Machine Learning. Variance is reduced, Bias is increased and better generalization is possible.

A new situation has emerged by examining the data set with regularization methods. It is seen that some attributes have no effect on the model and higher prediction success can be achieved by excluding these attributes from the model. At this stage, it is seen that the dataset to be used in the classification experiments will now be corrected and more reliable.

A common approach to running classification algorithms is to use an exploratory data analysis approach. Exploratory Data Analysis (EDA) is an approach to summarizing data by taking its key features and visualizing them with appropriate representations (Sahoo et al., 2019).

One of the issues analyzed in the problem is the classification of the type of customer demand by using a classification method. This information is defined in the dataset by an attribute called ISSUE_TYPE. Therefore, the dependent parameter labeled in classification algorithms is categorical. LR, SVM and ANN classifier algorithms are generally

used in models where categorical dependent variables are class labels. Figure 13 shows the initialization screen of the LR algorithm for Pyhton coding. The mathematical flow of the algorithm is coded by calling the suitable libraries.

```
[2]: import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     import seaborn as seabornInstance
     from sklearn.model_selection import train_test_split
     from sklearn.linear_model import LogisticRegression # import logistic regression
     from sklearn import metrics

     %matplotlib inline
```

```
[3]: dataset = pd.read_excel('omdSVM.xlsx')
     dataset.describe()
```

[3]:

	Complition_Time	Total_Assignee	Total_Worklog_Assginee	Total_Log_Hours_Assignee	COMMENTOR_COUNT	CO
count	64995.000000	64995.000000	64995.000000	64995.000000	64995.000000	
mean	556.486453	1.516809	0.691761	3.683068	1.985122	
std	546.193641	1.394241	0.725491	69.228870	1.485799	
min	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	53.000000	0.000000	0.000000	0.000000	1.000000	
50%	281.000000	1.000000	1.000000	1.000000	2.000000	
75%	1160.000000	3.000000	1.000000	2.000000	3.000000	
max	1440.000000	5.000000	5.000000	13380.000000	22.000000	

Figure 13. *Python Coding Screenshot for Logistic Regression Algorithm*

During the preparation of the data set for classification algorithms according to the EDA approach, various library and function updates are encountered depending on the software language used in the coding phase. As can be seen in Figure 14, one of these updates was encountered during the histogram graph display phase. *histplot* function is used instead of *distplot* function.

```
plt.figure(figsize=(15,10))
plt.tight_layout()
sns.distplot(dataset['ISSUE_TYPE'])
```

```
C:\Users\ali.alsac\AppData\Local\Temp\ipykernel_22852\2263707846.py:3: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(dataset['ISSUE_TYPE'])
```

```
plt.figure(figsize=(15,10))
plt.tight_layout()
sns.histplot(dataset["ISSUE_TYPE"], kde=True)
```

```
<Axes: xlabel='ISSUE_TYPE', ylabel='Count'>
```

Figure 14. *Histogram graph plot display update warning*

The code block of the ROC curve of the LR algorithm is given in Figure 15.

```
# Plot the ROC curve
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'AUC = {roc_auc:.4f}')
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve - Logistic Regression')
plt.legend(loc='lower right')
plt.show()
```

Figure 15. *The code block of the ROC curve of the LR algorithm*

As mentioned in the Evaluation Metrics section, ROC analysis is used to decide whether a test whose results are evaluated for predictive accuracy is useless or a very good test. According to the graph shared in Figure 16, the tests show that the model's approach to generating problem-specific solutions is positive. The area under the ROC curve, i.e. the AUC ratio, was calculated as 0.8622. It is a ratio that shows that the LR algorithm model is good at separating classes from each other.

Figure 16. *LR ROC Curve*

The code block given in Figure 17 was used to display all the classification algorithm results in a table. With this code, Model, Accuracy, Precision, Recall and F1 Score results can be seen in Table 10.

```
predictions_base = pd.concat(predictions, ignore_index=True, sort=False)
predictions_base = predictions_base.sort_values(by=['Recall'], ascending=False).style.hide(axis='index')

predictions_base
```

Figure 17. *Coding block for Prediction Results*

The data set used in the application consists of real life data. As seen in the data preparation section, the data is not balanced, that is, homogeneously distributed. Therefore, only accuracy is not used in the evaluation.

As seen in Table 10, SVM algorithm has the highest prediction success among the classification algorithms. LR and ANN algorithms gave very close results. In fact, all three algorithms have close values. Based on the principles mentioned in the evaluation metrics section, it is seen that the SVM algorithm has the best result when the accuracy is high and the F1 score is low. While creating the ANN architecture, the architectural structure of the model has a great impact on the accuracy rates. In this study, the architecture established for the ANN (single hidden layer with

Table 10. *Prediction Results*

Model	Accuracy	Precision	Recall	F1 Score
LR	0.728881	0.144476	0.761194	0.242857
SVM	0.732825	0.194476	0.732172	0.182833
ANN	0.724444	0.144111	0.771484	0.246841

ten neurons and 10000 iterations, sigmoid function as activation function and ten repetitions) was the model that gave accuracy rates of 70%. Optimizing the parameters of the ANN architecture also takes days.

In addition, compared to ANN, SVM method, after determining the kernel function (the most used kernel function is the radial basis function), results can be obtained in a single move. In this point of view, it is a much easier method to implement compared to ANN.

In this study, the ANN and SVM algorithms produced better results than the Logistic Regression algorithm. However, it should be considered that methods such as logistic regression can get results very quickly using a simple background, while methods such as ANNs are difficult to mature their architecture and reach iteration numbers such as 50000-100000 on the computer.

6. CONCLUSIONS AND FUTURE WORK

ITSM services have a very important place in the modernization processes of organizations. ITSM enables teams within organizations to produce value faster as software-centric services increase. It is now a necessity to add Artificial Intelligence-supported models to the systems used for collaboration, ease of use and a faster, quality value-producing ITSM service. In our study, data sets obtained through Jira tools used for ITSM services were used. It is imperative to improve existing systems and utilize up-to-date techniques and technologies in order to assign customer requests to the most accurate expert and to provide the most appropriate service in accordance with the service conditions included in the SLA agreements between the service providers and service recipients.

As a result of the experimental investigations, it has been revealed that the regularization approach has a positive effect on improving model performance as an important elimination tool in feature engineering. The most important problem encountered especially in the analysis of *real life* datasets is the lack of understanding of the suitability of the features to the designed machine learning models. In this study, the examination of the dataset with *L1* and *L2* regularization methods and the resulting regularization of the dataset provided a more suitable dataset for the next stage, classification.

This study shows that the classification method can be used to learn the problem type of customer requests. It is observed that ANN, SVM and LR algorithms are suitable algorithms for classification. As a result of this study, it is concluded that ITSM companies should not delay in establishing their own neural networks and quickly incorporating artificial intelligence into their business processes.

In summary, using classification and regularization methods in demand type (Issue-Type) forecasting allows for more accurate, interpretable, and robust models. This approach leverages the strengths of both techniques, resulting in better predictions and more informed decision-making.

In future studies, it is expected to make a positive contribution to the literature by understanding that regularization is an advantageous method, especially in examining data sets consisting of real-life data according to the feature engineering approach, and that classification studies carried out on data sets arranged according to the logic of regularization can give better results.

Peer Review: Externally peer-reviewed.

Author Contributions: Conception/Design of Study- M.M.Y., M.C.G., M.D., A.A., T.U.; Data Acquisition- T.U., A.A., M.C.G.; Data Analysis/Interpretation- M.M.Y., M.C.G., M.D., A.A., T.U.; Drafting Manuscript- A.A., T.U.; Critical Revision of Manuscript- M.M.Y., M.C.G.; Final Approval and Accountability- M.M.Y., M.C.G., M.D., A.A., T.U.; Material and Technical Support- M.D., A.A., T.U.; Supervision- M.M.Y., M.C.G.

Conflict of Interest: The authors have no conflict of interest to declare.

Grant Support: This work is supported by TUBITAK 1509 (program number 9210017) and TUBITAK 2244 (program number 119C056).

Acknowledgements: I would like to greatly acknowledge Experteam which is a trademark of Uzman Bilişim A.Ş.

ORCID IDs of the authors / Yazarların ORCID ID'leri

Ali Alsaç 0000-0002-8585-4501
Mehmet Mutlu Yenisey 0000-0002-4532-344X
Murat Can Ganiz 0000-0001-8338-991X
Mustafa Dağtekin 0000-0002-0797-9392
Taner Ulusinan 0009-0000-3647-0408

REFERENCES

ALAN, A., & KARABATAK, M. (2020). Veri Seti - Sınıflandırma İlişkisinde Performansa Etki Eden Faktörlerin Değerlendirilmesi. *Fırat Üniversitesi Mühendislik Bilimleri Dergisi, 32*(2). https://doi.org/10.35234/fumbd.738007

Anderson D, M. G. (1992). Artificial Neural Networks Technology. *Kaman Sciences Corporation*, 258(6).

Aran, O., Yildiz, O. T., & Alpaydin, E. (2009). An incremental framework based on cross-validation for estimating the architecture of a multilayer perceptron. *International Journal of Pattern Recognition and Artificial Intelligence, 23*(2). https://doi.org/10.1142/S0218001409007132

ARSLAN, H., ÜNEŞ, F., DEMİRCİ, M., TAŞAR, B., & YILMAZ, A. (2020). Keban Baraj Gölü Seviye Değişiminin ANFIS ve Destek Vektör Makineleri ile Tahmini. *Osmaniye Korkut Ata Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 3*(2). https://doi.org/10.47495/okufbed.748018

Bharambe, Prof. P., Bagul, B., Dandekar, S., & Ingle, P. (2022). Used Car Price Prediction using Different Machine Learning Algorithms. *International Journal for Research in Applied Science and Engineering Technology, 10*(4). https://doi.org/10.22214/ijraset.2022.41300

Bhattacharya, P., Neamtiu, I., & Shelton, C. R. (2012). Automated, highly-accurate, bug assignment using machine learning and tossing graphs. *Journal of Systems and Software, 85*(10). https://doi.org/10.1016/j.jss.2012.04.053

ÇELİK, E., DAL, D., & AYDİN, T. (2021). Duygu Analizi İçin Veri Madenciliği Sınıflandırma Algoritmalarının Karşılaştırılması. *European Journal of Science and Technology.* https://doi.org/10.31590/ejosat.905259

Cook, D., Dixon, P., Duckworth, W. M., Kaiser, M. S., Koehler, K., Meeker, W. Q., & Stephenson, W. R. (2001). Binary Response and Logistic Regression Analysis. *Project Beyond Traditional Statistical Methods, Ml.*

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning, 20*(3). https://doi.org/10.1023/A:1022627411411

Dangeti, P. (2017). Statistics for Machine Learning: Techniques for exploring supervised, unsupervised, and reinforcement learning models with Python and R. In *Packt Publishing.*

Deloitte, & TUBISAD. (2022). *Bilgi ve İletişim Teknolojileri Sektörü 2021 Pazar Verileri*.

Doğan, C. (2021). *İstatistiksel ve Makine Öğrenme ile Derin Sinir Ağlarında Hiper-Parametre Seçimi İçin Melez Yaklaşım* [Yüksek Lisans]. Hacettepe Üniversitesi.

Domingos, P. (2000). A Unified Bias-Variance Decomposition. *Aaai/Iaai.*

Emmert-Streib, F., & Dehmer, M. (2019). High-Dimensional LASSO-Based Computational Regression Models: Regularization, Shrinkage, and Selection. In *Machine Learning and Knowledge Extraction* (Vol. 1, Issue 1). https://doi.org/10.3390/make1010021

Friedrich, S., Groll, A., Ickstadt, K., Kneib, T., Pauly, M., Rahnenführer, J., & Friede, T. (2023). Regularization approaches in clinical biostatistics: A review of methods and their applications. In *Statistical Methods in Medical Research* (Vol. 32, Issue 2). https://doi.org/10.1177/09622802221133557

Geman, S., Bienenstock, E., & Doursat, R. (1992). Neural Networks and the Bias/Variance Dilemma. *Neural Computation, 4*(1). https://doi.org/10.1162/neco.1992.4.1.1

Golam Kibria, B. M., & Banik, S. (2016). Some ridge regression estimators and their performances. *Journal of Modern Applied Statistical Methods, 15*(1). https://doi.org/10.22237/jmasm/1462075860

Goldberg, N., & Eckstein, J. (2012). Sparse weighted voting classifier selection and its linear programming relaxations. *Information Processing Letters, 112*(12). https://doi.org/10.1016/j.ipl.2012.03.004

Ha, J., Kambe, M., & Pe, J. (2011). *Data Mining: Concepts and Techniques*. In Data Mining: Concepts and Techniques. https://doi.org/10.1016/C2009-0-61819-5

Hair, J. F., Black, W. C., Babin, B. J., & Anderson, R. E. (2010). Multivariate Data Analysis. In *Vectors*. https://doi.org/10.1016/j.ijpharm.2011.02.019

Hautamaki, V., Kinnunen, T., Sedlak, F., Lee, K. A., Ma, B., & Li, H. (2013). Sparse classifier fusion for speaker verification. *IEEE Transactions on Audio, Speech and Language Processing, 21*(8). https://doi.org/10.1109/TASL.2013.2256895

Helming, J., Arndt, H., Hodaie, Z., Koegel, M., & Narayan, N. (2011). Automatic Assignment of Work Items. C*ommunications in Computer and Information Science, 230*. https://doi.org/10.1007/978-3-642-23391-3_17

Jonsson, L., Borg, M., Broman, D., Sandahl, K., Eldh, S., & Runeson, P. (2016). Automated bug assignment: Ensemble-based machine learning in large scale industrial contexts. *Empirical Software Engineering, 21*(4). https://doi.org/10.1007/s10664-015-9401-9

Koçoğlu, F. Ö., & Esnaf, Ş. (2022). Machine Learning Approach and Model Performance Evaluation for Tele-Marketing Success Classification. *International Journal of Business Analytics, 9*(5). https://doi.org/10.4018/ijban.298014

Koçoğlu, F. Ö., & Özcan, T. (2022). A grid search optimized extreme learning machine approach for customer churn prediction. *Journal of Engineering Research.*

Kotsilieris, T., Anagnostopoulos, I., & Livieris, I. E. (2022). Special Issue: Regularization Techniques for Machine Learning and Their Applications. In *Electronics (Switzerland)* (Vol. 11, Issue 4). https://doi.org/10.3390/electronics11040521

Li, N., & Zhou, Z. H. (2009). Selective ensemble under regularization framework. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5519 LNCS. https://doi.org/10.1007/978-3-642-02326-2_30

Mantovani, R. G., Horvath, T., Cerri, R., Vanschoren, J., & De Carvalho, A. C. P. L. F. (2017). Hyper-Parameter Tuning of a Decision Tree Induction Algorithm. *Proceedings - 2016 5th Brazilian Conference on Intelligent Systems, BRACIS 2016.* https://doi.org/10.1109/BRACIS.2016.018

Mao, S., Xiong, L., Jiao, L. C., Zhang, S., & Chen, B. (2013). Weighted ensemble based on 0-1 matrix decomposition. *Electronics Letters, 49*(2). https://doi.org/10.1049/el.2012.3528

Muller, A. C., & Guido, S. (2017). Introduction to Machine Learning with Python: a guide for data scientist. In *O'Reilly Media, Inc.*

Orynbassar, A., Sapazhanov, Y., Kadyrov, S., & Lyublinskaya, I. (2022). Application of ROC Curve Analysis for Predicting Students' Passing Grade in a Course Based on Prerequisite Grades. *Mathematics, 10*(12). https://doi.org/10.3390/math10122084

ÖZBİLGİN, F., & KURNAZ, Ç. (2023). Koroner Arter Hastalığının İris Görüntülerinden Yerel İkili Örüntüler ve Yapay Sinir Ağı Kullanılarak Tahmini. *Karadeniz Fen Bilimleri Dergisi, 13*(2). https://doi.org/10.31466/kfbd.1266996

Özgür, A., Nar, F., & Erdem, H. (2018). Sparsity-driven weighted ensemble classifier. *International Journal of Computational Intelligence Systems, 11*(1). https://doi.org/10.2991/ijcis.11.1.73

Paper, D. (2019). Hands-on Scikit-Learn for Machine Learning Applications: Data Science Fundamentals with Python. In *Hands-on Scikit-Learn for Machine Learning Applications: Data Science Fundamentals with Python.* https://doi.org/10.1007/978-1-4842-5373-1

Sahoo, K., Samal, A. K., Pramanik, J., & Pani, S. K. (2019). Exploratory data analysis using python. *International Journal of Innovative Technology and Exploring Engineering, 8*(12), 4727–4735. https://doi.org/10.35940/ijitee.L3591.1081219

Şen, M. U., & Erdogan, H. (2013). Linear classifier combination and selection using group sparse regularization and hinge loss. *Pattern Recognition Letters, 34*(3). https://doi.org/10.1016/j.patrec.2012.10.008

ŞENEL, S., & ALATLI, B. (2014). Lojistik Regresyon Analizinin Kullanıldığı Makaleler Üzerine Bir İnceleme. *Eğitimde ve Psikolojide Ölçme ve Değerlendirme Dergisi, 5*(1). https://doi.org/10.21031/epod.67169

Sinha, K., Uddin, Z., Kawsar, H. I., Islam, S., Deen, M. J., & Howlader, M. M. R. (2023). Analyzing chronic disease biomarkers using electrochemical sensors and artificial neural networks. In *TrAC - Trends in Analytical Chemistry* (Vol. 158). https://doi.org/10.1016/j.trac.2022.116861

Şipal, B., Ormancı, B. B., & Altınel, A. B. (2022). KELİME ANLAM BULANIKLIĞINI GİDERMEK İÇİN DİFÜZYON REGÜLARİZASYON VE NORMALİZASYON TEKNİKLERİNİN KULLANILMASI. In *MÜHENDİSLİK ALANINDA ULUSLARARASI ARAŞTIRMALAR VI* (pp. 75–85).

Tanyildizi, E., & Demirtas, F. (2019). Hiper Parametre Optimizasyonu Hyper Parameter Optimization. *1st International Informatics and Software Engineering Conference: Innovative Technologies for Digital Transformation, IISEC 2019 - Proceedings.* https://doi.org/10.1109/UBMYK48245.2019.8965609

TAZEGÜL, A., YAZARKAN, H., & YERDELEN, C. (2016). İşletmelerin Finansal Başarılı ve Başarısız Olma Durumlarının Veri Madenciliği ve Lojistik Regresyon Analizi İle Tahmin Edilebilirliği. *Ege Akademik Bakis (Ege Academic Review), 16*(1). https://doi.org/10.21121/eab.2016119960

Tian, Y., & Zhang, Y. (2022). A comprehensive survey on regularization strategies in machine learning. In *Information Fusion* (Vol. 80). https://doi.org/10.1016/j.inffus.2021.11.005

Tinoco, S. L. J. L., Santos, H. G., Menotti, D., Santos, A. B., & Dos Santos, J. A. (2013). Ensemble of classifiers for remote sensed hyperspectral land cover analysis: An approach based on Linear Programming and Weighted Linear Combination. *International Geoscience and Remote Sensing Symposium* (IGARSS). https://doi.org/10.1109/IGARSS.2013.6723730

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). Data Mining: Practical Machine Learning Tools and Techniques. In *Data Mining: Practical Machine Learning Tools and Techniques.*

YENİSU, E. (2021). Ekonomiyi Harekete Geçiren Kilit Sektörler Nelerdir? Türkiye Üzerine Bir Girdi-Çıktı Analizi. *İzmir İktisat Dergisi, 36*(4). https://doi.org/10.24988/ije.721302

YETGINLER, B., & ATACAK, İ. (2020). Sentiment Analyses on Movie Reviews using Machine Learning-Based Methods. *Artificial Intelligence Studies, 3*(2). https://doi.org/10.30855/ais.2020.03.02.01

Yildiz, M., Alsac, A., Ulusinan, T., Ganiz, M. C., & Yenisey, M. M. (2022). IT Support Ticket Completion Time Prediction. *Proceedings - 7th International Conference on Computer Science and Engineering,* UBMK 2022. https://doi.org/10.1109/UBMK55850.2022.9919591

Yin, X. C., Huang, K., Hao, H. W., Iqbal, K., & Wang, Z. Bin. (2012). Classifier ensemble using a heuristic learning with sparsity and diversity. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 7664 LNCS*(PART 2). https://doi.org/10.1007/978-3-642-34481-7_13

Yoon, B. L. (1989). Artificial neural network technology. ACM SIGSMALL/PC Notes, 15(3), 3–16. https://doi.org/10.1145/74657.74658

Zhang, L., & Zhou, W. Da. (2011). Sparse ensembles using weighted combination methods based on linear programming. *Pattern Recognition, 44*(1). https://doi.org/10.1016/j.patcog.2010.07.021

Zibran, M. F. (2016). On the effectiveness of labeled latent dirichlet allocation in automatic bug-report categorization. *Proceedings - International Conference on Software Engineering.* https://doi.org/10.1145/2889160.2892646

How cite this article

Alsac, A., Yenisey, M.M., Ganiz, M.C., Dagtekin, M, Ulusinan, T. (2023). The efficiency of regularization method on model success in issue type prediction problem. *Acta Infologica, 7*(2), 360-383. https://doi.org/10.26650/acin.1394019