T.C.
BAHÇEŞEHİR UNIVERSITY

FACULTY OF ENGINEERING AND NATURAL SCIENCES

CAPSTONE FINAL REPORT

PREDICTION OF CONCRETE COMPRESSIVE STRENGTH USING MACHINE LEARNING

Ramazan Afşin – Civil Engineering

Gülbahar Erol – Computer Engineering

Eren Güneştaş – Computer Engineering

Ali Yağız İlban – Computer Engineering

Elif İlteroğlu – Civil Engineering

Ömer Lütfi Kara – Civil Engineering

Advisors:

Assist. Prof. Görkem Kar – Computer Engineering

Assist. Prof. İrem Şanal – Civil Engineering

ISTANBUL, June 2022

STUDENT DECLARATION

By submitting this report, as partial fulfillment of the requirements of the Capstone course, the students promise on penalty of failure of the course that

- they have given credit to and declared (by citation), any work that is not their own (e.g. parts of the report that is copied/pasted from the Internet, design or construction performed by another person, etc.);
- they have not received unpermitted aid for the project design, construction, report or presentation;
- they have not falsely assigned credit for work to another student in the group, and not take credit for work done by another student in the group.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

LIST OF ABBREVIATIONS

IoT	Internet of Things
M2M	Machine-to-Machine
IEEE	The Institute of Electrical and Electronics Engineers
ML	Machine Learning
DL	Deep Learning
ANN	Artificial Neural Network
HPC	High Performance Concrete
UHPC	Ultra-High Performance Concrete
ABA	Adaptive Boosting Algorithm
SVM	Support Vector Machine
PCE	Concrete Pressure Device
CSV	Comma-Separated Value
RF	Random Forest
DNN	Deep Neural Network
kNN/KNN	k-Nearest Neighbour
DT	Decision Tree
ROC	Receiver Operating Characteristic
ReLU	Rectifier to Linear Unit

1. OVERVIEW

Concrete, one of the most important parts of the construction industry, is obtained by mixing 7 main materials in certain proportions. The use of additives in different proportions significantly changes the durability of concrete. The quality of ready mixed concrete is a product that can only be understood at the end of the 28th day after the test piece is checked. Knowing the concrete quality in advance will provide the project with factors such as the resources being under control and the construction plan being more controlled (independent). For this reason, the aim of our graduation project is; The previously obtained concrete data will be made available with machine learning algorithms and these data will be used for model training in the determined machine learning algorithm. This model obtained will have the lowest margin of error when compared with the real data obtained in concrete strength tests. The main purpose of the tests of civil engineering students is that the rate and level of each ingredient affects the formation of concrete obtained from many mixtures. In this project, the main aim is how changing scales and substance ratios affect and deteriorate the quality of concrete. The final results will be transferred to computer engineering students, and computer engineering students will enter this data into the system they use, calculate their margin of error and prepare them for presentation.

In the Computer Engineering Department, Gülbahar Erol will gather information about Machine Learning algorithms reviews and comparisons. Eren Güneştaş will make improvements on the optimization of the model to be used. Ali Yağız İlban will take part in the testing and development of the obtained algorithms. Civil engineering student Elif İlteroğlu is responsible for the use of the PCE device, which is still in use today, and the research and ordering of the materials required for concrete. Ramazan Afşin will examine the condition of the concrete obtained before it enters the test device and will ensure the collection and flow of information to be transferred to Computer Engineering Students. Ömer Lütfi Kara will transfer the data table about the durability of concrete, which was entered into the latest PCE device and whose measurements were made, to Computer Engineer Students.

1.1. Identification of the need

In this research project, it is aimed to measure the compressive strength of concrete to be obtained from a concrete mixture by using Machine Learning algorithms. In order to determine the concrete to be used in a project, the compressive strength value of concrete is one of the most important unknowns of the project. Compressive strength value after pouring test concrete can be calculated at the end of the 28th day. This time and the material spent is an important problem for the continuity of the project. This ambiguity in the middle of construction projects has not yet been resolved at the desired rate, and concrete compressive strength estimation studies are in the focus of all researchers to eliminate this ambiguity. For this reason, the estimation of the concrete compressive strength value with the machine learning algorithm in advance will affect the work of civil engineers and everyone working in this field. Determining the concrete compressive strength estimation method obtained in this research and optimizing the estimation values is of serious importance and is a resource for all researchers who will work in the field of computer engineering, industrial engineering, artificial intelligence engineering or mathematics.

1.2. Definition of the problem

Durability of hardened concrete is often a problem, as concrete with different strengths can be produced due to the mixing ratio of the materials, the number of days required for hardening, the kneading/mixing time and the quality of the materials. In addition, to test the strength of the concrete, it is possible to comment on the strength according to the values obtained by drying, turning it into concrete and subjecting it to tests with various machines. However, there are minimum times required for these tests. We can consider this time as a serious waste of time. In addition, the tests cause great costs for different reasons. Both the raw material money spent for concrete production and the concrete waste generated during this process, as well as the labor cost and the cost of using the testing machines make this process even more difficult. In addition, the lost time causes extra costs.

The aim of this project is to provide the strongest concrete production under the most accurate conditions by creating an estimation model without the need for a compressive strength test based on the reasons stated above.

1.2.1. Functional requirements

Functions of the team that, they create the concrete by pouring cement themselves, tests the durability of the concrete with the press machine and/or various machines and use the results obtained, and/or use the results of various tests made in the previous times on the internet, using the most ideal mixing ratio with high performance and accuracy via the Machine Learning Algorithm.

1.2.2. Performance requirements

Determining the required ratios for the strength of the concrete by making the most accurate evaluation with the data set used, which is expected from the project. In this way, it is aimed to waste time for compressive strength tests and to prevent garbage consisting of concrete used for tests.

1.2.3. Constraints

Our team consists of 3 Computer Engineers and 3 Civil Engineers. In this project, only expenditure is planned for the experimental part. This costs 123 TL with all the materials.

Depending on the density of the materials used in concrete production, the durability of concrete also changes. The water/binder ratio, aggregate size, binder type or waste composition affect the compressive strength of concrete. The compressive strength of the concrete is evaluated by breaking the standard size cylinders or cubes after the sample castings are formed and kept for a while. Since these tests are costly and time consuming, it would be economical and efficient to evaluate the compressive strength using ML using various other parameters with different efficiencies.

Concrete production causes a significant amount of carbon dioxide emit. It is possible to say that this amount is 1/1 ratio. However, it is also seen as the main source of greenhouse gas emit. From this point of view, our project greatly reduces environmental pollution.

It is possible to say that it is a cost-reducing factor for the construction industry. Up to 28 days are required to perform the compressive strength tests. This causes delay in the project process. Failure to perform the test will overlook the quality control, which can cause serious problems. In this case, fast and safe solutions are needed. Evaluations made with ML allow us to reach quick solutions without reducing the quality.

1.3. Conceptual solutions

In supervised learning-based tasks, the model is made with the labeled data set. This means that the data sets have observation samples with expected results. Types of supervised learning algorithms include classification and regression. If the outputs are limited to a limited set of values, classification algorithms are used; Regression algorithms are used when a range can have any numerical value within it.

The aim of this project is to save time in a more partial way by basing the compressive strength test on ML estimation on various parameters. Supervised learning is appropriate for the type of data used and the purpose of the project. In this context, we have determined a few algorithms. ANN, SVM, ABA.

The SVM algorithm is generally used in classification problems. Considering that the samples belonging to two classes are linearly distributed, it is aimed to distinguish these two classes from each other with a decision function obtained using the training data.

The ANN algorithm is used in areas such as classification, prediction, control, data filtering and interpretation. A neural network is a machine learning algorithm based on a human neuron model. The human brain is made up of millions of neurons. It sends and processes signals in the form of electrical and chemical signals. It is capable of machine learning as well as pattern recognition. Systems of interconnected "neurons", such as information processing technique, can calculate values from inputs. It can produce meaningful results from these calculations.

The ABA algorithm is a technique used as the Ensemble Method. It was developed for binary classification purposes. The most convenient and common algorithm used with ABA is single-level decision trees. Because these trees are so short and contain only one classification decision, they are often called decision logs. It strengthens weak learners.

In line with these studies, it is aimed to obtain the most accurate result by making examinations with these algorithms and comparing the results.

1.3.1. Literature Review

In our academic article research; The parameters used in data sets commonly vary between 6 and 9. The most common cause of this variability is the addition of extra ingredients [1] or physical attributes of concrete[5]. Other common one is creation of values obtained by reusing parameters [4].

One of the commonly used method as ML is ANN [1, 2, 4, 5]. In one of the studies using this method[1], they were aimed to increase the strength of the concrete by adding shredded used tires in the concrete is also a very good example to reduce the pollution to nature.

Although not many preferred, the method encountered method is Fuzzy Neural Network[3].

In terms of the concretes used, there are UHPC [4] and HPC [2] among those mentioned. Regarding the remaining concretes, there is no distinction in terms of classification.

According to the articles we researched, we see that samples of different sizes and shapes (cylinder-cube) are used to see the effect of the shape and size of the samples on the compressive strength of concrete. According to the results, the degree of cracking throughout the sample is observed more intensely in cube samples. While a main inclined fracture surface is nucleated in cylindrical samples, it has been observed that the side edges of the cubes are fragmented, leading to hourglass failure mode [15].

The results obtained for the samples belonging to the 28-day curing period were more approximate than the results of the samples belonging to the 7-day curing period [16].

As a result of some academic research comparing machine learning methods, the following conclusions were reached; “Using the K-Nearest Neighbor (KNN) approach, which is commonly favored in the literature, Şahan et al. were able to achieve a 99.14% success rate. With the LS-SVM (Least Squares Support Vector Machines) method, Polat and Güneş were able to achieve a 98.53% success rate. In another work using the SVM model, Akay was able to to achieve a 99.51% success rate. The ANN method was stated to have a success rate of 97.4% by Karabatak and Ince. Kahramanli and Allahverdi, on the other hand, found 99.31% progress in their analysis using the ANN method.” [17], which indicates that the highest rate was reached with SVM method but it should not be overlooked that KNN and ANN also have very close percentage success rates.

In a study on "Identification of Encrypted Traffic" (Okada et al., 2011), it was concluded that when Estimated Features Method applied to SVM, Bayes and Decision Tree, SVM algorithm outperforms the Naive Bayes Kernel Estimation algorithm and Decision Tree algorithm by respect to %97.2, %92.3 and %69.4 overall accuracy [18].

In a study using RF and DNN methods on “Malware Detection”, it was concluded that RF had a higher success rate than DNN in general, and multi-layer DNN was more successful in some features [19].

Finally, according to the citations made in a study using machine learning in cancer detection research; Among the SVM, KNN, Decision Tree, NB methods, it was seen that the SVM classification method achieved the most successful result with an accuracy of 97.3%.

In another quote from the same research; Among the SVM, BN, RF methods, the method in which the highest accuracy rate and the ROC area is calculated with the most optimal performance has been reported as the RF technique.

As stated in another study cited by the research; When comparing various ML methods for breast cancer detection, it was stated that among ANN, SVM and DT, SVM achieved the best performance with the highest accuracy and lowest error rate[20].

Figure 1. Device that tests the strength of concrete obtained in today's technology.(the figure from our School)

1.3.2. Concepts

The language, programs and libraries we use are all open source. For this reason, we will not pay any fees for the algorithms and for using the necessary programs to implement the algorithms.

"These results show that the ANN model is able to give a more precise and high accuracy prediction of blast furnace slag compressive strength, compared to the multiple regression model[11]." Because of this explanation, we can say that ANN performs better than Multiple regression analysis.

“The findings showed that ANN achieved the best accuracy for all performance measures because it can provide the most optimal performance when viewed from the four indicators[21].” It is clear from the research here that ANN is successful compared to SVM, albeit with a small margin, in concrete compressive strength tests.

Table 1. Comparison of the three conceptual solutions.

	ANN	SVM	Regression
Performance	high	medium	low
Cost	low	low	low

According to the research we have done and the studies we have examined, we have seen that ANN has achieved much more successful results. However, we evaluated this value as low because the working principle of ANN and the cost of the model are higher than the others. Studies with SVM can be described as successful according to regression models. Here we made our evaluations between ANN and SVM Classification models and Regression models. All evaluations were interpreted in line with the literature studies.

1.4. Architecture

The Civil Engineering team will obtain the data containing the concrete mix quantities and strength test results from the Ready Mixed Concrete Plant, prepare these data in a format that can be used in ML algorithms and clearly define the mix content.

A prediction model will be produced using the ML algorithm with the obtained data set. This model will be supported by optimization algorithms to produce faster and more accurate predictions.

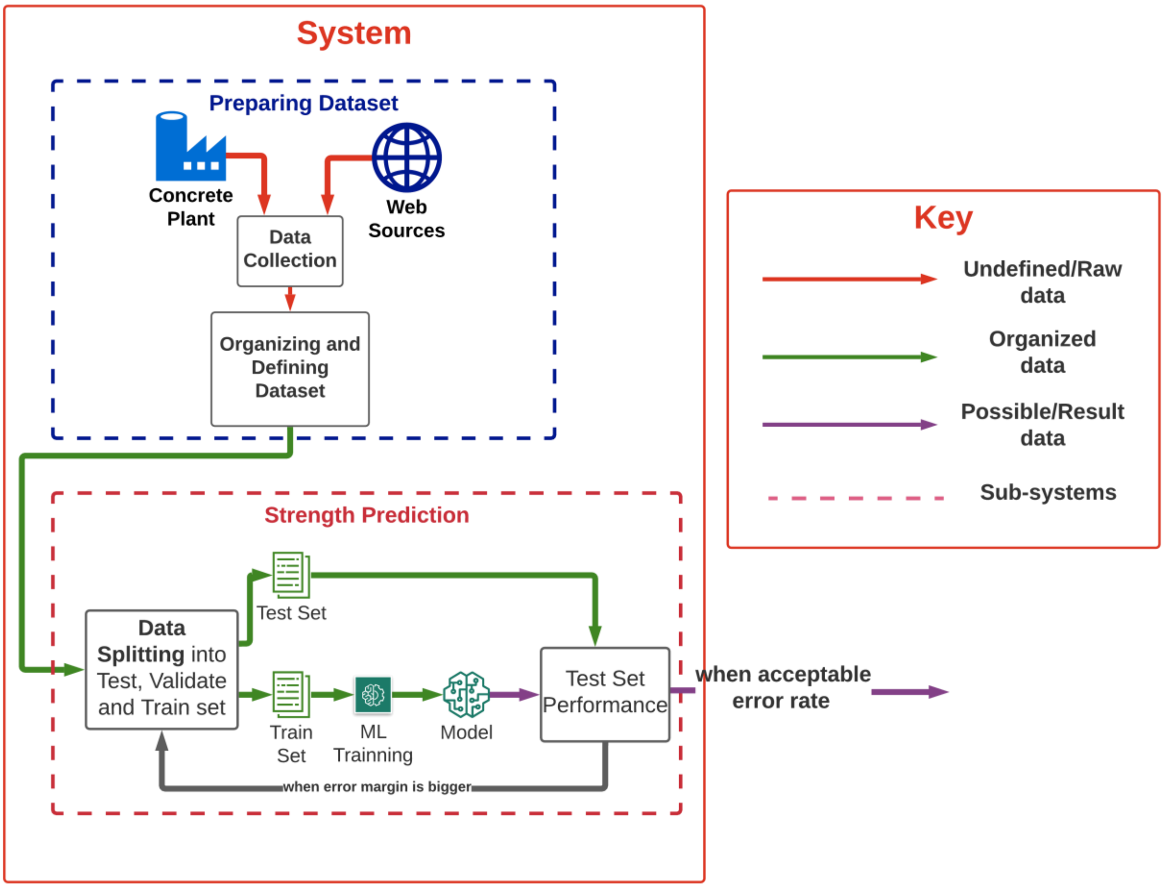


Figure 2. Interface diagram for the system.

2. WORK PLAN

2.1. Work Breakdown Structure (WBS)

First, it will collect the data from the concrete compressive strength tests of the civil engineering students and transmit it to the computer engineering students in the most descriptive way. It will also convey the materials and quantities used in the mixture of concrete.

Computer engineering students will use this data to determine the most successful concrete mix design using artificial intelligence and determine the resulting compressive strength database. Computer engineering students will share this database with civil engineering students. Civil engineering students, on the other hand, will start the experimental process by purchasing the necessary materials for the most successful concrete mix. After these processes, the results of artificial intelligence will be compared with the experimental results of the compressive strength test.

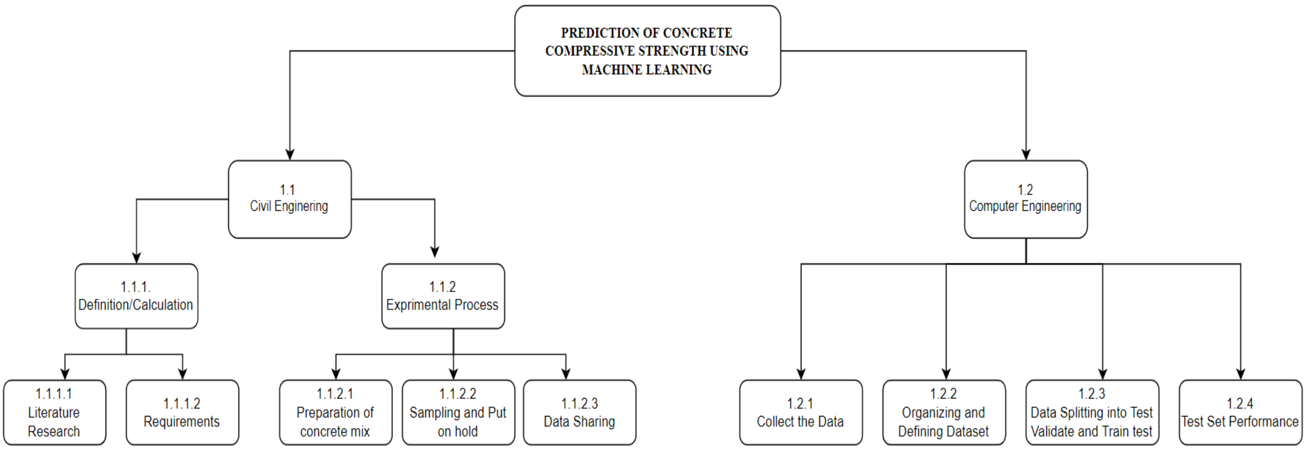


Figure 3. Work Breakdown Structure for the project

2.2. Responsibility Matrix (RM)

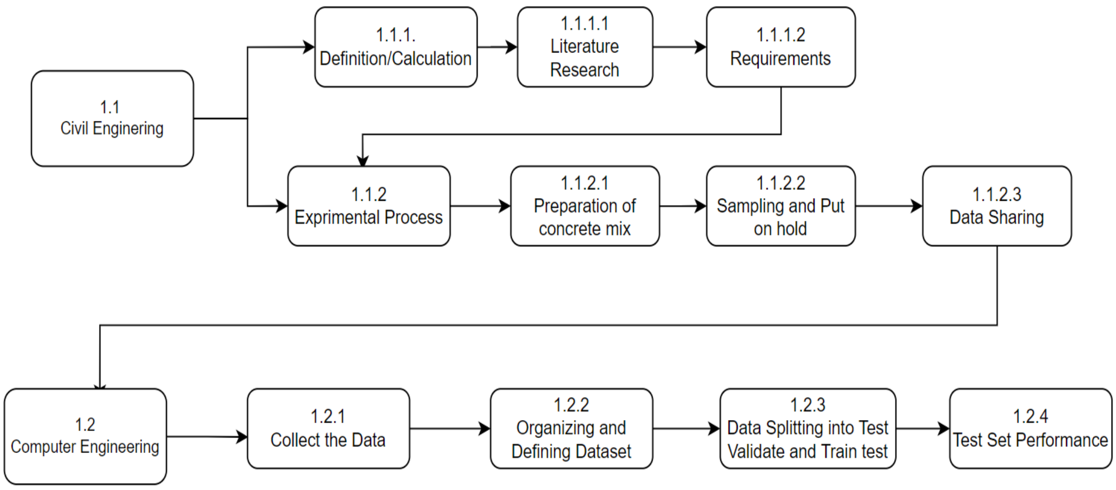
Table 2. Responsibility Matrix for the team

TASK	ÖMER KARA	RAMAZAN AFŞİN	ELİF İLTEROĞLU	ALİ YAĞIZ İLBAN	GÜLBAHAR EROL	EREN GÜNEŞTAŞ
MECHANICAL	R	S				
TESTING	S	R		R		S
CONTROL		S	R	R	S	S
RESEARCH	S		R		R	S
OPTIMIZATION				S		R
PLANNING	R	S	S		R	
REPORTING	S	R	S	R	S	S
INTEGRATION	S	S	R	S	S	R

R=RESPONSIBLE S=SUPPORT

2.3. Project Network (PN)

Figure 4. The project networks



2.4. Gantt chart

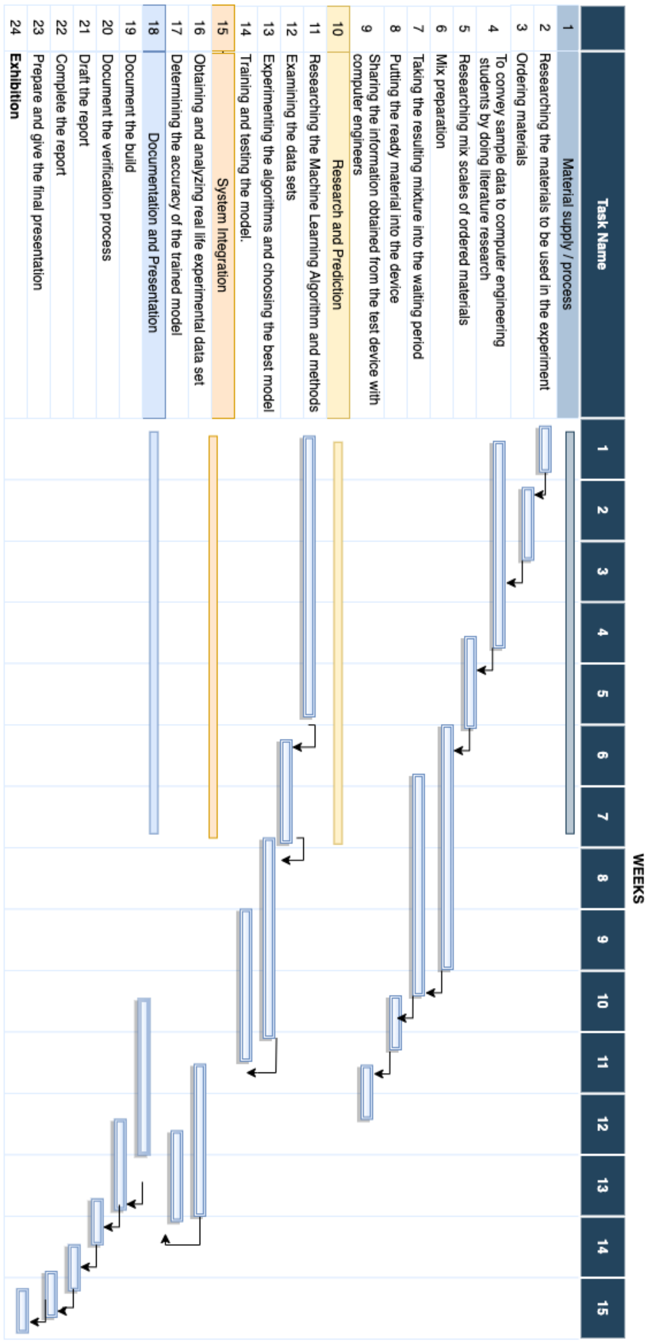


Figure 5. Gantt chart

2.5. Costs

Table 3. Cost Table

CONCRETE COST	
CEMENT (1KG)	20 TL
AGGREGATE(1KG)	100 TL
WATER(1 KG)	3 TL

Computer Engineering team has no need about developing Capstone Project.

2.6. Risk assessment

Table 4. Risk Matrix

RISK LEVEL		Severity of the event on the project success		
		Minor	Moderate	Major
Probability of the event occuring	Unlikely	VERY LOW	LOW	MEDIUM
	Possible	LOW	MEDIUM	HIGH
	Likely	MEDIUM	HIGH	VERY HIGH

VERY LOW	This event is very low risk and so does not require any plan for mitigation. In the unlikely event that it does occur there will be only a minor effect on the project.
LOW	This event is low-risk; a preliminary study on a plan of action to recover from the event can be performed and noted.
MEDIUM	This event presents a signficant risk; a plan of action to recover from it should be made and resources sourced in advance.
HIGH	This event presents a very signficant risk. Consider changing the product design/project plan to reduce the risk; else a plan of action for recovery should be made and resources sourced in advance.
VERY HIGH	**This is an unacceptable risk. The product design/project plan must be changed to reduce the risk to an acceptable level.**

Table 5. Risk assesment

Failure event	Probability	Severity	Risk Level	Plan of Action
Covid-19 lockdown	**Likely** It seems likely tahat there will be at least limited Access to school,and some team members might not want to travel.	**Moderate** This will make it more difficult to build and integrate subsytems.	**HIGH**	Communication can be achieved through online interviews.
Defect of test machine	**Possible**	**Moderate** If the machine is defective, it will prolong the process as it needs to be re-adjusted.	**MEDIUM**	Re-setting the machine
Weather	**Possible** Weather conditions can always change.	**Minor** Weather conditions will not affect our project much.	**LOW**	Weather conditions may vary independently.
Material quality received	**Possible** The materials received may be of poor quality.	**Moderate** Poor quality materials can undermine the accuracy of the experiment.	**MEDIUM**	If the materials are too poor to be used, they can be bought new.
Overfitting and underfitting problems	**Possible**	**Moderate**	**MEDIUM**	Train the model with different parameters

3. SUB-SYSTEMS

3.1. Civil Engineering

Concrete compressive strength test is prone to human errors and is time consuming, since it can be applied at the earliest 7th day of concrete pouring. In order to avoid this loss of time, it is necessary to obtain the most successful concrete mix. In this project, we will learn the materials and quantities required for the most successful concrete mixture by evaluating the data obtained by performing multiple concrete compressive strength tests in an artificial intelligence environment.

3.1.1. Requirements

Civil engineering students will first buy materials to be used for concrete mix. Then the concrete will begin the compressive strength test processes. He will give the data he obtained in this process to computer engineering students. Finally, the most successful concrete mix will be prepared according to the data obtained from the computer students and a test trial will be made.

3.1.2. Technologies and methods

As we know, technology is developing day by day and this development affects many business areas today. One of them is the construction industry. The discovery of new materials that develop and continue to develop every year and the quality of the materials used change. In our project, we will have the most solid concrete ratios thanks to the synthesis of the data obtained as a result of the durability test of the concrete obtained thanks to the method computer students and the information obtained.

3.1.3. Conceptualization

In the concrete compressive strength test, a force is applied to the cube (20x20) or cylindrical (15/30 cm) sample with a press device, without impact and at constant values. This process is continued until the sample is broken, and the force is not applied when the fracture occurs. The amount of force that causes the breakage of the sample is proportional to the cross-sectional area and the compressive strength is found.

Before the core sample is taken, Schmidth Hammer test and Ultrasound tests can be done. After the sample is taken, the correlation between the experiments is found by establishing a correlation between the experiments. Concrete quality can be determined with Schmidt Hammer and Ultrasound without damaging other carrier elements.

In addition to these, Bending Tensile Strength and Splitting Tensile Strength tests can also be performed. However, in the bending tensile strength test, it is not equivalent to the compressive strength test performed on the samples, and it is not a test where we can reach the concrete class. The Split Tensile Strength Test can be used for core samples, and the required amount of load is determined to break the loaded surface with a special system.

3.1.4. Physical architecture

After the materials we ordered arrive, we move on to the construction phase. The main materials we will use in the construction phase are; Aggregas, Cements, Water. After mixing these materials, we wait for the concrete to reach its solid state. The collection of the data obtained after the solid-state concrete enters the PCE device and the presentation of the collected data to the computer engineering students.

3.1.5. Materialization

- Cement whose main raw materials are limestone and clay and which is used for bonding mineral parts (sand, gravel, brick, briquette, etc.).
- Mixing water used in concrete production.
- Materials such as sand, gravel, crushed stone used in concrete production are aggregate.
- Fine aggregate (such as sand, crushed sand)
- Coarse aggregate (such as gravel crushed stone)
- Superplasticizer that provide the same consistency or workability in concrete with less water.
- Slag, fly ash, silica fume, stone flour, etc. stored in silos in ground powder form such as cement. various mıneral admixtures.

3.1.6. Evaluation

The test methods applied to find the compressive strength are simpler than the methods applied to find other strength types.

• The compressive strength value of concrete is taken as a basis in the design of almost all structures. In many structures, it is assumed that the concrete will not be exposed to different loads such as tensile, bending, and fatigue, and calculations are made by assuming that the most important loads on the concrete are compressive loads.

• There is an approximate correlation between the compressive strength of concrete and its tensile and bending strengths. Therefore, if the compressive strength is known, an idea can be obtained about the magnitudes of other types of strengths.

• Knowing the compressive strength provides qualitative information about other (durability) properties of concrete. For example, high compressive strength indicates low water permeability and high durability in concrete.

3.2. Computer Engineering

This project aims to save time needed by estimating compressive strength testing. This subsystem includes deciding to use a learning algorithm in line with the information obtained from the researches, training and testing this algorithm with the data set obtained from Civil Engineering department.

3.2.1. Requirements

Functional Requirements

Before we train our model with the data set we have obtained, we need to normalize our data set in the pre-processing phase. Normalization will scale the feature defined as quantity in the data set.

We need data to train our model. However, the trained model should be tested with a data set that has not been used for training before. The reason for this is to measure how the trained model will react to a data it has never encountered before.

The determined supervised learning algorithm may have to be in a back propagation structure in order to eliminate its own errors.

In order to define parameter values of the algorithms determined for the learning algorithm (such as the network amount for the hidden layers or the amount of the cluster for the cluster algorithms.). It should be examined and analyzed with an optimization algorithm or functions such as a heat map, correlation table etc.

Non-Functional Requirements

An error percentage of the model tested with the test set must be determined and this determines the performance of the algorithm.

The performance determined for the learning algorithm should be tested with real experimental results and should provide the specified performance.

3.2.2. Technologies and methods

Since we will not store a very large data, we can keep these informations that we will evaluate in Excel or simply csv document. However, if the data size is larger than we expected and/or we need more data and include extra data sets from external sources, we will store the data by using Azure Database. In any case, the parameters to be used will be as indicated in the table below.

Table 6. Data parameters of data set

Data Parameters
Cement (kg/m^3)
Mineral Mixtures (kg/m^3)
Superplasticizer (kg/m^3)
Water (kg/m^3)
Coarse Aggregate (kg/m^3)
Fine Aggregate (kg/m^3)
Age of Concrete (days)

In machine learning, it can be summarized as the evaluation and interpretation of the data set, determining the optimal result, testing it and counting it as successful if it has a valid margin of error. We will usually do the development of Machine Learning over Python language and/or C++.

3.2.3. Conceptualization

Figure 1.4 is designed assuming that the Neural Network technique will be used to create the prediction model. The error margin iteration decision, shown in Figure 1.4, can be achieved in a certain performance situation or by the amount of iteration given as a parameter. Concretes with different characteristics may adversely affect our estimation results. For this, the concretes defined in the data set can be clustered with the K-means clustering algorithm and a different neural network model can be produced for each cluster. In case the concrete compressive strength prediction capability of the Neural Network structure is insufficient, Multi Regression Analysis method or Support Vector Machine techniques can be used to retrain our Estimation Model. A regression analysis can be done.

3.2.4. Physical architecture

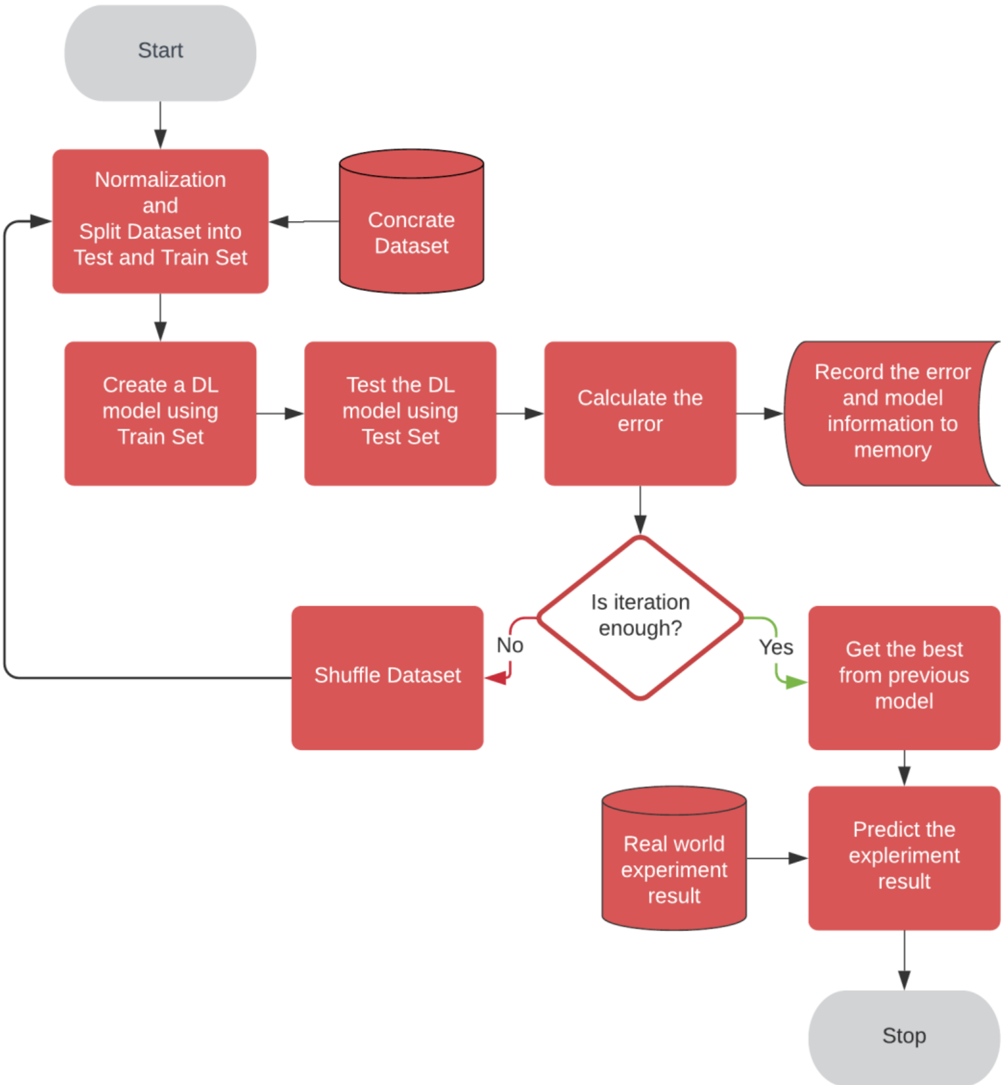


Figure 6. Software Architecture System

3.2.5. Implementation

After examining the data from the other team, a model will be created with the determined algorithms. It is aimed to reach the best estimate by comparing the results obtained. At the end of the project, the model with the best estimate will be compared with the experimental result data.

3.2.6. Evaluation

We will determine the success of the system by ensuring the accuracy of the model created by the ML that we customize/write ourselves without receiving the data from the other department, with the data sets we found from the Internet and from different sources, with the same set that we teach. A performance value with low standard deviation and high predictive value obtained with test data was created for our concrete durability prediction model developed with artificial learning.

In order to determine the accuracy of the model formed, it will be tested by taking 15% of the data set and mixing it after each piece is taken. If our model has an accuracy of over 80%, it will be successful. Our Prediction Model has performance metric scores. A real experiment should be done for this metric score evaluation. The concrete is crushed in the strength tester, creating a real concrete. The results obtained are estimated by the model. The result of the estimation is explained and interpreted whether it is provided by the performance metric scores or not.

4. INTEGRATION AND EVALUATION

4.1 Experiment Stage

First of all, we determined the tasks that we need to do to complete our project. Firstly, we collected 369 data by doing literature research to send data to computer engineering students , and converted this data to csv format on the computer and sent it . Secondly , we decided to have a meeting with our project adviser Mrs. Şanal and make 12 concrete mixes . We identified and supplied the missing materials in the civil engineering laboratory in our school so that we could make concrete mixes. After completing the materials, we obtained the concrete by looking at the mixing ratios we determined in the civil engineering laboratory to make concrete. We took the concrete out from the molds the next day and put it in water to be kept for 7 , 14 , 28 days . After the days are completed, we will put the concrete we have made to the compressive strength test.

4.1.1. Materials

Used materials;

Cement; Cement is basically defined as a hydraulic binder material obtained by grinding a mixture of natural limestone stones and clay after heating at high temperature.

Agrega; Aggregate is a mixture of sand and gravel that forms the raw materials of concrete.

Fine aggreggate; Is the aggregate obtained from natural sand, crushed sand (fine gravel) or their mixture and passed through a square mesh sieve with 4 mm mesh. Fine aggregate grains should be hard and solid.

Course aggreggate; It is the aggregate obtained from natural gravel, crushed stone (coarse stone chips) or their mixture and remaining on the square hole sieve with 4 mm mesh.

Water; Mixing water used in the production of concrete has two important functions: Turning dry cement and aggregate into a plastic, workable mass. To provide the hardening of the plastic mass by chemical reaction with cement.

Devices used

Sample Molds; Sample molds for determining the concrete compressive strength are in the form of cubes or cylinders. Generally, cube samples with a side size of 150 mm are preferred. Top surfaces of cylindrical specimens must be capped or trimmed. Therefore, cube samples are more practical.

Concrete mixer; A concrete mixer or concrete mixer is a device that homogeneously combines aggregates such as cement, sand or gravel with water to form concrete.

Weight; Analytical balance is a balance class designed to measure small masses in the sub-milligram range.

Water mold; Container filled with water used to hold concrete samples

Concrete pressure device; Concrete compressive strength machines are test devices that work by compressing hydraulic fluids.

4.1.2. Evaluation

Before starting our experiments, we arranged a meeting with our adviser, Mrs.Şanal. We talked about the details of our project, how we should proceed in our project, where we should start. In this process, we conducted a literature search and conveyed approximately 370 sample data that we had collected to computer engineering students.

In our experiments, we considered 3 variables while determining the compressive strength of concrete.

1. Water/Cement ratio
2. Whether the mixtures contain superplasticizers
3. Waiting time of the concrete in the water pool after it comes out of the mold (days)

These variables play an important role in measuring the compressive strength of concrete. The main purpose of these experiments is to ensure that we reach the strongest concrete mixture, whichever variable we change.

We decided to do 12 experiments in total with our project adviser Mrs.Şanal, whose water/cement ratios are certain according to these 3 variables.

Table 7. Table of experiments to be carried out according to variables

<table>
<tr><th>W/C ratio</th><th>Superplasticizer</th><th>Time(Days)</th></tr>
<tr><td rowspan="3">0,4</td><td rowspan="3">+</td><td>7</td></tr>
<tr><td>14</td></tr>
<tr><td>28</td></tr>
<tr><td rowspan="6">0,5</td><td rowspan="3">+</td><td>7</td></tr>
<tr><td>14</td></tr>
<tr><td>28</td></tr>
<tr><td rowspan="3">-</td><td>7</td></tr>
<tr><td>14</td></tr>
<tr><td>28</td></tr>
<tr><td rowspan="3">0,6</td><td rowspan="3">-</td><td>7</td></tr>
<tr><td>14</td></tr>
<tr><td>28</td></tr>
</table>

Considering the superplasticizer variable for these 12 experiments;

1 with a Water/Cement ratio of 0.4,

2 with 0.5 Water/Cement ratio,

1 with a Water/Cement ratio of 0.6,

We had to prepare a mix recipe. We obtained the prescriptions by doing the necessary research. We presented these mixture recipes we found to Irem, got his approval, and started the experiments.

Table 8.Mixing ratios of concrete mixes to be made

	1 dm3 (1 lt)					
	W/C	Cement (g/m3)	Water (g/m3)	Superplasticizer (g/m3)	Coarse Aggregate	Fine Aggregate
M1	0,6	380	228	0	932	670
M2	0,4	500	200	3	1120	613
M3	0,5	436	218	0	840	719
M4	0,6	380	190	7	1013	730

4.1.2.1 Concrete Making in the Laboratory

In order to obtain a quality concrete, it is necessary to blend the components in the right ratio and with the right method. In fact, concrete consists of three main components; Water, cement and aggregate. But in this experiment we also use superplasticizer, which is variable additive in some of the blends.

While preparing the concrete mix;

1. First, the proportions of the components to be used in the mixture are accurately measured and placed in the mixer's chamber.

Figure 7. Weighing of Materials

2. The components placed in the chamber are mixed well with the help of a mixer. Its consistency is checked and if its consistency is in place, it is taken from the machine.

Figure 8. Mixing the Ingredients

3. The sample mold is well lubricated so that the concrete comes out of the mold smoothly and evenly.

Figure 9. Lubrication of the Mold

4. After the mold is lubricated, the concrete mix is poured into the mold gradually. While pouring little by little, an iron rod is inserted into the mixture and removed so that the mixture spreads evenly in the mold.

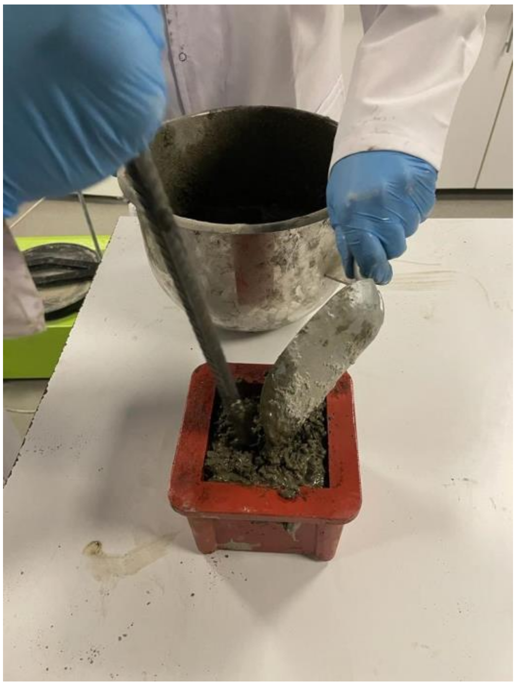


Figure 10. Pouring Concrete into Mold

5. After the mixture is poured into the mold, it is manually tapped on the exterior of the mold with the help of a wedge to give a vibrator effect.

Figure 11. Vibration of Concrete

6. The mixture is kept in the mold for 1 day.
7. After it comes out of the mold, it is kept in the water pool for the specified time.

Figure 12. Putting the Samples in the Pool

8. After removing it from the water pool, it is put into the press machine and the results of the compressive strength are taken.

Figure 13. Putting the Concrete into the Testing Machine

4.1.2.2 Preparation Process of the Mixture

Before starting the experiments, we reconfirmed whether the necessary materials were available in our laboratory. We bought 3 10x10x10 cube molds and superplasticizers. After starting the experiments;

→ In the first week, we prepared the mixtures without superplasticizer with a water/cement ratio of "0.5" and "0.6" and a mixture with a water/cement ratio of "0.4" containing superplasticizer and poured them into molds to wait 28 days in the water pool.

→ In the second week, we prepared the mixtures with water/cement ratio of "0.5" and "0.6" without superplasticizer and the mixture with water/cement ratio of "0.4" containing superplasticizer and poured them into molds to wait for 14 days in the water pool.

→ In the third week, we prepared the mixture containing superplasticizer with a water/cement ratio of "0.5" and poured it into molds to wait 28 days and 14 days in the water pool.

→ In the fourth week we prepared all the mixes and took the samples for each of them to sit in the water pool for 7 days and poured them into the molds.

→ After the concrete samples come out of the molds, they are thrown into the waiting pool and will be put into the press machine after the waiting time in the pool is over. We will also convey the results of the concrete samples on the press machine to the computer engineering students.

→ Speaking for now, all our molds will be ready on Monday and we will put the samples on the press machine on Monday. On 16.05.2022, we will share our values with other group members.

Table 9.Experimental Process

<table>
<tr><th>Week (affter starting the experiments)</th><th>W/C</th><th>Superplasticizier</th><th>Days</th></tr>
<tr><td rowspan="3">First week</td><td>0,4</td><td>+</td><td>28</td></tr>
<tr><td>0,5</td><td>-</td><td>28</td></tr>
<tr><td>0,6</td><td>-</td><td>28</td></tr>
<tr><td rowspan="3">Second week</td><td>0,4</td><td>+</td><td>14</td></tr>
<tr><td>0,5</td><td>-</td><td>14</td></tr>
<tr><td>0,6</td><td>-</td><td>14</td></tr>
<tr><td rowspan="2">Third week</td><td rowspan="2">0,5</td><td rowspan="2">+</td><td>14</td></tr>
<tr><td>28</td></tr>
<tr><td rowspan="4">Fourth week</td><td>0,4</td><td>+</td><td>7</td></tr>
<tr><td rowspan="2">0,5</td><td>+</td><td>7</td></tr>
<tr><td>-</td><td>7</td></tr>
<tr><td>0,6</td><td>-</td><td>7</td></tr>
</table>

4.1.2.3 Compressive Strength Test Results of the Mixture

The M1, M2, M3, M4 concrete mixtures given in Table 8 were removed from the water pool and introduced into the concrete compressive strength test machine . A total of 12 mixtures were made. The test results of these concrete mixtures are given in the graphs below.

Figure 14.Breaking of Concrete in the Testing Machine

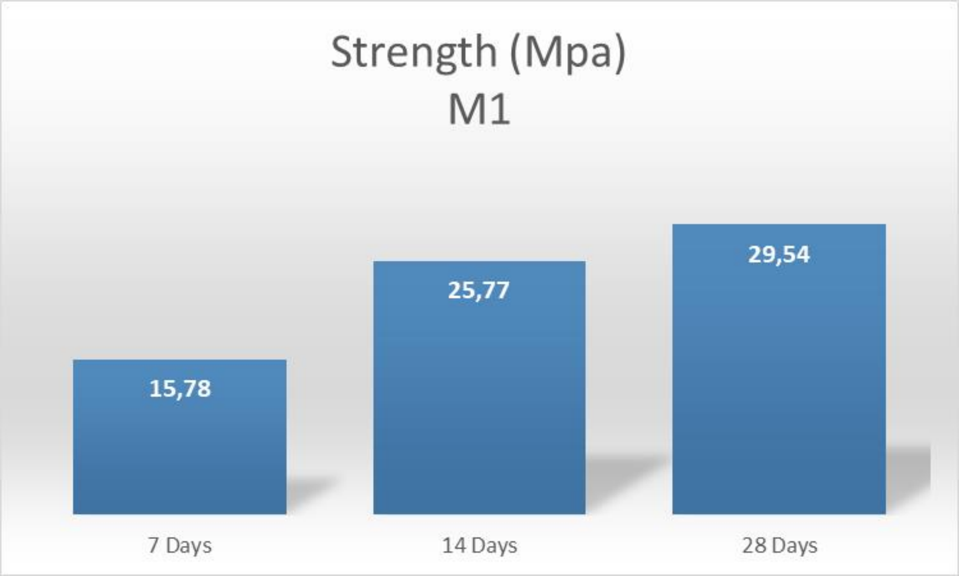


Figure 15.Compressive strength test results of M1 concrete mix

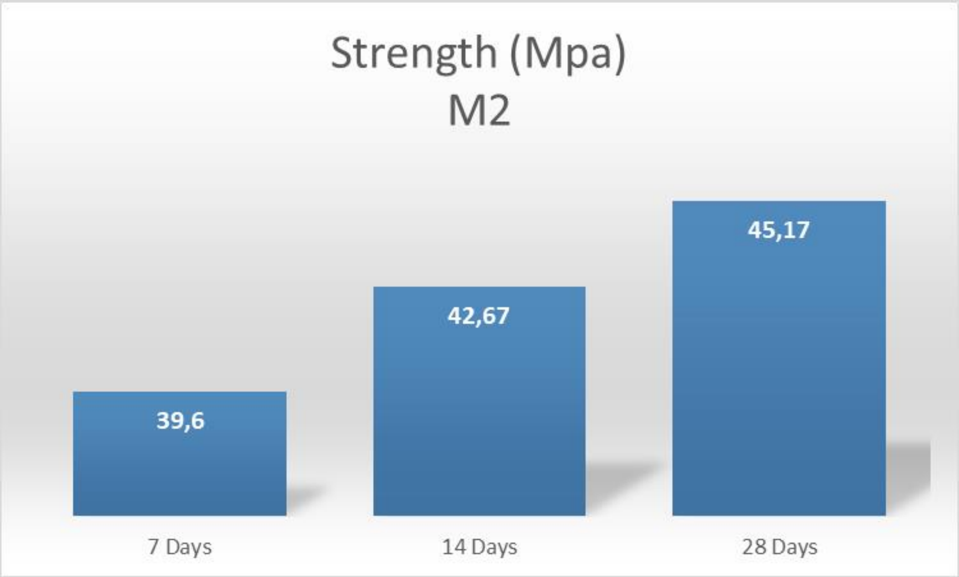


Figure 16.Compressive strength test results of M2 concrete mix.

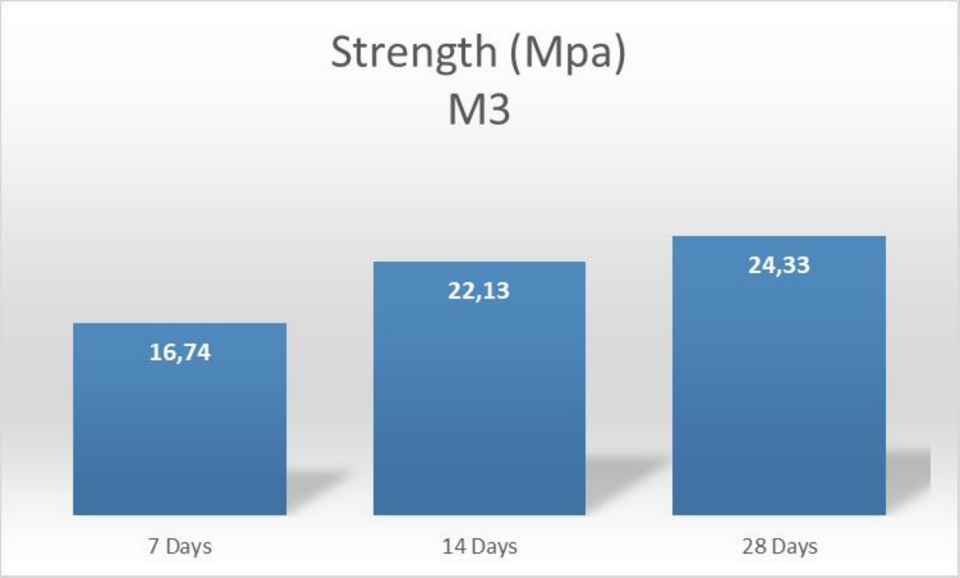


Figure 17.Compressive strength test results of M3 concrete mix.

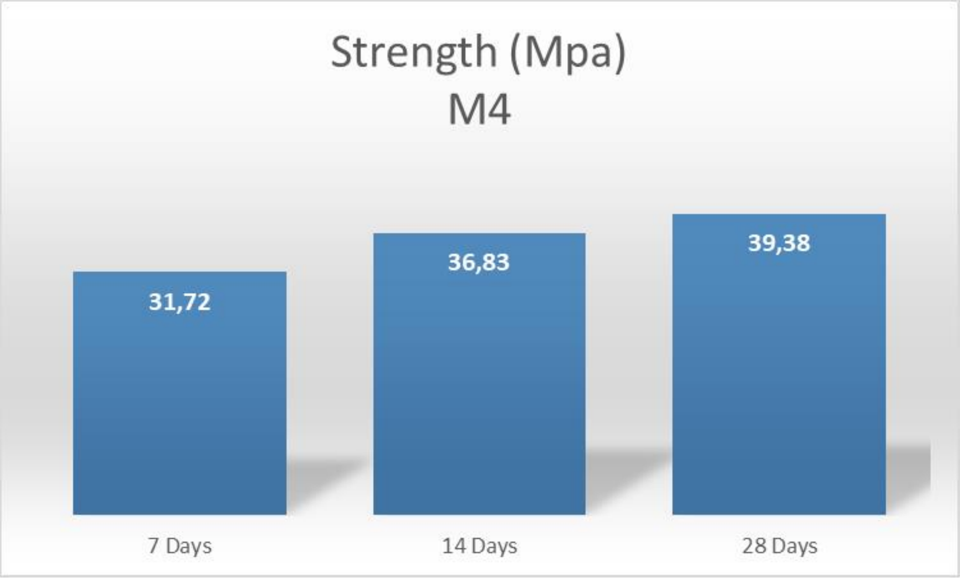


Figure 18.Compressive strength test results of M4 concrete mix.

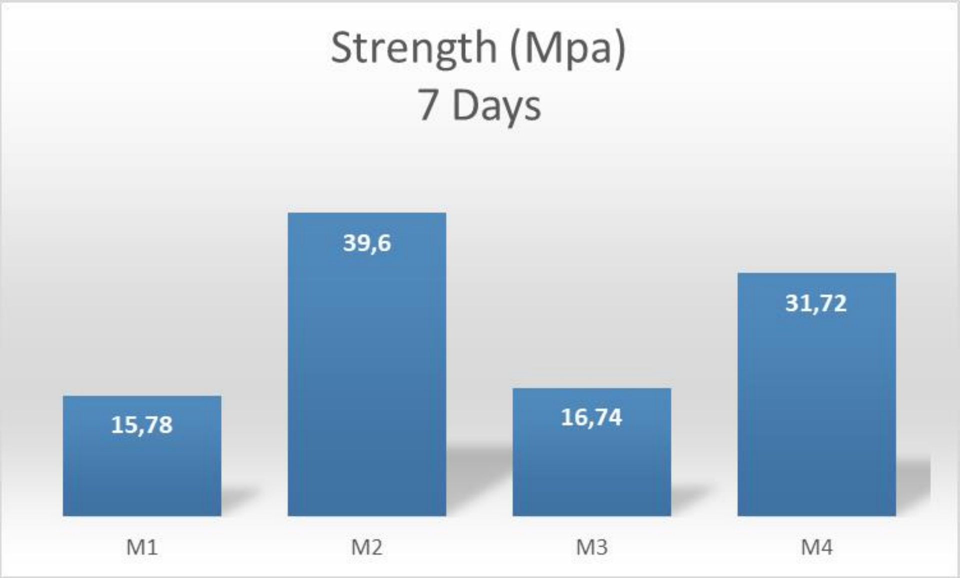


Figure 19.7-day test results of mixtures.

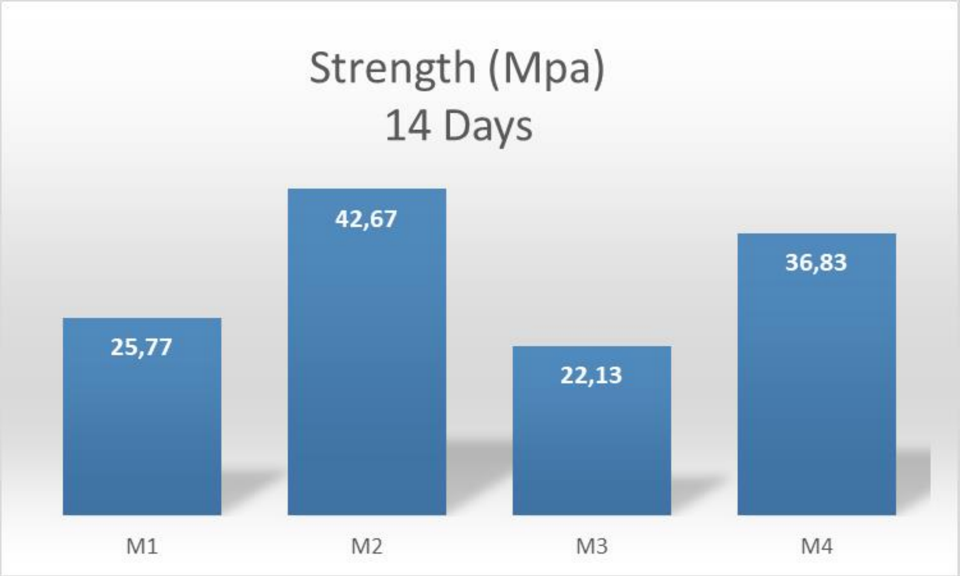


Figure 20.14-day test results of mixtures.

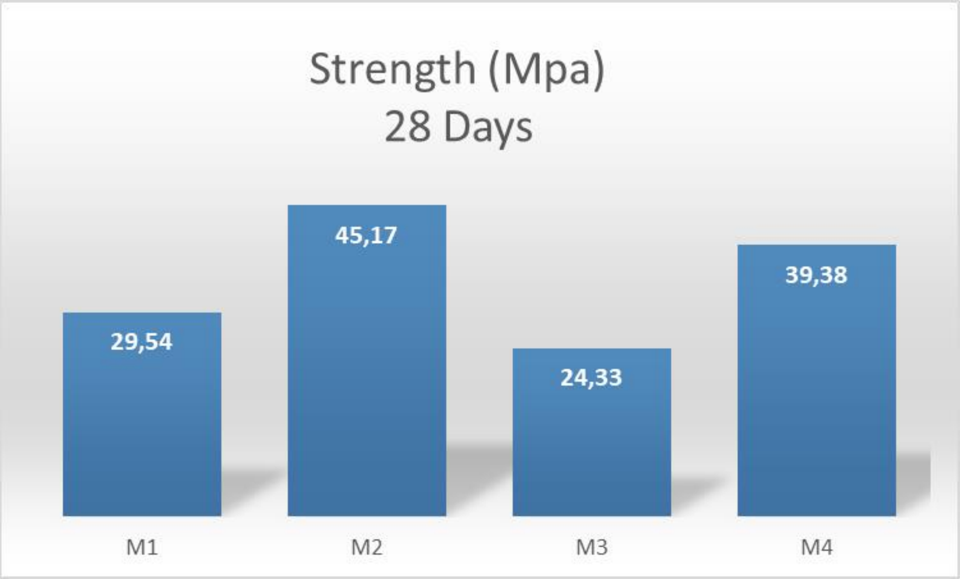


Figure 21. 28-day test results of mixtures.

When these graphs were examined, it was seen that the strength of the concrete increased as the waiting time of the concrete increased. Based on these graphics, it was seen that the effect of superplasticizer on concrete strength test was great. It has been observed that superplasticizer increase the strength of concrete in a short time.

We have reached many results by examining many data in the total of 366 experiments we gave to computer engineering students. These results are:

1. The most prominent factor in the variation of values was the amount of water and cement.
2. Apart from these parameters, the other factor affecting the concrete pressure the most was the increase in the number of days.
3. When the effect of the additives used on the experiments was examined, reductions in the ratio of water and cement were observed. The conclusion we will draw from here

is that if we use additives in the mixtures, our water and cement amounts have been reduced.

4. Aggregate usage was kept constant in all of our examined experiments. Decreases were observed in the concrete pressure tests of the experiments, in which there was a decrease in aggregates.
5. High amount of water and cement ratio gives us the highest data in the durability tests reached in these examined experiments.

As a result, it allowed us to have a little idea as a result of the tests examined.

4.2 Machine Learning Modelling

It was decided to work on the Regression model according to the literature research and the studies made with the data obtained from the Civil Engineering team. We have a data set of 368 rows. 7 columns for data(ingredients of mixture), last column for result(strength of cement). The data is homogeneous and does not require much intervention. Separate models were created on these three methods.

	count	mean	std	min	25%	50%	75%	max
Cement	368.0	329.424185	94.890475	136.10	275.100	311.250	376.000	540.00
FlyAsh	368.0	53.646196	59.467745	0.00	0.000	0.000	124.425	168.30
Water	368.0	182.695380	19.130836	140.00	169.975	186.000	192.000	228.00
Superplasticizer	368.0	4.382337	5.808848	0.00	0.000	0.000	9.500	28.20
CoarseAggregate	368.0	998.323913	75.760593	801.00	956.450	994.600	1053.950	1125.00
FineAggregate	368.0	778.994293	81.597854	77.60	754.225	784.000	822.050	905.40
Age	368.0	48.111413	67.367115	1.00	7.000	28.000	56.000	365.00
Strength	368.0	34.660272	14.681839	6.27	23.210	33.655	42.485	79.99

Figure 22. Data Set Distribution

If the mean value is large in the median, it can be interpreted as positive skewness. In this case, common values are overloaded and may mislead the model. Conversely, a mean value lower than the median may cause common values to be ignored. Based on this, there seems to be a skewed distribution in the Fly Ash and Superplasticizer columns.

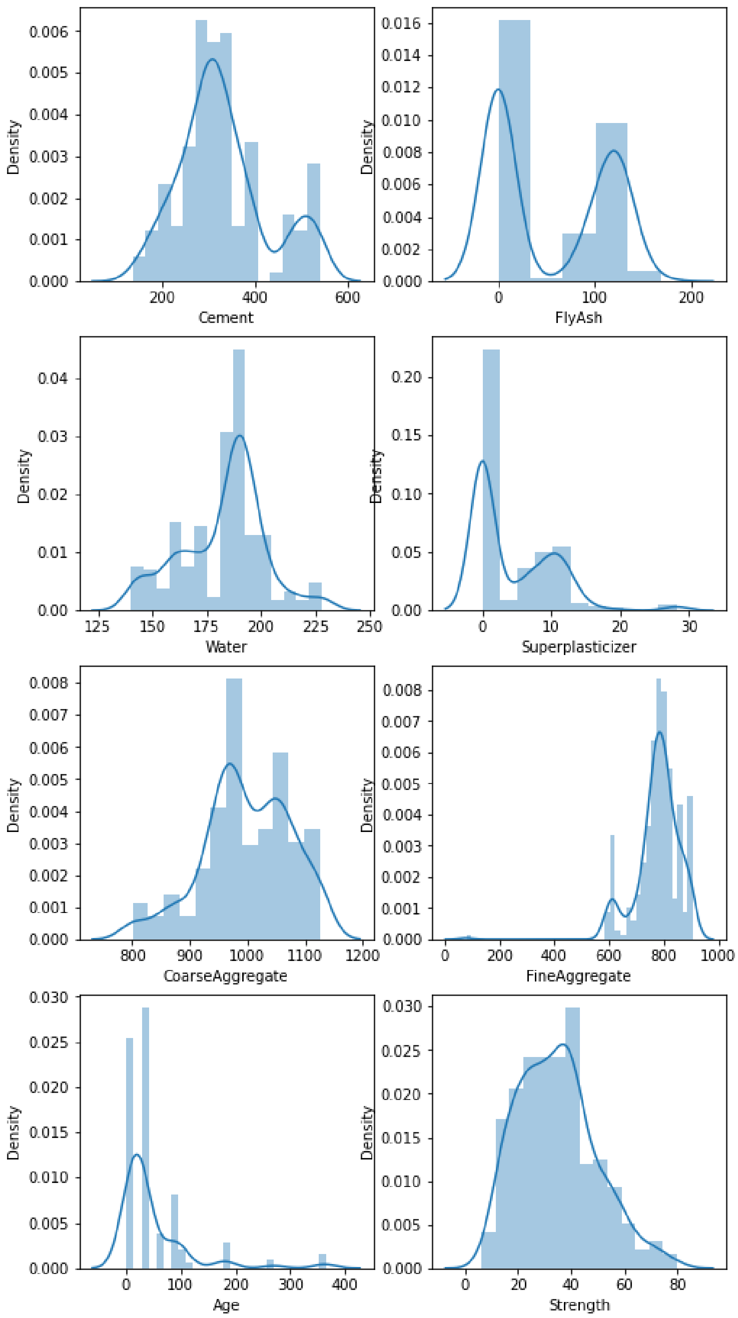


Figure 23. Ingredients' Density Graph

Cement, Water and Coarse Aggregate have a normal distribution, while other parameters have a skewed distribution. We can interpret outliers by looking at skewed distributions. We organized these outliers with the quartile method. Since skewed distributions affect the models negatively, editing should be done, but the skewness here is not at a level to affect the model. This is why we only need to edit outliers.

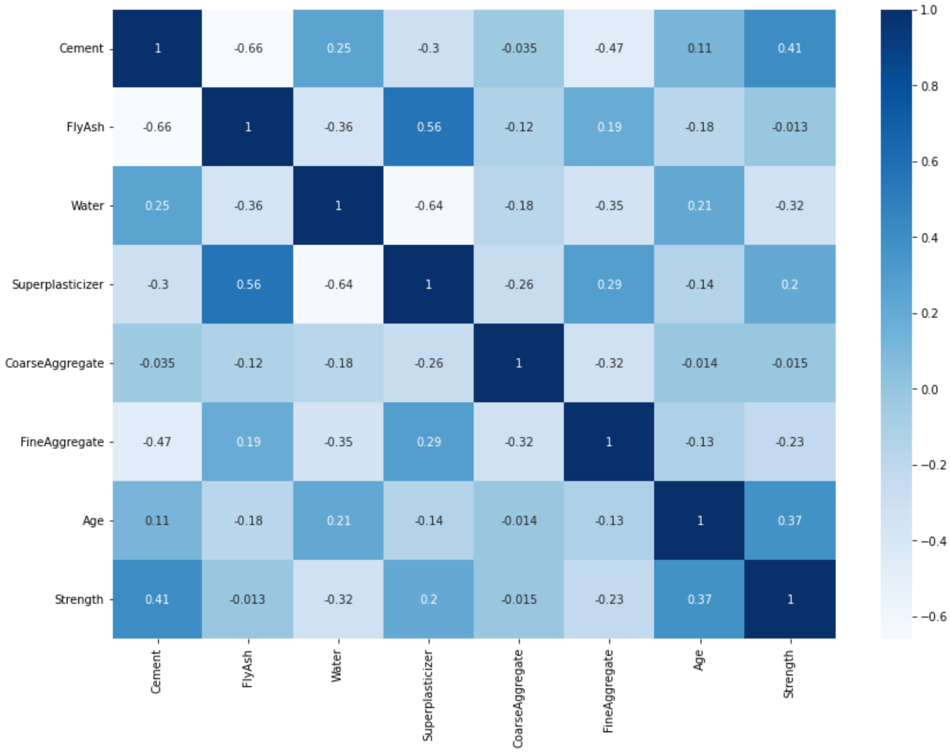


Figure 24. Correlation Matrix of Ingredients

Except for cement, there is no high correlation between the compressive strength and the data. Age and Superplasticizer are two other properties that are strongly associated with Compressive Strength. There are negative correlations between water and superplasticizer. These parameters have opposite effects on each other. In addition, Cement and Fine Aggregate have a negative correlation. Since a correlation value of 90% will manipulate the model and negatively affect the model, it is necessary to eliminate the parameters, but since these correlations will not cause negative effects on model training, no adjustments have been made.

4.2.1 Implementation

Regression analysis is a reliable method for determining which variables have an impact on a topic of interest. It is used to model the relationship between dependent (target) and independent (predictive) variables with one or more independent variables. The process of performing a regression allows you to confidently determine which factors are most important, which factors can be ignored, and how these factors affect each other. The purpose of the regression model is to construct a mathematical equation that describes y as a function of the variables x. This equation can then be used to predict the outcome (y) based on the new values of the prediction variables (x).

It helps to find the correlation between the variables and allows us to predict the continuous output variable based on one or more predictive variables. It is mainly used for forecasting, time series modeling and determination of cause-effect relationship between variables.

Regression analysis helps us understand how the value of the dependent variable corresponding to an independent variable changes when other independent variables are held constant.By doing regression, we can confidently determine the most important factor, the least important factor, and how each factor affects other factors.

4.2.1.1 SVM

"Support Vector Machine" is a machine learning method used to optimally separate two or more data sets from each other. Although it is generally used in solving classification problems, it is also a preferable method in solving regression problems.

$$\hat{y} = \begin{cases} 0 \ if \ w^T.x + b < 0, \\ 1 \ if \ w^T.x + b \geq 0 \end{cases}$$

Equation 1. SVM Formula

SVM is aiming to draw a line (or plain) using the boundary reference points determined between the clusters, which is used to separate two clusters from each other with higher margin. The larger the distance between the separator and the reference regions (in other words margin), the easier the data can be separated from each other and the lower the error rate.

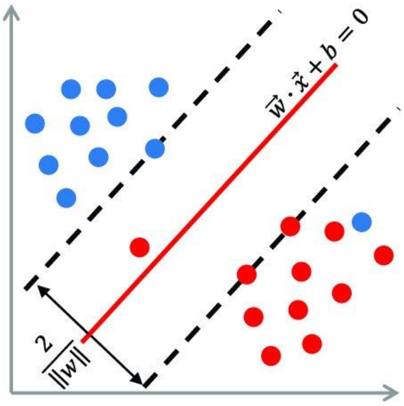


Figure 25. Linear Support Vector [25]

SVM includes many parameters. There are most important 2 parameters, Kernel is for the shape of the divider, C is for the regularization parameter.

Kernel is used when separating multidimensional data and when it is necessary to use a special shape to separate it. Especially Figure 18. reinforces the situation we have stated.

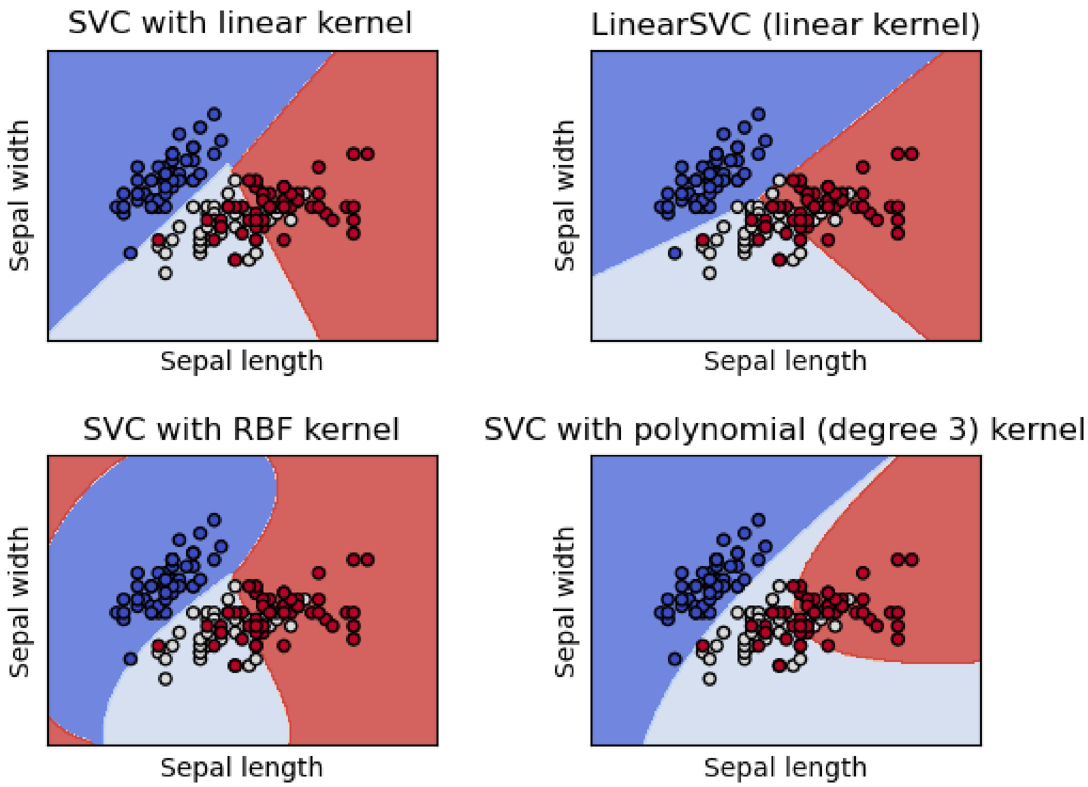


Figure 26. Kernel Differences [26]

C is more like penalty parameter of the error term. Keeping the C value high will allow machine learning to produce results more accurately, but there will be an "overfitting" problem. If the C parameter is kept too low, the learning process will not be successful enough and will produce incorrect results.

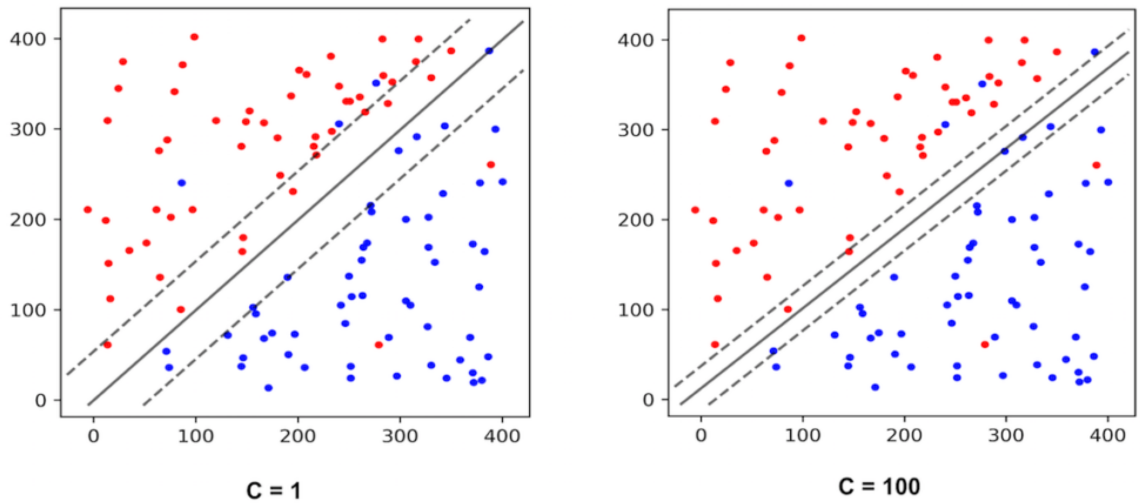


Figure 27. C Parameter Difference [27]

Preferred Python 3.10 and PyCharm as the development environment. Used Support Vector Regression, SVR, one of the estimators of SVM from the Scikit-Learn library.

4.2.1.2 ANN

Making concrete compressive strength estimation with machine learning algorithms is a very complex process. We have 8 Integer data as input; There are Cement, Blast Furnace Slag, Fly Ash, Water, Superplasticizer, Coarse Aggregate, Fine Aggregate, and Age and an output Concrete Strength Integer value will be estimated using this input data. When we look at the problem at hand, it was thought appropriate to establish a neural network to solve our problem, and the Neural network will achieve an effective result for more than 8 Features. ANN is a deep learning application and is often used to solve complex regression or classification problems.

$$y = wx + b$$

Equation 2. Single Neuron Formula

The basic component of the neural network is the neuron. Neural clusters form layers, and the layers combine to form a neural network. A simple neuron receives input and produces an output, its mathematical structure is also shown above. X refers to the value of the feature taken from the outside, and w is the variable that determines the weight of that feature in that equation by calling it weight. At each determined step, the best w value for the X value will be found by searching for the optimization method. b, on the other hand, is an individual type of weight we call bias, and it regulates the equation regardless of the input. A structure with two features and two neurons is simply shown in the figure20.

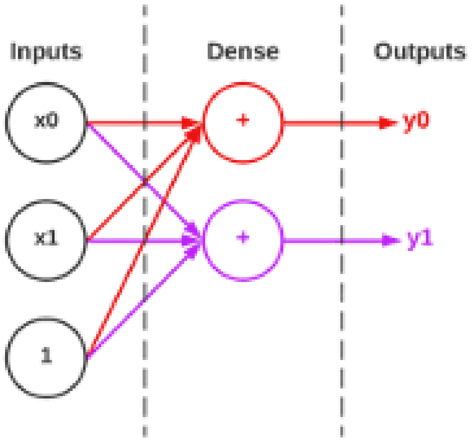


Figure 28. Neurons

In our problem, there are 8 features, features are Cement, Fly Ash, Water, Superplasticizer, Coarse Aggregate, Fine Aggregate and Age. The equation of our neuron structure will be as follows.

$$y = w_0x_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_5x_5 + w_6x_6 + w_7x_7 + b$$

Equation 3. eight-featured neuron formula

Neural networks can only learn linear relationships without activation functions. Activation functions are essential to fit the curves. ReLU (Rectifier to Linear Unit) is the most common activation function and is successful in complex training.

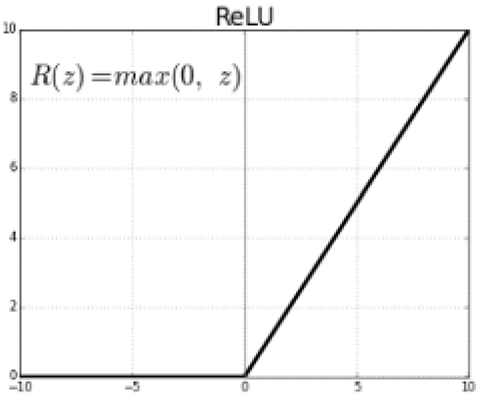


Figure 29. ReLU function

The activation function is applied to the output produced by the neuron. ReLU is a simple function that uses to bend to the simple line by rectifying the negative part of the output to 0. A neuron whose activation function is applied is simply shown in the figure21.

$$\boldsymbol{a} = \mathbf{max}(\mathbf{0}, \boldsymbol{y})$$

Equation 4. ReLU formula

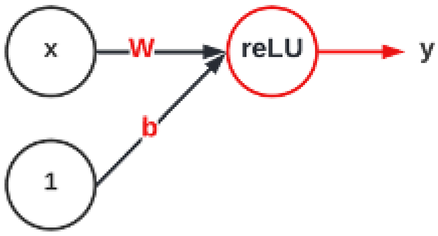


Figure 30. Linear single neuron with ReLU

Neural networks typically organize their neurons into layers. When we collect together linear units having a standard set of inputs, we get a dense layer. The layers before the output layer, whose output we cannot see, are called hidden layers. Also, since we are trying to solve the regression problem, our last layer, which we use to determine our output, does not have an activation process. An example neural network model with two inputs, four neurons, and two hidden layers is shown in the figure 32.

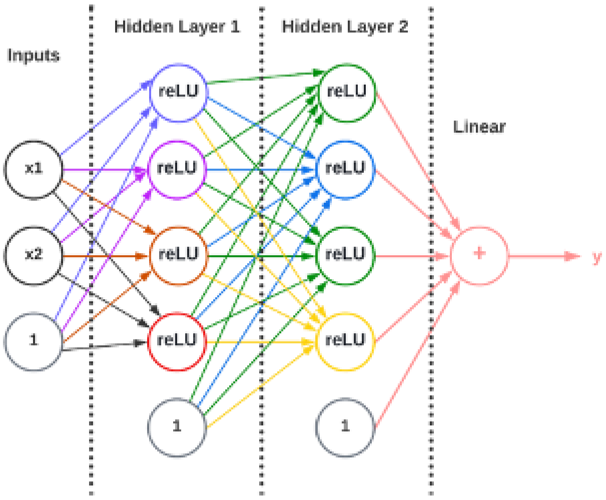


Figure 31. Simple neural network

Our Loss value represents the difference between the output from the neuron output and the Concrete Strength column data we are trying to estimate. The following loss function is used to generate the Loss value in our neural network. Of the function parameters, a represents the output of the activation function, and parameter y represents the actual value to be reached.

$$\mathcal{L}(a, y) = -(y log(a) + (1 - y) \log(1 - a))$$

Equation 5. MSE Loss Function

Selecting the correct model is not enough. You need a function that measures the performance of a Machine Learning model for given data. Cost Function quantifies the error between predicted values and expected values. Then, the loss outputs produced for each feature will be calculated as a cost function via MSE (Mean square error).

$$J(w, b) = \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}(a^i, y^i)$$

Equation 6. Cost Function

When this error function is plotted with respect to weight parameters of the Linear Regression Model, it forms a convex curve which makes it eligible to apply Gradient Descent Optimization Algorithm to minimize the error by finding global minima and adjust weights.

The lowest value is sought in the plotted graph and the low value shows that the performance of our model is good.

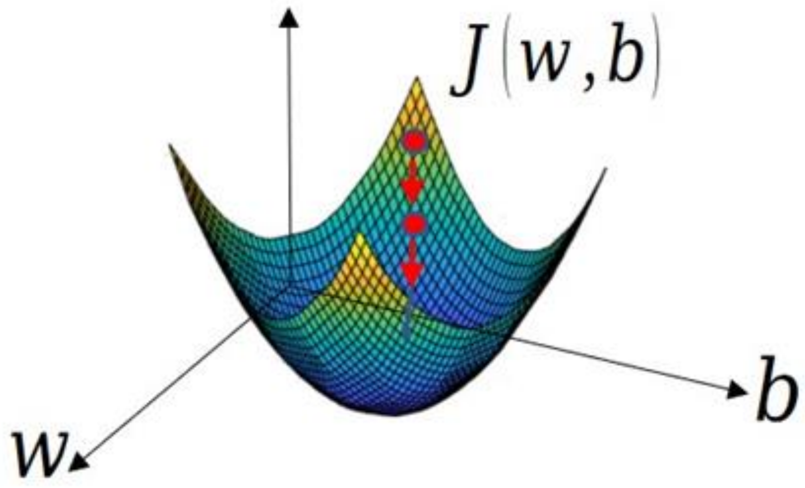


Figure 32. Sample gradient descent[22]

We have indicated what problem the network is seeking to solve. Now we must define how to solve this problem. A complete round of the training data is called an epoch. Each batch operation means using the training data once and bringing the model one step closer to a better result. If we take giant steps in each epoch, we can skip the lowest value, or if we take small steps, we can find ourselves in an education that has no end. For this reason, it is necessary to choose an effective optimization algorithm to solve the problem.

The two most important parameters of gradient training are Learning rate and Batch size. Figuring out how to choose the most accurate of these values is not obvious. These values are determined according to the results obtained. Extensive experiments are required to determine these hyperparameters, so rmsprop or adam algorithms are successful for hyperparameter optimization and adam is a great optimizer for our training.

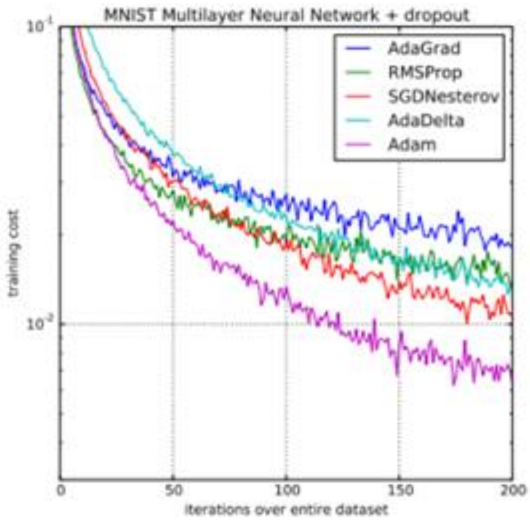


Figure 33. Differences of optimizers[24]

Suppose we are recording epoch outputs from the training process. The outputs of this training process can be graphed, and this process can be interpreted to make training better.

In the training process, data are of two types signal and noise. While the signal data improves the prediction ability of our model, the noise data seems to be random and will train our model, causing our model to be trained incorrectly. The training phase and validation phases of our model have been recorded epoch by epoch, and the loss outputs of the process will be used for graphing. We are in search of the most appropriate weights and parameters to minimize the loss in the training set. And to verify this result, we will evaluate it on the validation data. The output graph will be interpreted to shape the model structure.

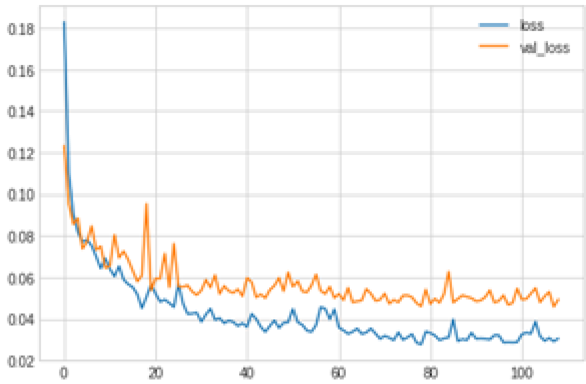


Figure 34. Epoch, Loss graphic

When our model is trained with signal data or noise data, the training loss graph will decrease, but the validation loss graph will fall only when the signal is learned. As a result, both curves will fall as the model learns the signal data, but the difference between the two graphs will increase as the noise is learned. This graph will clearly show us at which stage of the training we learned noise data. The learning of the signal data and the noise data will take place at the same time, but it is intended to continue the training as long as this ratio is tolerable. As a result of training, it can reveal overfitting and underfitting problems.

Overfitting is the case where the noise data in the training set is learned a lot by the model, and the data set is memorized. Underfitting is the situation where the loss value is not low enough as a result of insufficient learning of the signal data in the training set. Being able to balance these two problems is the most important point of education. With the reduction of noise, more focus should be placed on learning the signal.

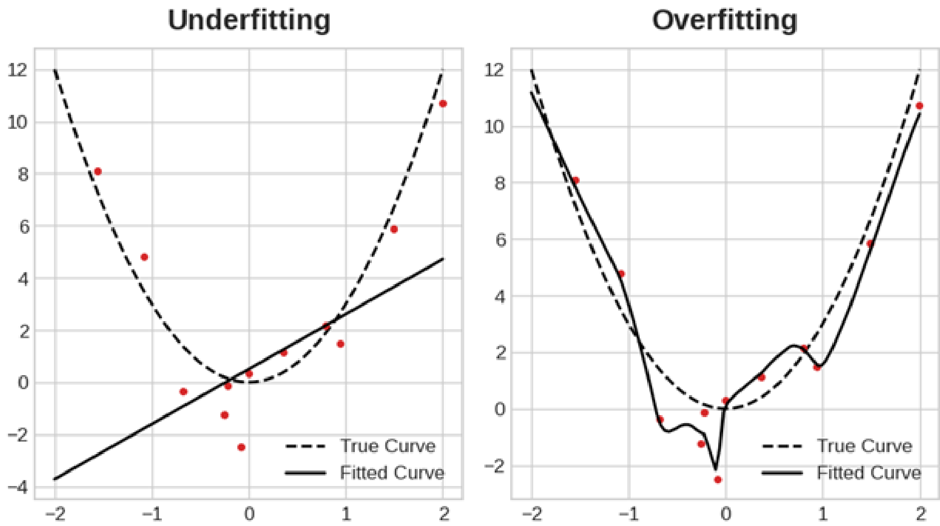


Figure 35. Underfitting and Overfitting representation[23]

The capacity of our model determines the size and complexity of our model. Complexity has to do with how many neurons and how those neurons are connected. In case of underfitting, our capacity may need to be increased. To expand a network, it is necessary to increase the number of layers, and to increase the depth of the network, it is essential to add neurons to the layers. Increasing the depth is the right option for non-linear models, increasing the width is the right option for linear models.

We know that the validation loss will start to increase as the model learns more about noise during training. To avoid this, we need to stop the training when the validation loss transitions to a non-reducing state. We can stop training when the validation loss starts to increase (in the case of its lowest value). This will ensure that the noises are not learned. Early stopping will prevent the training from being interrupted and the necessary signals being learned. For this reason, the training epochs should be a little more than they should be, and with early stopping, both sufficient trainings will be provided, and overfitting will be prevented by early stopping.

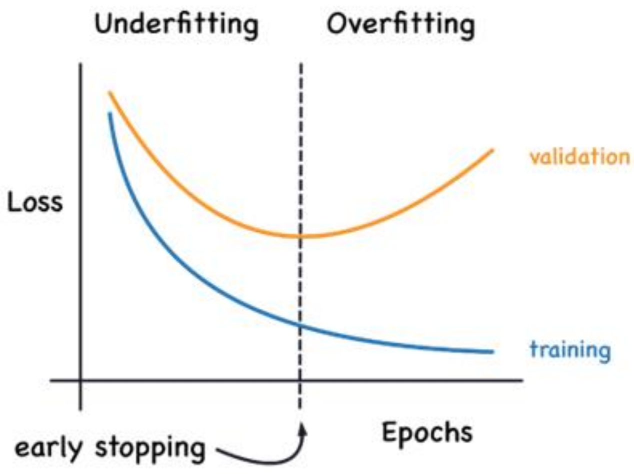


Figure 36. Early Stopping

Before model training, it is necessary to normalize the data set. At the same time, the results with the network weight and the size of the activation can be normalized. Unnormalized data can cause unstable educational behavior. For this, the neural results are normalized to the training group's own mean and standard deviation, and then converted to a new scale with two trainable rescalings. Generally, batch normalization can be helpful for the optimization process, it tends to need fewer epochs to complete the training. At the same time, it can prevent the "stuck" problem of education. If there is a problem during the training, it should definitely be considered. This step is shown in appendix B.

However, we did not use batch normalization and dropout because they positively affect the training of our model.

4.2.1.3 Flaml Library

Our data set consists of numerical values. Random forest method is a suitable method for our target, since our target value is also numerical and not large in size. Random forest uses randomness in both observation and variable selection. Random Forest is one of the most powerful supervised learning algorithms that can perform regression as well as classification tasks. It is an ensemble learning method that combines multiple decision trees and estimates the final output based on the average of each tree output. It can handle thousands of input variables without variable deletion. It gives estimates of which variables are important in classification.

LightGBM is a histogram-based algorithm that works. It reduces the computational cost by making the variables with continuous values discrete. The training time of the decision trees is directly proportional to the calculation and therefore the number of divisions. Thanks to this method, both the training time is shortened and the resource usage is reduced.

XGBoost(eXtreme Gradient Boosting) is a high-performance version of the Gradient Boosting algorithm optimized with various modifications. The most important features of the algorithm are that it can achieve high predictive power, prevent over-learning, manage empty data and do them quickly. Software and hardware optimization techniques have been applied to obtain superior results using less resources. It is cited as the best of the decision tree-based algorithms. Instead of examining each value in the data, XGBoost divides the data into pieces (quantiles) and works according to these pieces. As the amount of parts is increased, the algorithm will look at smaller intervals and make better predictions. Of course, in this case, the learning time of the model also increases. We can show performance as the negative side. However, if this is ignored, quite good results are obtained.

Extra trees are a type of ensemble learning technique that aggregates the results of different interrelated decision trees, similar to a Random forest. It works by creating a large number of unpruned decision trees from the training data set. Estimates are made by averaging the estimation of the decision trees in the case of regression or using majority voting in the case of classification.

FLAML is a Python library that automatically, efficiently and economically finds the right machine learning models. It saves users from choosing models and hyperparameters for each model. It can provide us with the best method over several models. In this project, work was done on 'lgbm', 'rf', 'xgboost', 'extra_tree', 'xgb_limitdepth' models. Here, the best results were obtained with the XGBRegressor method.

4.2.2. Evaluation

4.2.2.1. SVM

Trained SVR model by splitting the data set into 70% training and 30% testing. After training SVR model, it was subjected it to some evaluation criteria. R^2 Score is a frequently used metric that calculates how accurately the estimation was produced.

The Explained Variance Score is very similar to the R^2 Score in general, but the only difference is that it does not take into account the systematic offsets values and produces results in this way. In tests with a low R^2 Score and a high Variance score, it indicates that the data set contains biases.

On the second part which is under the "Result of Reality" is for predicting the real data set and comparing it with itself.

For more information about code, please check Appendix C.

```
R2 Score: 0.8688163136527673
Explained Variance Score: 0.8694801870145954
MAE Score: 0.19236792876407038
MSE Score: 0.07055796021838595
===============RESULT OF REALITY================
R2 Score: 0.13516472868750817
Explained Variance Score: 0.3912031403910833
MAE Score: 0.5019001456327342
MSE Score: 0.32431436992689017
```

Figure 37. SVR Result

The fact that the R^2 score is this high and the difference with Variance is very small indicates that the data has very little deviation and high consistency.

On the contrary, it is understood that the predictions made with real data have a very low accuracy rate, while the Variance score is much higher, so there are many deviations in the real data set.

While doing research in the field of data science, we realized that with the "Hyper Tunning" feature, the most optimal settings can be obtained automatically and high values can be obtained. Check Appendix D for more information about code.

```
SVR(C=10, gamma=0.1)
R2 Score: 0.859510642547601
Explained Variance Score: 0.8599891815470231
MAE Score: 0.20717417623802634
MSE Score: 0.07556307320100006
===============RESULT OF REALITY================
R2 Score: 0.10152001744259509
Explained Variance Score: 0.3989400041437079
MAE Score: 0.5044364123390096
MSE Score: 0.3369311811170798
```

Figure 38. SVR with GridSearch Function

Since it performs random searches for optimal values, the values can be reached in a very short time, it can take a very long time to process, but as a result, we got a higher R score than we expected. However, it is not as accurate as in the scenario where the parameters are entered manually.

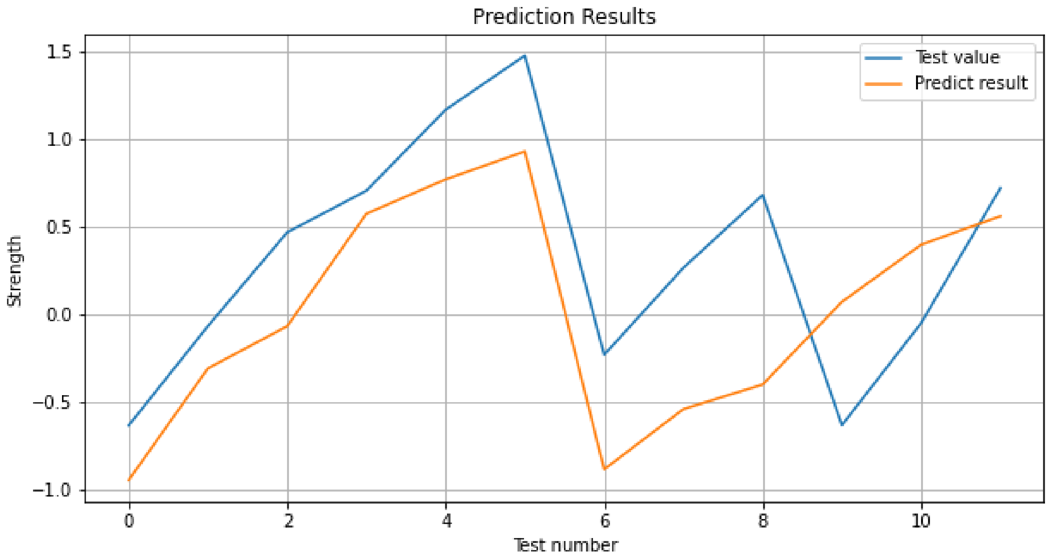


Figure 39. SVR Prediction Result

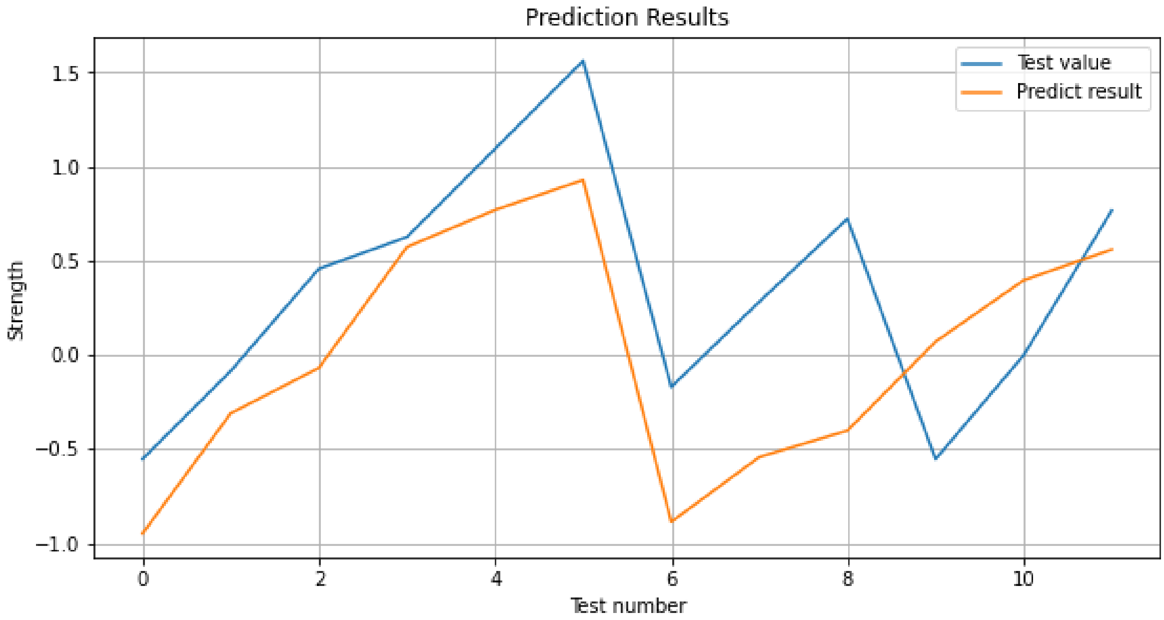


Figure 40. SVR GridSearch Prediction Result

4.2.2.2. ANN

The structure and training parameters of our artificial neural network model to be created were searched through the BayesianOptimization algorithm and the most suitable hyperparameters for this training were found. The model was created with the parameters found and the training stages were recorded and graphed. The accuracy of the training process was evaluated in the graph. The number of repetitions was determined as 150 epochs and with EarlyStop around 65 epoch was stopped at the training stage to prevent overfitting. Modelling codes are shown in appendix B.

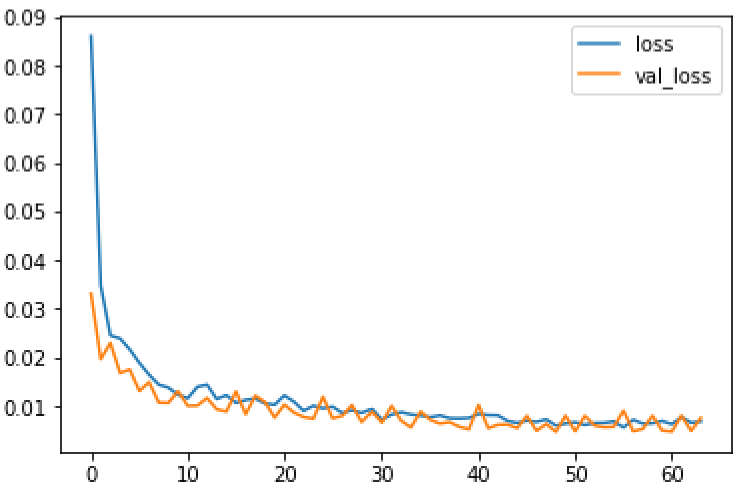


Figure 41. ANN training

The created model was tested with the test data set obtained from civil engineers. The test data set was obtained with the results of the concrete demolition processes created by the teams of civil engineers.

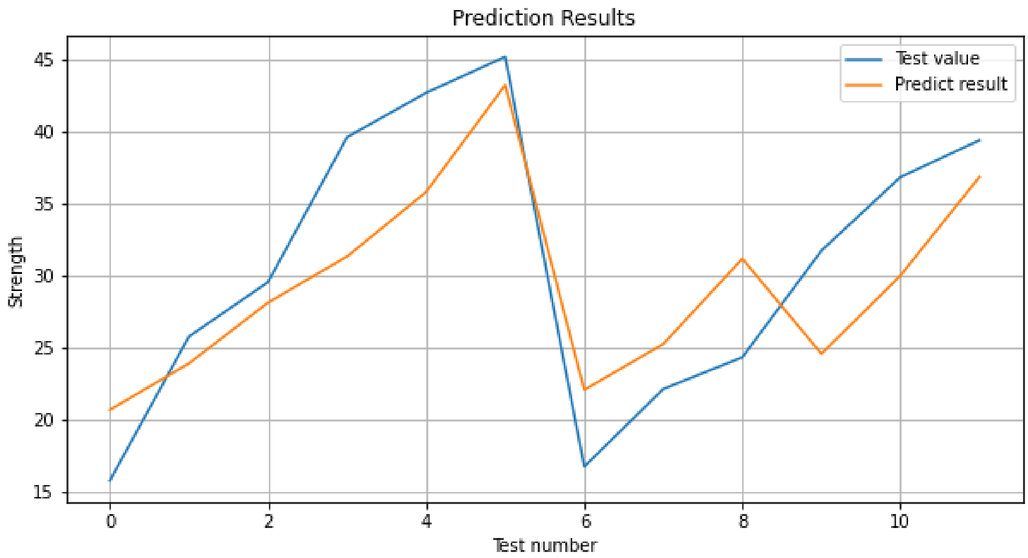


Figure 42. ANN Prediction Results

The test results were graphed and the r2 score was 0.7743. Appendix B are the codes that provide the results.

```
MAE:   3.932589
MSE:   20.738845
r2 score:   0.774345
```

Figure 43. ANN scores

4.2.2.3. Flaml

The best estimation from the model scanning result we made using AutoML with the

Flaml library resulted in XGBRegressor. This part shown in Appendix A. With this library, searched several models with hyperparameter optimization. Given best result is XGBRegressor and results are shown as Figure 44.

```
automl.model.estimator

XGBRegressor(colsample_bylevel=0.9030482524943064,
             colsample_bytree=0.9278972006416252, grow_policy='lossguide',
             learning_rate=0.0068766724195393905, max_depth=0, max_leaves=206,
             min_child_weight=1.9495322566288034, n_estimators=4248, n_jobs=-1,
             reg_alpha=0.01857648400903689, reg_lambda=6.021166480604588,
             subsample=0.9451618245005704, tree_method='hist',
             use_label_encoder=False, verbosity=0)
```

Figure 44. Flaml Library - Best model result

The success rate of the model we trained with XGBRegressor is shown in figure 45.

```
MAE: 2.483616515896845
MSE: 12.346970170210023
r2 Score: 0.9449777810684231
```

Figure 45. XGBRegressor result

The predicted results with the test results are shown in figure 46.

```
MAE: 5.599214909871418
MSE: 43.91536001902251
r2 Score: 0.5221654977553623
```

Figure 46. XGBRegressor result

There may be many reasons for the low predictive results. The scope, variety and low number of experiments may be factors.

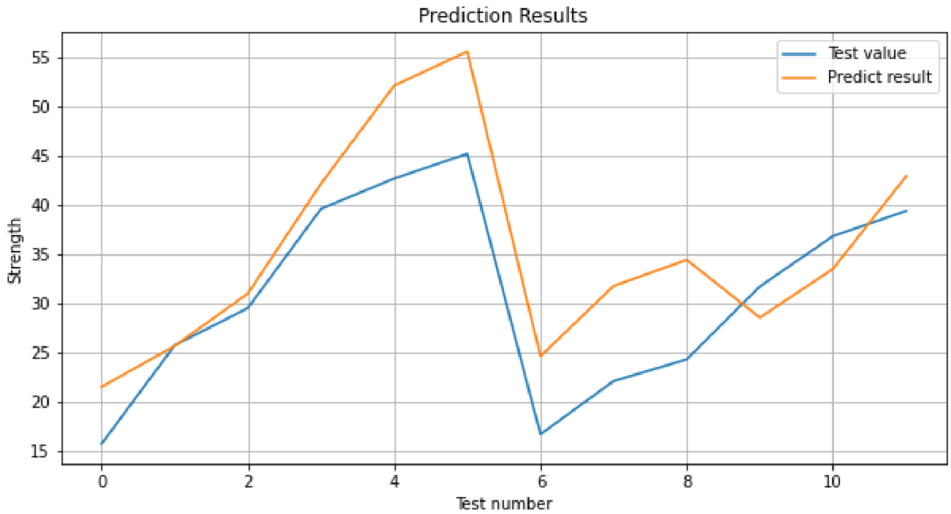


Figure 47. XGBResgressor prediction result

The graphical version of the predicted outcome of the model trained with the data set from the other team is shown in Figure 47. The model trained with this data set was predicted with the results of the experiment.

5. SUMMARY AND CONCLUSION

Our project has been a project that aims civil engineering students and computer engineering students to work with each other and gather and develop in a common project. At the beginning of our project, a literature search was conducted to determine which programs could be used and researched by computer engineering students. As a result of these studies, computer engineering students were given a data set consisting of 366 experiments with different components and these components in different proportions, with sample waiting at different times. After giving the data that could create the program to the computer engineering students, the experimental test stages were started as civil engineer students. First, a project program was drawn up and the missing materials were identified. The missing materials were procured from different companies within the scope of the project budget and collected in the laboratory section. Experiments were started in the laboratory within a week. In the company of our project manager, an adviser, 12 tests were determined with different ratios, different components, and the waiting time of the concrete samples was different, and the construction phases were started. We achieved different concrete strength results in these 12 tests, but it was aimed to determine the most optimized concrete ratio expected from us. These determined rates are as follows. An average concrete in the normal strength class consists of approximately 75%

aggregate, 10% cement and 15% water in absolute volume. When necessary, chemical additives can be added up to approximately 2% of the cement weight. Model training was started with the data set delivered by Civil Engineers. Especially in the optimization part, various parameters, settings, and libraries were tried. While working with SVM, ANN, and Regression (Flaml) models each other respectively, the data set was set to be 70% training and the remaining 30% testing. Took care to ensure that the R2 score passed the 80% accuracy threshold that we targeted.

As a counclusion, this project aims to measure the compressive strength of concrete to be obtained from a concrete mix using Machine Learning algorithms. Since the number of effective parameters in concrete is high, both time and material are lost. The project aims to minimize or even eliminate these losses. We desired to obtain the most suitable concrete under the right conditions with the estimation model. In line with our goal, dependent and independent variables will be used together. We will comment on the data affected according to a set of inputs. Based on this, we conducted research on our scope of supervised learning. According to the literature research we have done, we have concentrated on classification and regression models in the project target. We mainly examined ANN, SVM, and Regression models. Our goal was to achieve 80% success. When we analyze the data set we received, our data consists of purely numerical and continuous data. Therefore, we focused only on regression models. We have worked with ANN, SVM, and other regression models. According to the results we obtained, ANN gave the most successful effect with a rate of 0.77. Our target rate was 80% and an approximate result was obtained.

ACKNOWLEDGEMENTS

We wish to thank our adviser Assist. Prof. Görkem Kar and Assist. Prof İrem Şanal for their support and quick feedback.

This work was wholly funded by Bahçeşehir University.

REFERENCES

1. J.L. Acevedo-Dávila, L. M. Torres-Treviño, *Compression Strength Prediction of Mixtures Concrete with Scrap Tire with Neural Network Approach.* Mexica. Seventh Mexican International Conference, 2008
2. S. Saad, M. Ishtiyaque and H. Malik, "*Selection of most relevant input parameters using WEKA for artificial neural network based concrete compressive strength prediction model*," 2016 IEEE 7th Power India International Conference (PIICON), 2016, pp. 1-6, doi: 10.1109/POWERI.2016.8077368.
3. Xiaoyun Zhang, H. Wang, Delin Wang and Chendong Li, "*Prediction of concrete strength based on self-organizing fuzzy neural network*," Proceeding of the 11th World Congress on Intelligent Control and Automation, 2014, pp. 5631-5634, doi: 10.1109/WCICA.2014.7053679.
4. J. Zhang and Y. Zhao, "*Prediction of Compressive Strength of Ultra-High Performance Concrete (UHPC) Containing Supplementary Cementitious Materials,*" 2017 International Conference on Smart Grid and Electrical Automation (ICSGEA), 2017, pp. 522-525, doi: 10.1109/ICSGEA.2017.150.
5. P. Singh and P. Khaskil, "*Prediction of Compressive Strength of Green Concrete with Admixtures Using Neural Networks*," 2020 IEEE International Conference on Computing, Power and Communication Technologies (GUCON), 2020, pp. 714-717, doi: 10.1109/GUCON48875.2020.9231230.
6. H. Erdal, "*Contribution of Machine Learning Methods to the Construction Industry: Prediction of Compressive Strength*", Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi, c. 21, sayı. 3, ss. 109-114, Tem. 2015, doi:10.5505/pajes.2014.26121
7. J. A. Abdalla, R. Hawileh and A. Al-Tamimi, "*Prediction of FRP-concrete ultimate bond strength using Artificial Neural Network*," 2011 Fourth International Conference on Modeling, Simulation and Applied Optimization, 2011, pp. 1-4, doi: 10.1109/ICMSAO.2011.5775518.
8. Guan Ping and Liu Peng, "*Study of strength criterion for dynamic biaxial compressive properties of concrete under constant confining pressure*," 2011 International Conference on Electric Technology and Civil Engineering (ICETCE), 2011, pp. 753-755, doi: 10.1109/ICETCE.2011.5775828.
9. R. Wang, Y. Dai, C. Han, K. Xu and L. Dong, "*Application of DPO—BP in strength prediction of concrete*," 2017 IEEE 3rd Information Technology and Mechatronics

Engineering Conference (ITOEC), 2017, pp. 1003-1006, doi: 10.1109/ITOEC.2017.8122505.

10. M. KAYA KELEŞ, A. E. KELEŞ and Ü. KILIÇ, "*PREDICTION OF CONCRETE STRENGTH WITH DATA MINING METHODS USING ARTIFICIAL BEE COLONY AS FEATURE SELECTOR*," 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), 2018, pp. 1-4, doi: 10.1109/IDAP.2018.8620905.
11. F. H. Chiew, "*Prediction of Blast Furnace Slag Concrete Compressive Strength Using Artificial Neural Networks and Multiple Regression Analysis*," 2019 International Conference on Computer and Drone Applications (IConDA), 2019, pp. 54-58, doi: 10.1109/IConDA47345.2019.9034920.
12. P. Chopra , R. K. Sharma ve M. Kumar, "*PREDICTING COMPRESSIVE STRENGTH OF CONCRETE FOR VARYING WORKABILITY USING REGRESSION MODELS*", International Journal of Engineering and Applied Sciences, c. 6, sayı. 4, ss. 10-22, Ara. 2014, doi:10.24107/ijeas.251233
13. Hongwei Song, Ayaz Ahmad, Furqan Farooq, Krzysztof Adam Ostrowski, Mariusz Maślak, Slawomir Czarnecki, Fahid Aslam, " *Predicting the compressive strength of concrete with fly ash admixture using machine learning algorithms* ", Construction and Building Materials, vol. 308, 2021.
14. De-Cheng Feng, Zhen-Tao Liu, Xiao-Dan Wang, Yin Chen, Jia-Qi Chang, Dong-Fang Wei, Zhong-Ming Jiang, " *Machine learning-based compressive strength prediction for concrete: An adaptive boosting approach* ", Construction and Building Materials, vol. 230, 2020.
15. Robert Flatt , " *Shape and size effects on the compressive strength of high-strength concrete* ", Cement and Concrete Research, vol.38, March 2008.
16. Selahattin ALBAYRAK *1, Ahmet ÇAVDAR1, Şinasi BİNGÖL2 1Gümüşhane Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, İnşaat Müh. Böl., 29000, Gümüşlhane/Türkiye.
17. Ö. S. Keskin, A. Durdu, M. F. Aslan and A. Yusefi, "Performance comparison of Extreme Learning Machines and other machine learning methods on WBCD data set," *2021 29th Signal Processing and Communications Applications Conference (SIU)*, 2021, pp. 1-4, doi: 10.1109/SIU53274.2021.9477984.
18. Y. Okada, S. Ata, N. Nakamura, Y. Nakahira and I. Oka, "Comparisons of Machine Learning Algorithms for Application Identification of Encrypted Traffic," *2011 10th International Conference on Machine Learning and Applications and Workshops*, 2011,

pp. 358-361, doi: 10.1109/ICMLA.2011.162.

19. M. Sewak, S. K. Sahay and H. Rathore, "Comparison of Deep Learning and the Classical Machine Learning Algorithm for the Malware Detection," *2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, 2018, pp. 293-296, doi: 10.1109/SNPD.2018.8441123.
20. E. A. Bayrak, P. Kırcı and T. Ensari, "*Comparison of Machine Learning Methods for Breast Cancer Diagnosis*," 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT), 2019, pp. 1-3, doi: 10.1109/EBBT.2019.8741990.
21. H N Muliauwan, D Prayogo, G Gaby, K Harsono, "*Prediction of Concrete Compressive Strength Using Artificial Intelligence Methods*" 2020 Journal of Physics Conference Series, doi: 10.1088/1742-6596/1625/1/012018.
22. *coursera.* (2022). www.coursera.org: https://www.coursera.org/specializations/deep-learning/
23. *Kaggle.* (2022). www.kaggle.com: https://www.kaggle.com/code/ryanholbrook/overfitting-and-underfitting/
24. *machinelearningmastery.* (2022). machinelearningmastery.com: https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/
25. A Mature-Tomato Detection Algorithm Using Machine Learning and Color Analysis - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/A-linear-Support-Vector-Machine-SVM-case_fig2_332777280 [accessed 29 May, 2022]
26. *scikit-learn.* (2022). scikit-learn.org: https://scikit-learn.org/stable/auto_examples/svm/plot_iris_svc.html
27. *medium.com.* (2022). medium: https://medium.com/deep-learning-turkiye/nedir-bu-destek-vekt%C3%B6r-makineleri-makine-%C3%B6%C4%9Frenmesi-serisi-2-94e576e4223e

APPENDIX A

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb
from scipy import stats
from sklearn import preprocessing
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
from xgboost import XGBRegressor


#import data set
df = pd.read_csv('/content/drive/MyDrive/Concrete_Data.csv',delimiter=";")
df = df[df['Blast Furnace Slag ']==0.0]
df.drop(['Blast Furnace Slag '],axis=1,inplace=True)
df.head()


for i in range(len(df.columns)):
  df.columns[i].strip()


df.info()


df.isnull().sum()


#data set distribution
df.describe().T
```

```
df.drop_duplicates(inplace=True)

#column graphs
fig, ax2 = plt.subplots(4, 2, figsize=(8, 16))
sb.distplot(df['Cement '],ax=ax2[0][0])
sb.distplot(df['FlyAsh '],ax=ax2[0][1])
sb.distplot(df['Water  '],ax=ax2[1][0])
sb.distplot(df['Superplasticizer'],ax=ax2[1][1])
sb.distplot(df['CoarseAggregate  '],ax=ax2[2][0])
sb.distplot(df['FineAggregate '],ax=ax2[2][1])
sb.distplot(df['Age '],ax=ax2[3][0])
sb.distplot(df['Strength'],ax=ax2[3][1])

components = [col for col in df.columns.tolist() if col not in ['Strength']]

def analysis_components(df):
  for col in components:
        print("*"*50)
        print("Column name: ", col)

        fig, (ax1,ax2,ax3)=plt.subplots(1,3,figsize=(13,5))

        #boxplot
        sb.boxplot(x=col,data=df,orient='v',ax=ax1)
        ax1.set_ylabel(col, fontsize=15)
        ax1.set_title(f'Distribution of {col}', fontsize=15)
        ax1.tick_params(labelsize=15)

        #distplot
        sb.distplot(df[col],ax=ax2)
        ax2.set_xlabel(col, fontsize=15)
        ax2.set_ylabel(col, fontsize=15)
        ax2.set_title(f'{col} vs Strength', fontsize=15)
        ax2.tick_params(labelsize=15)

        #histogram
        ax3.hist(df[col])
        ax3.set_xlabel(col, fontsize=15)
        ax3.set_ylabel(col, fontsize=15)
```

```
        ax3.set_title(f'{col} vs Strength', fontsize=15)
        ax3.tick_params(labelsize=15)

        plt.subplots_adjust(wspace=0.5)
        plt.tight_layout()
        plt.show()
        print("#"*50)
analysis_components(df)

#correlation graph
df_corr=df.corr()

plt.figure(figsize=(14,10))
sb.heatmap(df_corr, annot=True, cmap='Blues')
plt.show()

#Outlier detection
data_model=df.copy()
data_model.boxplot(figsize=(35,15))

#Quantile method and fixing with median value
for col_name in data_model.columns[:-1]:
    q1 = data_model[col_name].quantile(0.25)
    q3 = data_model[col_name].quantile(0.75)
    iqr = q3 - q1

    low = q1-1.5*iqr
    high = q3+1.5*iqr
    data_model.loc[(data_model[col_name] < low) | (data_model[col_name] > 
high), col_name] = data_model[col_name].median()


data_model.boxplot(figsize=(35,15))

X = df.drop("Strength", axis=1)
y                              =                              df["Strength"]

#Feature Engineering / Train-Test Split
```

```
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.3,
random_state = 123)

#Data Normalization with Robust Scaler
rbs = preprocessing.RobustScaler()
scaled_Xtrain = rbs.fit_transform(X_train)
scaled_XTest = rbs.transform(X_test)

#Models

#Evaluate method for error values and r2 value
def evaluate(y, predictions):
    mae = mean_absolute_error(y, predictions)
    mse = mean_squared_error(y, predictions)
    r_squared = r2_score(y, predictions)
    return mae, mse, r_squared

#dataframe for model results
models = pd.DataFrame(columns=["Model", "MAE", "MSE", "r2 Score"])

#Flaml lib
!pip install flaml
from flaml import AutoML

automl = AutoML()
automl_settings = {"task":"regression"}
automl.fit(scaled_Xtrain, y_train, **automl_settings)

automl.model.estimator

predictions = automl.predict(scaled_XTest)
mae, mse, r2 = evaluate(y_test, predictions)
print("MAE:", mae)
print("MSE:", mse)
print("r2                              Score:",                            r2)

#Experimetal Results
df_exp=pd.read_csv('/content/drive/MyDrive/CapstoneDeney.csv',delimiter=";"
```

```
)



    for i in range(len(df_exp.columns)):
      df_exp.columns[i].strip()

    X = df_exp.drop("Strength", axis=1)
    y = df_exp["Strength"]

    rbs = preprocessing.RobustScaler() #robust
    scaled_X = rbs.fit_transform(X)

    automl.model.estimator

    predictions = automl.predict(scaled_X)
    mae, mse, r2 = evaluate(y, predictions)
    print("MAE:", mae)
    print("MSE:", mse)
    print("r2 Score:", r2)



    #graph
    fig = plt.figure(figsize=(10,5))
    plt.plot(y, label='Test value')
    plt.plot(predictions, label='Predict result')
    plt.title('Prediction Results')
    plt.xlabel('Test number')
    plt.ylabel('Strength')
    plt.grid()
    plt.legend()
```

APPENDIX B

```
def model_builder(hp):
  hp_units = hp.Int('units', min_value=256, max_value=1024, step=32)

  model = keras.Sequential()
  #model.add(keras.layers.Dense(input_shape=[7]))

  # Tune the number of units in the first Dense layer
  model.add(keras.layers.Dense(units=hp_units, activation='relu', input_sha
pe=[7]))
  model.add(layers.Dropout(0.3))
  model.add(layers.BatchNormalization())
  model.add(keras.layers.Dense(units=hp_units, activation='relu'))
  model.add(layers.Dropout(0.3))
  model.add(layers.BatchNormalization())
  model.add(keras.layers.Dense(units=hp_units, activation='relu'))
  model.add(layers.Dropout(0.3))
  model.add(layers.BatchNormalization())
  model.add(layers.Dense(units=1))

  # Tune the learning rate for the optimizer
  # Choose an optimal value from 0.01, 0.001, or 0.0001
  hp_learning_rate = hp.Choice('learning_rate', values=[1e-2, 1e-3, 1e-4])

  model.compile(optimizer=keras.optimizers.Adam(learning_rate=hp_learning_r
ate),
                loss='mse',
                metrics=['mse','mae'])
  return model

stop_early = EarlyStopping(
    min_delta=0.001,
    patience=30,
    restore_best_weights=True,
)

tuner = kt.BayesianOptimization(model_builder,
                                objective='mse',
```

```
                                max_trials=20,
                                seed=42,
                                executions_per_trial=2)

tuner.search(X_train_scaled, y_train, epochs=50, validation_data=(X_valid_s
caled, y_valid), verbose=1, callbacks=[stop_early])

# Get the optimal hyperparameters
best_model = tuner.get_best_models(num_models=1)[0]
y_predicted = manuel_model.predict(X_test_scaled)
r2_score(y_test, y_predicted)
fig = plt.figure(figsize=(10,5))
plt.plot(y_test, label='Test value')
plt.plot(y_predicted, label='Predict result')
plt.title('Prediction Results')
plt.xlabel('Test number')
plt.ylabel('Strength')
plt.grid()
plt.legend()
```

APPENDIX C

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import RobustScaler
from sklearn import metrics
from sklearn.svm import SVR
from sklearn.model_selection import train_test_split


#dataset = pd.read_excel('D:/data.xlsx')
dataset = pd.read_excel('D:/data_long.xlsx')
real = pd.read_excel('D:/finalxlsx.xlsx')

X = dataset.iloc[:, 0:7].values
y = dataset.iloc[:, 7].values

sc_X = RobustScaler()
sc_y = RobustScaler()
X = sc_X.fit_transform(X)
y = sc_y.fit_transform(y.reshape(-1,1))


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state = 42)

regressor = SVR(kernel = 'rbf', C= 9)
regressor.fit(X_train, y_train)

y_pred = []
for x_Test in X_test:
    temp = regressor.predict(np.array([x_Test]))
    y_pred.append(temp)


print("R2 Score:",metrics.r2_score(y_test, y_pred))
```

```
print("Explained  Variance  Score:",metrics.explained_variance_score(y_test, y_pred))

print("MAE Score:",metrics.mean_absolute_error(y_test, y_pred))

print("MSE Score:",metrics.mean_squared_error(y_test, y_pred))

print("===============RESULT OF REALITY================")

X_real = real.iloc[:, 0:7].values #pretend it as X_test
y_real = real.iloc[:, 7].values  #pretend it as y_test

sc_real_X = RobustScaler()
sc_real_y = RobustScaler()
X_transform = sc_real_X.fit_transform(X_real)
y_transform = sc_real_y.fit_transform(y_real.reshape(-1,1))


y_real_pred = []

y_real_pred = regressor.predict(X_transform)

print("R2 Score:",metrics.r2_score(y_transform, y_real_pred))

print("Explained Variance Score:",metrics.explained_variance_score(y_transform, y_real_pred))

print("MAE Score:",metrics.mean_absolute_error(y_transform, y_real_pred))

print("MSE Score:",metrics.mean_squared_error(y_transform, y_real_pred))
```

APPENDIX D

```
import numpy as np
import pandas as pd
from sklearn.preprocessing import RobustScaler
from sklearn import metrics
from sklearn.svm import SVR
from sklearn.model_selection import train_test_split
from sklearn.model_selection import GridSearchCV

#dataset = pd.read_excel('D:/data.xlsx')
dataset = pd.read_excel('D:/data_long.xlsx')
real = pd.read_excel('D:/finalxlsx.xlsx')

X = dataset.iloc[:, 0:7].values
y = dataset.iloc[:, 7].values

sc_X = RobustScaler()
sc_y = RobustScaler()
X = sc_X.fit_transform(X)
y = sc_y.fit_transform(y.reshape(-1,1))


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state = 42)


param_grid = {'C': [0.1, 1, 9,10, 100, 1000],
              'gamma': [1, 0.1, 0.01, 0.001, 0.0001],
              'kernel': ['rbf']}

grid = GridSearchCV(SVR(), param_grid, refit = True, verbose = 3)
grid.fit(X_train, y_train)


grid_predictions = grid.predict(X_test)

print(grid.best_estimator_)
```

```
	print("R2 Score:",metrics.r2_score(y_test, grid_predictions))

	print("Explained Variance Score:",metrics.explained_variance_score(y_test, 
grid_predictions))

	print("MAE Score:",metrics.mean_absolute_error(y_test, grid_predictions))

	print("MSE Score:",metrics.mean_squared_error(y_test, grid_predictions))

	X_real = real.iloc[:, 0:7].values #pretend it as X_test
	y_real = real.iloc[:, 7].values  #pretend it as y_test

	sc_real_X = RobustScaler()
	sc_real_y = RobustScaler()
	X_transform = sc_real_X.fit_transform(X_real)
	y_transform = sc_real_y.fit_transform(y_real.reshape(-1,1))


	y_real_pred = []

	y_real_pred = grid.predict(X_transform)

	print("===============RESULT OF REALITY================")

	print("R2 Score:",metrics.r2_score(y_transform, y_real_pred))

	print("Explained Variance 
Score:",metrics.explained_variance_score(y_transform, y_real_pred))

	print("MAE Score:",metrics.mean_absolute_error(y_transform, y_real_pred))

	print("MSE Score:",metrics.mean_squared_error(y_transform, y_real_pred))
```