**Abstract**

This project examines the application of machine learning algorithms to predict whether a student will be successful or not. The specific focus of the project is the comparison of machine learning methods in terms of how much they improve the prediction performance.

Two different machine learning methods were used in this thesis. They are Linear Regression and Support Vector Regression. Feature engineering, the process of modification and selection of the features of a data set, was used to improve predictions made by these learning algorithms.

We used a data set containing information of students of diverse cultures and ethnicities. The machine learning algorithms were applied to the data after pre-processing to predict the student's success.

Through this project, we have presented a machine learning model to predict the performance of students in an academic organization. The algorithm employed is a machine learning technique called Linear Regression.

In this era of computerization, education is not only limited to old lecture method. The regular question is on to find out new ways to make it more effective and efficient for students. Nowadays, lots of data is collected in educational databases, but it remains unutilized. In order to get required benefits from such a big data, the proper analysis is required. Machine Learning is an emerging powerful tool for analysis and prediction. It is a field of computer science that gives computer systems the ability to learn with data, without being explicitly programmed.

**Contents**

1. Introduction…………………………………………………….1
2. Purpose…………………………………………………...…….3
3. Methods

3.1. Machine Learning basics…………………………..…….4

3.1.1. Definition

3.1.2. Predictive analysis

3.2. Selected methods………………..………………...……..6

3.2.1. Linear regression

3.2.2. Support Vector Regression

3.3. Feature engineering……………………………………..12

3.4. Evaluation methods…………..…………………………14

1. System Design…...………………………………...….……..16
   1. List of Diagrams

4.1.1 Use Case diagram

4.1.2 Collaboration diagram

4.1.3 Activity diagram

1. About Dataset……………………………………...….……..19
2. Implementation and results…………………….………........21

* Loading Libraries
* Importing the dataset
* Splitting the dataset into the Training set and Test set
* Training various models

1. Evaluation………………………...…………………………31
2. Screenshots of Project……………………………………….33
3. Conclusions….……………….…...………….……………...VI
4. Future Scope…………………...………………………...…VII
5. References………………………………………………....VIII

**List of Figures**

4.1.1 Use Case Diagram…………………………….……………………..16

4.1.2 Collaboration Diagram………………………………………………17

4.1.3Activity Diagram…………………………………………………..…18

9.1 Code Snippet 1…………………………………………………………33

9.2 Code Snippet 2…………………………………………………………34

9.2 Code Snippet 3…………………………………………………………35

**Chapter 1: Introduction**

With the wide usage of internet, there has recently been a huge increase in publicly available data that can be analyzed. The opportunity is that this type of data is ideal for computers to process, because it is stored digitally in a well-formatted way, and computers can process data much faster than humans can.

Computers can analyze digital data to find patterns that is too complex for a human to do. The basic idea of machine learning is that a computer can automatically learn from experience. Applications of machine learning cover a wide range of areas such as virtual personal assistants, predictions while commuting, videos surveillance, social media services, email spam and malware Filtering.

Depending on the type of input data, machine learning algorithms can be divided into supervised and unsupervised learning. This project focuses on supervised learning, more specifically predictive analytics, which is the process of using machine learning to predict accuracy. Predictive analytics has a wide range of applications, such as fraud detection, analyzing population trends, or understanding user behavior.

The specific focus of this project is education. The aim is to predict student performance. Data about students is used to create a model that can predict the grade of the student, based on other properties. First, the training data set is taken as input. The data set is containing different types of information. The data set is in tabular format. Each row represents a student and each column, or variable, contains certain information or features about a student, such as gender, nationality, semester or birth place.

In addition to this, a column representing the grade of the student is used as the variable that the algorithm is trying to predict. The algorithm creates a model, which is a function that outputs the accuracy in of the student, using other variables as input.

This project evaluates the effectiveness of different machine learning algorithms and methods. There are many algorithms that are used in creating predictive models, we focused on two of them, which are linear regression and support vector regression. The project also measures the improvement made by feature engineering, which refers to modifying the data to make it more suitable for machine learning.

**Chapter 2: Purpose**

Student retention is an important issue in education. Intervention programs can improve retention rates but they need prior knowledge of the students’ performance. This is where performance prediction becomes important. The use of machine learning to predict student performance is commonly done in the field of academia. In virtual learning, dropout prediction is a common focus due to both high dropout rates and data that is available easily. Areas outside of virtual learning are also common contexts where dropout or performance predictions are used for research. The purpose of the research of these studies varies. In some of the cases, the aim is to find the best method for prediction. In other cases, the aim is simply to evaluate whether machine learning is a viable approach for predicting student dropout or performance.

**Chapter 3: Methods**

**3.1. Machine learning basics**

**3.1.1. Definition**

“A computer program is said to learn from experience *E*with respect to some class of tasks *T*and performance measure *P*if its performance at tasks in *T*, as measured by *P*, improves with experience *E*.” -(Mitchell, 1997):

Basically, machine learning is the ability of a computer to learn from experience. Experience is usually given in the form of input data. Looking at this data, the computer can find dependencies in the data that are too complex for a human to form. Machine learning can be used to reveal a hidden class structure in an unstructured data, or it can be used to find dependencies in a structured data to make predictions. Latter is the main focus of the thesis.

**3.1.2. Predictive analytics**

Predictive analytics is the act of predicting future events and behaviors present in previously unseen data, using a model built from similar past data. It has a wide range of applications in different fields, such as finance, education, healthcare, and law. The method of application in all these fields is similar. Using previously collected data, a machine learning algorithm finds the relations between different properties of the data. The resulting model is able to predict one of the properties of future data based on properties.

Table 1 shows example data about students who passed or failed at an exam, along with other information about students.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Age | Gender | GPA | Absences | Passed |
|  |  |  |  |  |
| 14 | F | 3.2 | 5 | 1 |
|  |  |  |  |  |
| 13 | M | 2.4 | 7 | 0 |
|  |  |  |  |  |
| 15 | M | 3 | 6 | 1 |
|  |  |  |  |  |

Table 1. Example data.

The aim is to predict if the student has passed the exam or not by looking at the other variables (the column of the table). In this case, the column “Passed” is called the dependent variable, and every other variable is called the independent variable. In the “Passed” column, “1” means student has passed the exam and “0” means failure in the exam. By applying a machine learning algorithm to this data, a function can be created, also known as the prediction model, that gives the value for the dependent variable as output, and takes every other variable as input.

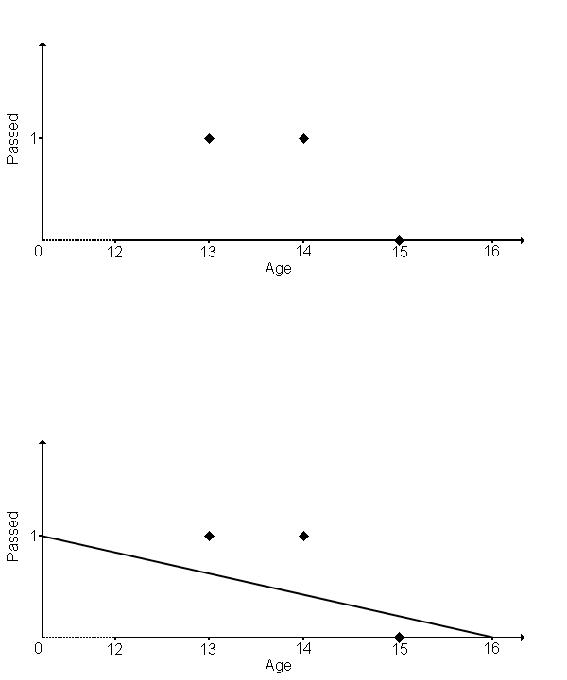
The act of creating a prediction model from previously known data is called training, and such data is called the training data or a training set. After the model is created, it must be applied to another data set to test its effectiveness. Data used for such purpose is called test data or test set. The reason for using two different sets is to ensure that the model is flexible enough to be used on data sets other than the one it was built with. Otherwise, the problem of overfitting may occur, which is when a model is accurate with its original data set, but performs poorly on other data sets, because it is overly complicated. A common method to avoid overfitting is to divide the input data set into training and test sets. To evaluate the model with test data, the model is used to predict the dependent variable in the test set. Then, the predicted values and actual values of the dependent variable are compared. Evaluation is more complicated than looking at the number of correct predictions. There are multiple different evaluation criteria.

**3.2. Selected methods**

There are numerous algorithms to create a prediction model. we used two algorithms: Linear Regression and Support Vector Regression. While they all essentially have the same task, which is predicting a dependent variable based on independent variables, they are based on different mathematical methods.

**3.2.1. Linear regression**

Regression method takes a finite set relations between dependent variable and independent variables, and creates a continuous function generalizing these relations. Table 2 shows another data set containing information about students.



|  |  |
| --- | --- |
| Age | Passed |
|  |  |
| 15 | 0 |
|  |  |
| 14 | 1 |
|  |  |
| 13 | 1 |
|  |  |

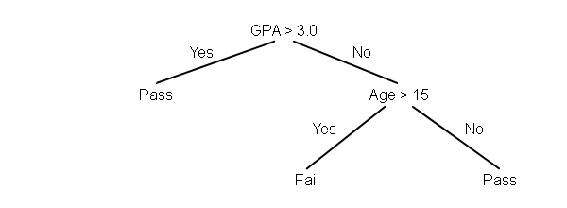
Table 2. Student data.

For the sake of simplicity, the data has only one independent variable. Figure 1 depicts a two-dimensional graph that shows the relation between the student age and the dependent variable indicating whether they have passed the exam or not.

Figure 1. Graph representation of data.

Depending on the type of regression method, regression creates a straight line or a curve that fits the best to the data. Figure 2 shows the graph after the regression.

Figure 2. Graph after the regression.



After the regression model has been constructed, predictions about previously unknown cases, such as age 12 and age 16, can be made. Two things should be noted. The first is that regression does not have to cover the exact points in the previous dotted graph. For example, the function no longer has the same values for ages 13, 14 and 15. This is acceptable, because regression algorithm makes an approximation (Watt *et al*., 2016). Another thing to note is that the function can have any value between 0 and 1 as output. Since 0 and 1 are the only acceptable values, a threshold is needed to convert any output of the function to 0 or 1. For example, threshold can be 0.5, and if the passed value is equal or greater than 0.5, it is 1, otherwise it is 0. Using such threshold, output values for ages 13 and 14 are 1, and 0 for age 15.

**3.2.2. Support Vector Regression**

Support Vector Machine can also be used as a regression method, maintaining all the main features that characterize the algorithm (maximal margin). The Support Vector Regression (SVR) uses the same principles as the SVM for classification, with only a few minor differences. First of all, because output is a real number it becomes very difficult to predict the information at hand, which has infinite possibilities. In the case of regression, a margin of tolerance (epsilon) is set in approximation to the SVM which would have already requested from the problem. But besides this fact, there is also a more complicated reason, the algorithm is more complicated therefore to be taken in consideration. However, the main idea is always the same: to minimize error, individualizing the hyperplane which maximizes the margin, keeping in mind that part of the error is tolerated.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| http://www.saedsayad.com/images/SVR_1.png |  |  |
|  |  |  |
| http://www.saedsayad.com/images/SVR_2.png  Linear SVR |  |  |
| http://www.saedsayad.com/images/SVR_4.png |  |  |
|  |  |  |
| Non-linear SVR |  |  |
| The kernel functions transform the data into a higher dimensional feature space to make it possible to perform the linear separation. |  |  |
| http://www.saedsayad.com/images/SVR_6.png |  |  |
| http://www.saedsayad.com/images/SVR_5.png |  |  |
| Kernel functions |  |  |
| http://www.saedsayad.com/images/SVM_kernel_1.png |  |  |

**3.3. Feature engineering**

In machine learning, feature engineering is the process of selecting or creating features (variables) in a data set to improve machine learning results. Feature selection can include removing unnecessary or redundant features. The process of removing unnecessary variables requires assessing the relevance of the variable. This can be done by creating a model to test the correlation of the variable with the dependent variable. Feature creation includes modifying the variables and creating new ones by combining multiple different variables.

The first use of feature engineering in the thesis is the selection of the relevant variables. Input data may contain too many variables, some of which do not improve the prediction performance, and thus make the predictive model overly complicated. In such a case, unnecessary variables must be removed to make the model more efficient. Deciding which variable to remove can be done manually using domain knowledge or it can be done automatically. In the case of this project, feature selection was done by observing the output of the linear regression model to find how much correlation each variable has with the dependent variable.

The second use of feature engineering in the project is the modification of variables. This can refer to combining multiple variables to create a new variable, calculating a variable differently so that it can be used better in classification, or categorizing a variable so that it has a limited range of possible values. An example of variable modification can be made with a student data set containing the native language of the student as one of the variables. Table 5 contains the data.

|  |  |  |  |
| --- | --- | --- | --- |
| Student Id | Age | Native language | Passed |
|  |  |  |  |
| 1 | 14 | Finnish | 1 |
|  |  |  |  |
| 2 | 15 | Finnish | 0 |
|  |  |  |  |
| 3 | 13 | Turkish | 0 |
|  |  |  |  |
| 4 | 13 | Finnish | 1 |
|  |  |  |  |
| 5 | 16 | Finnish | 1 |
|  |  |  |  |
| 6 | 15 | Arabic | 0 |
|  |  |  |  |
| 7 | 14 | English | 1 |
|  |  |  |  |
| 9 | 14 | Finnish | 1 |
|  |  |  |  |
| 10 | 15 | Finnish | 0 |
|  |  |  |  |

Table 5. Example data for feature engineering.

In this example, the variable “Native language” has four possible values, which are “Finnish”, “Turkish”, “Arabic', and “English”. However, the vast majority of this variable has the value “Finnish”, and rest of them form a small group. In such a case, variable might be modified so that possible values are “Finnish” and “Other”. In an environment where education language is Finnish, modifying the data in such way does not affect the importance of the variable, while making it simpler. This is a manual process, and deciding the usefulness and results of such modification requires domain knowledge. In this thesis, feature modification is done by creating a new custom variable as a function of different variables.

**3.4. Evaluation methods**

In order to evaluate the effectiveness of a prediction model, predicted values must be compared with actual values. There are multiple criteria for prediction effectiveness. Table 6 shows the possible results of prediction for binary values.

|  |  |  |
| --- | --- | --- |
|  | Predicted as True | Predicted as False |
|  |  |  |
| Actually True | True Positive | False Negative |
|  |  |  |
| Actually False | False Positive | True Negative |
|  |  |  |

Table 6. Possible prediction results.

http://www.htmlpublish.com/newTestDocStorage/DocStorage/b9e146adfbf3483fb55fb09d5f1426e2/GRADU-1498472565_images/GRADU-149847256519x1.jpg

The matrix that shows the possible prediction results is called a confusion matrix. There are different evaluation criteria that can be obtained from these values. One is accuracy, defined as (Powers, 2011):

*Accuracy*= *TP*+*TN*

*TP*+*TN*+*FP*+*FN*

Accuracy is basically the ratio of correct predictions. However, accuracy has limitations in evaluating the prediction performance. Especially, accuracy does not show how the cases of minority class are classified, when the class distribution is imbalanced. As an example, a data set that contains 100 students, 90 of which has passed the exam, might be considered. A crude prediction (known as the majority rule) that does not use any machine learning method, but instead predicts that every student will pass the exam, has 90% accuracy. The model should perform better than just guessing that each case belongs to the majority class.

In this report, three other criteria are used. Two of them are precision and recall, which are defined as:

|  |  |  |  |
| --- | --- | --- | --- |
| *Precision*= | *TP* | *Recall*= | *TP* |
| *TP*+ *FP* | *TP*+*FN* |

Precision and recall are used together to make a better evaluation. The main idea is that accurately predicting positive outcome is not enough. A good predictive model must have a good combination of successful positive predictions and successful negative predictions. The third criteria that is used by this thesis is called F-measure, and it is defined as:

*F*=2 *Precision Recall*

*Precision*+*Recall*

F-measure is a way of having a single value that takes both precision and recall into account. F-measure is the final evaluation criteria for comparisons in this thesis.

**Chapter 4: System Design**

**4.1 List of Diagrams**

Figure 4.1.1 Use Case Diagram

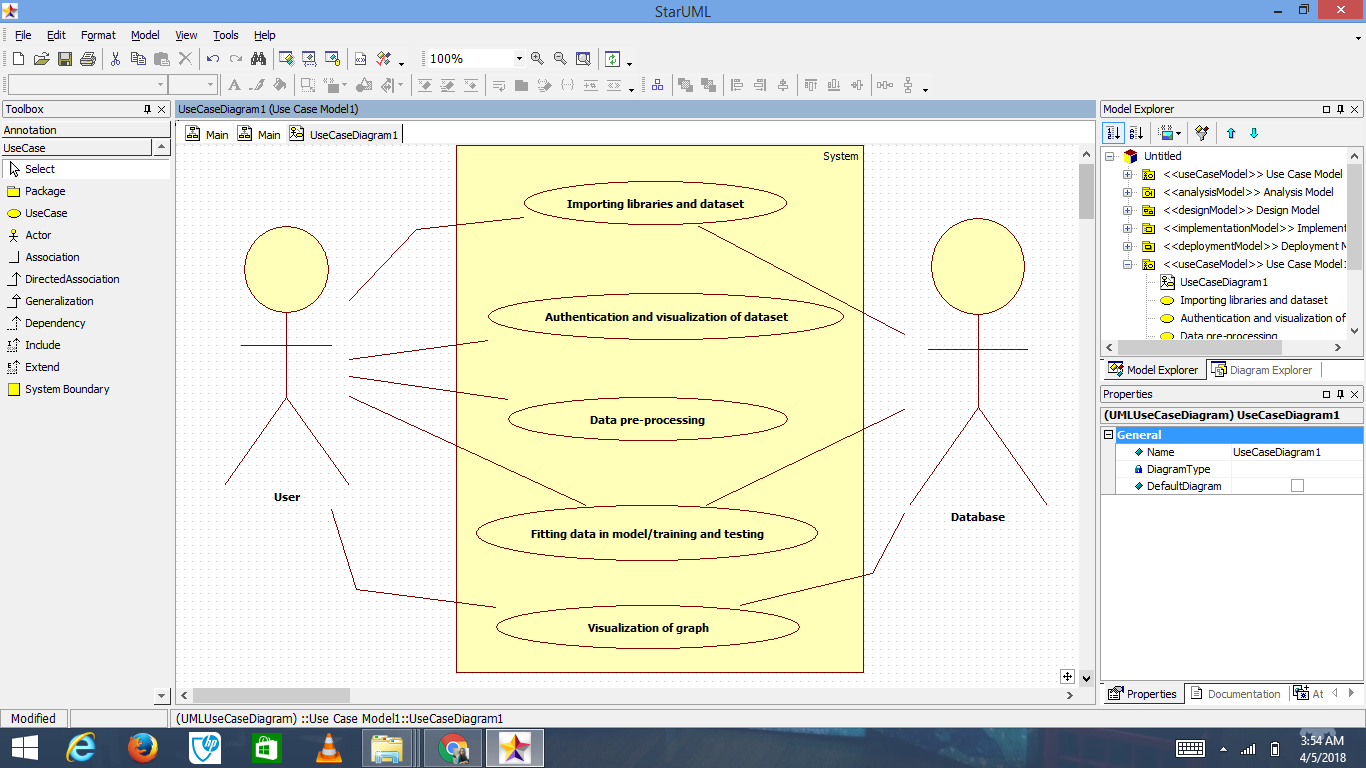


Figure 4.1.2 Collaboration Diagram

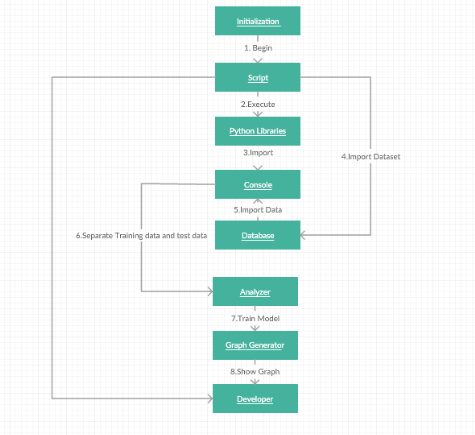
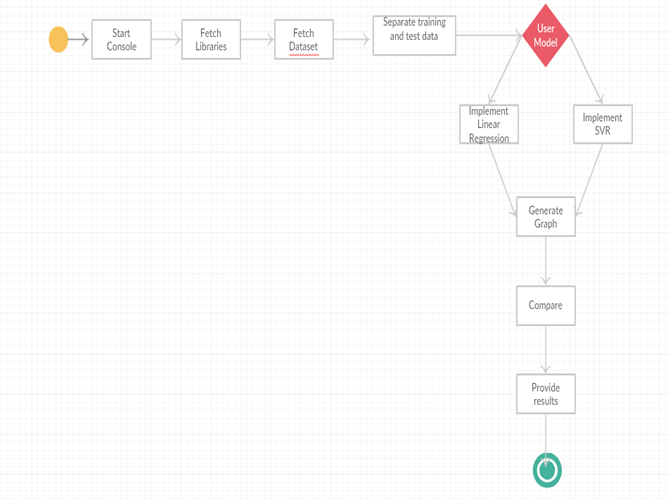


Figure 4.1.3 Activity Diagram



**Chapter 5: About Dataset**

We used the dataset from the file “xAPI-Edu-Data.csv” which has been made available to us by the UCI data repository

The data is separated into two sets: • Training data • Test data

Each row of these datasets carries information about one candidate’s credentials. There are 16 features (independent variables) and 1 dependent variable from which we are going to predict grade. It contains information about 480 students from various countries, mostly in the Middle East. Training data has 385 rows and test data has 96 rows. The data has a total of 17 variables. The dependent variable column is empty in the test data to be predicted. For our analysis, we divided our training data to two sets: training and validation datasets.

It has 481 rows.

(see Table 7).

|  |  |  |
| --- | --- | --- |
| Column | Description | Type |
|  |  |  |
| Gender | Gender of student | Nominal |
|  |  |  |
| Nationality | Nationality of student | Nominal |
|  |  |  |
| PlaceofBirth | Country of birth for student | Nominal |
|  |  |  |
| StageID | Educational stage, for example Middle | Nominal |
|  | school, high school |  |
| GradeID | Grade level of the student | Nominal |
|  |  |  |
| SectionID | Classroom of the student | Nominal |
|  |  |  |
| Topic | Course topic | Nominal |
|  |  |  |
| Semester | Semester of the year | Nominal |
|  |  |  |
| Relation | Parent responsible for the student | Nominal |
|  |  |  |
| Raisedhands | Number of times the student raised hands | Quantitative |
|  | during the class |  |
| VisitedResources | Number of times the student visited the | Quantitative |
|  | course content |  |
| AnnouncementsView | Number of times the student checked new | Quantitative |
|  | Announcements |  |
| Discussion | Number of times the student joined the | Quantitative |
|  | discussion groups |  |
| ParentAnsweringSurvey | Did the parent answer the school surveys | Nominal |
|  |  |  |
| ParentschoolSatisfaction | Parents level of satisfaction for the school | Nominal |
|  |  |  |
| StudentAbsenceDays | Number of days the student has been | Quantitative |
|  | Absent |  |
| Class | Grade of student for the course | Quantitative |
|  |  |  |

Table 7. Variable descriptions for the first data set.

Nominal types have a specific set of values, while quantitative types can have values which can be ordered. Variable “Class” is the dependent variable, meaning it is the variable that the model is trying to predict. It can have three different values, which are “L”, “M”, and “H”. These represent whether the students are in Primary school, Middle school or High school respectively.

Before any feature engineering, a modification of dependent variables in both data sets was made. They were converted to binary variables. For the first data set, values “M” and “H” were converted to 1, value “L” was converted to 0. For the second data set, values equal to or greater than 10 were converted to 1 and values less than 10 were converted to 0. This way, we encoded categorical data into comparable formats to be able to implement the model.

**Chapter 6: Implementation and results**

The aim of the research was to compare different machine learning methods and feature engineering in the student performance prediction. The prediction models were created using python language. It is a language commonly used for machine learning applications. It has built-in functions for the two methods selected for this project, which are linear regression and support vector regression. It also creates the necessary output for evaluating and refining the results of predictions. The code written in the python language is run on an application called Spyder.

**Loading Libraries**

**#Loading the required libraries**

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

from sklearn import preprocessing, cross\_validation, svm

numpy

Type : Module numpy.\_\_init\_\_

It provides

1. An array object of arbitrary homogeneous items

2. Fast mathematical operations over arrays

3. Linear Algebra, Fourier Transforms, Random Number Generation

matplotlib

Type : Module matplotlib.\_\_init\_\_

This is an object-oriented plotting library.

A procedural interface is provided by the companion pyplot module, which may be imported directly. It is used to generate graphs and plot our predictions

sklearn

sklearn is a Python module integrating classical machine learning algorithms in the tightly-knit world of scientific Python packages (numpy, scipy, matplotlib).

It aims to provide simple and efficient solutions to learning problems that are accessible to everybody and reusable in various contexts: machine-learning as a versatile tool for science and engineering.

**# Importing the dataset**

*For importing the dataset into data.*

data = pd.read\_csv('xAPI-Edu-Data.csv')

arr = [ ]

for i in range(17):

if i != 1 and i !=4 and i!=7:

arr.append(i)

*f is the list of independent variables which we perform training and we match it with the test data.*

f = [0,1,2,3,4,5,10,11,12,13]

*The values of the data get stored in X*

X = data.iloc[:, arr].values

*dependent is used to predict the value of gradeID*

*As it is of object type, it cannot be used to perform operations.*

dependent = data.iloc[:, 4].values

*So, we need to convert it to array type.*

*To convert dependent object into an array.*

y = []

dependent = np.array(dependent)

*As gradeID starts with G- so to extract its quantitative value we have to remove G-.*

for i in dependent:

y.append(int(str(i)[-2:]))

*To convert y into array.*

y = np.array(y)

LabelEncoder*can be used to normalize labels.*

*OneHot encoding is needed for feeding categorical data to many scikit-learn estimators, notably linear models and SVMs with the standard kernels.*

from sklearn.preprocessing import LabelEncoder, OneHotEncoder

labelencoder = LabelEncoder()

for i in f:

X[:, i] = labelencoder.fit\_transform(X[:, i])

onehotencoder = OneHotEncoder(categorical\_features = f)

X = onehotencoder.fit\_transform(X).toarray()

*#* ***Splitting the dataset into the Training set and Test set***

from sklearn.cross\_validation import train\_test\_split

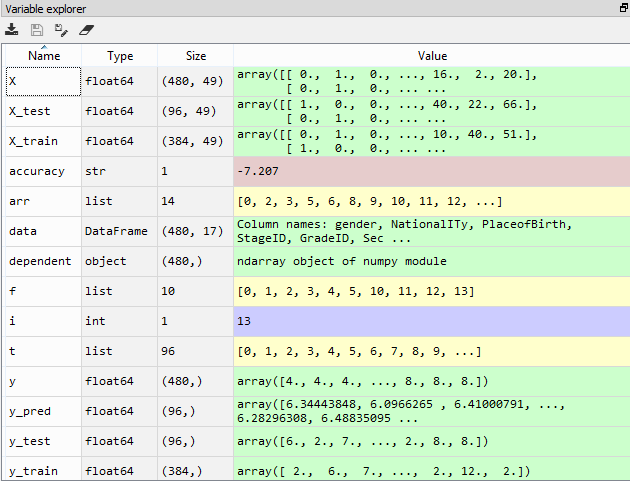
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.2, random\_state = 0)

y\_test = y\_test.astype(np.float64)

y\_train = y\_train.astype(np.float64)

y = y.astype(np.float64)

Variables generated



**Training various models**

#Training the **Linear Regression** model

***#fitting multiple-linear regression to training set***

from sklearn.linear\_model import LinearRegression

regressor= LinearRegression()

regressor.fit(X\_train , y\_train)

accuracy = regressor.score(X\_test, y\_test)

accuracy = "{:.3f}".format(accuracy \* 100)

print("Accuracy is " + str(accuracy) + " %")

***#predicting test set results***

y\_pred= regressor.predict(X\_test)

t = [i for i in range(96)]

***#ploting results on training set***

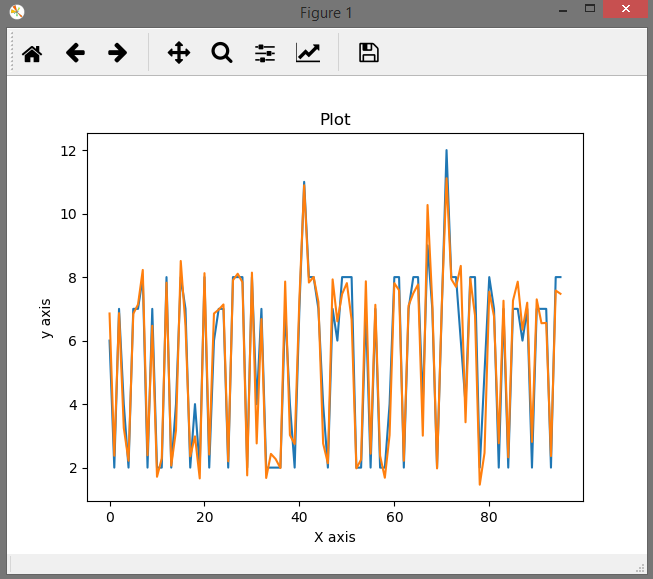
plt.plot(t,y\_test,t,y\_pred)

plt.title('Plot')

plt.xlabel('X axis')

plt.ylabel('y axis')

plt.show()



Accuracy is 94.120 %

Actual data

Predicted value

#Training the **Support Vector Regression** model

regressor= svm.SVR()

regressor.fit(X\_train , y\_train)

accuracy = regressor.score(X\_test, y\_test)

accuracy = "{:.3f}".format(accuracy \* 100)

print("Accuracy is " + str(accuracy) + " %")

***#predicting test set results***

y\_pred= regressor.predict(X\_test)

t = [i for i in range(96)]

**#ploting results on training set**

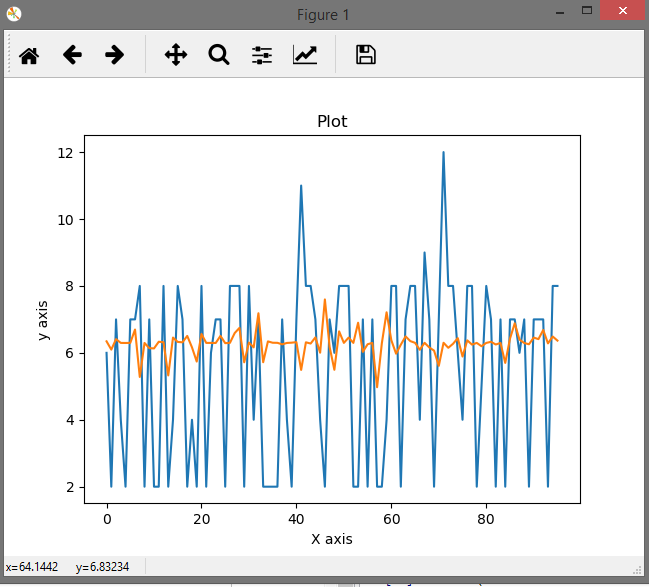
plt.plot(t,y\_test,t,y\_pred)

plt.title('Plot')

plt.xlabel('X axis')

plt.ylabel('y axis')

plt.show()



Accuracy is 7.207%

Actual data

Predicted values

As we can see **Linear Regression** gives the better results and **SVR** performs below expectations, but that maybe dueto small dataset or the way we imputed missing values.

**Chapter 7: Evaluation**

**Method comparison**

The first step in evaluating the results is to compare the machine learning methods in terms of their prediction performance. Tables 6.1 two machine learning methods for the data set.

|  |  |
| --- | --- |
| Method | Accuracy (%) |
|  |  |
| Linear regression | 93.3 |
|  |  |
| Support vector regression | -7.3 |
|  |  |

Table 6.1. Method comparison for the first data set, with raw data.

Machine learning models built with this data set were not as accurate as those of the data set. Furthermore, methods have a different order of success in this data set. This dataset has linear regression as the most effective model. With the engineered data, different methods show similar performances. However, raw data contains one exception when it comes to method performance similarity.

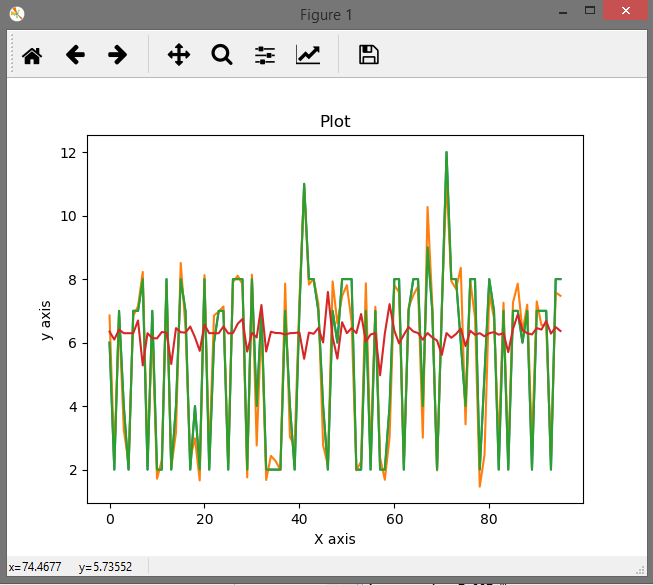


Figure: Comparison of accuracy of both algorithms

Actual data

Linear Regression

SVR

**Chapter 8: Screenshots of Project**

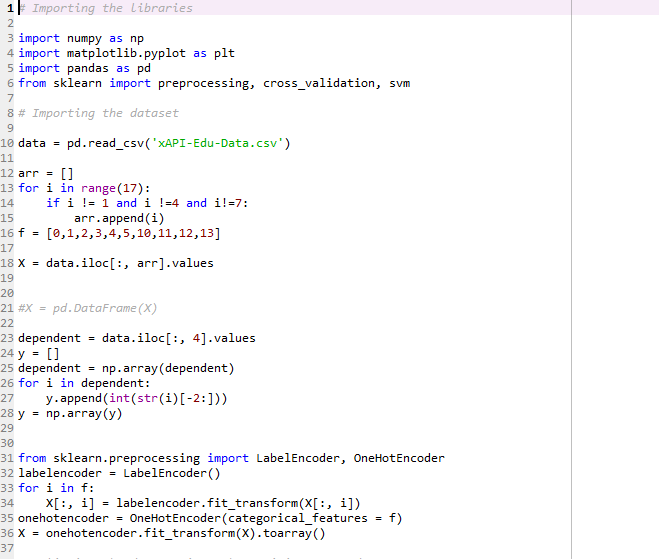
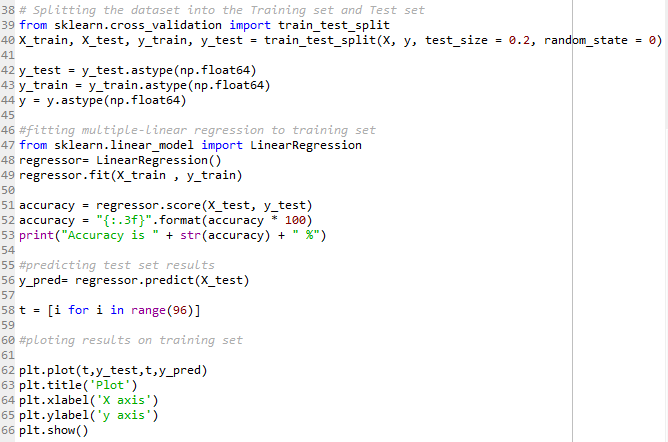


Fig 9.1 Code Snippet 1

Fig 9.2 Code Snippet 2

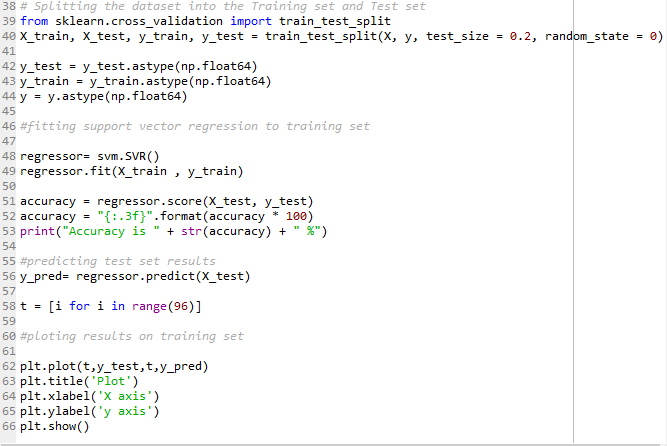


Fig 9.3 Code Snippet 3

**Conclusion**

The success of machine learning in predicting student performance relies on the good use of the data and machine learning algorithms. Selecting the right machine learning method for the right problem is necessary to achieve the best results. However, the algorithm alone cannot provide the best prediction results. Feature engineering, the process of modifying data for machine learning, is also an important factor in getting the best prediction results.

The aim of this project was to compare method selection and feature engineering, in terms of their ability to improve the prediction results. One data set was analyzed with two different machine learning methods, and their results were compared using four evaluation measures. Methods used were linear regression and support vector regression. For the evaluation of feature engineering, machine learning methods were applied to the raw and modified versions of the data separately. The main method of feature engineering was feature selection Feature engineering was done both with automatic functionality and manual interpretation of the data

The linear regression method gave much better results compared to the support vector regression method. Accuracy value for the first method is 94.120 %, while accuracy value for the second method is 7.207 %. Both methods used fare nearly identical, but the results are very different. This indicates that better methods cannot offset the limitations of the data.

The results of this study indicate that feature engineering provides more improvement to prediction results than method selection.

This shows that using machine learning is an effective way of predicting the student performance.

**Future Scope**

This research has certain limitations that must be noted. There was not an access to a dedicated student data set, and the study relies on public data sources. In addition, both data sets were small, having less than thousand records. A research that has access to more comprehensive data may offer more conclusive results.

Another area that future research can improve is the variety of the machine learning methods. This research used linear regression and support vector regression. Other methods, such as clustering and artificial neural networks can be used to have a better understanding of the importance of method selection.

Final area that can be improved is the process of feature creation. Since the data is limited, the amount of feature modification that can be made is also limited. Both data sources used in this research consists of a single table, and custom variables were created using variables from the same table. With a more comprehensive data

**References**

* [http://scikit-learn.org](http://scikit-learn.org/)  Machine Learning in Python
* <https://archive.ics.uci.edu/ml/>  UC Irvine Machine Learning Repository
* Spyder is the Scientific PYthon Development EnviRonment. It is a powerful interactive development environment for the Python language with advanced editing, interactive testing, debugging and introspection features and a numerical computing environment thanks to the support of *IPython* (enhanced interactive Python interpreter) and popular Python libraries such as *NumPy* (linear algebra), *SciPy* (signal and image processing) or *matplotlib* (interactive 2D/3D plotting).