# Final Year Project

Using machine learning to creating teaching statistics


Author: Matthew McDonald

Supervisor: Ron Austin

Birmingham City University

Bachelor of Science with Honours Computer Games Technology (Faculty of Computing, Engineering and the Built Environment)

# Abstract

The aim of this project is to develop a program that can evaluate student performance based on attendance and online resource interaction using multiple machine learning algorithms. All the algorithms will be ensembled together in an attempt to create a more accurate final model.

The created product proves that this type of program can be useful however it is dependent on its access to sizable amounts of student information as well as the final intended goal of the finished product. The final product of this project shows a good beginning point for specialised development. Further evaluation can be found in the conclusion section.

# Table of Contents

# Table of Figures

# Introduction

Many machine learning architectures have been designed to accommodate for a multitude of information types. Although there are many possible implementations of machine learning, this report will focus on methods that specifically adhere to statistical analysis on continuous data. The major topics that will be discussed include: what is statistical analysis and how they are currently used in the academic field, what is machine learning and how it applies to statistical analysis, and the current methods that are currently being used and where they succeed or fail compared to others. This paper seeks to focus on the key aspects required to create a successful machine learning algorithm designed for academic data processing.

## Research Aim

The aim of this project is to develop a program that can evaluate student performance based on attendance and online resource interaction. Multiple machine learning algorithms are to be created and be modelled to predict the student performances. Each algorithm will have access to the same data but will be developed independently to best develop each learning model. Finally, all the algorithms will be ensembled together in an attempt to create a more accurate final model to witch the results will be compared to the resulted collected from the other designed models.

## Research Objectives

1. Collect student data, specifically attendance:

   The first objective involves coordinating with university staff and project leaders to attain student data without breaching any data protection laws or other areas on concern.

2. Identify a collection of machine learning algorithms applicable to this problem:

   Identifying the models used is important as it will impact the final results greatly. Research for each model will be done to ensure each model is suitable for the problem presented.

3. Create the learning algorithms and train them on sample student data:

   Following the design steps outlined below, this objective is to create the sperate models so that they can use the data collected in the previous steps.

4. Refine and collate results though ensemble:

   Finally, each model will be further refined, as to say variables that can impact the final accuracy of the final output will be changed to maximise performance. Also,

all the separate models will be ensembled together in an attempt to create a superior model.

# Rationale

With the increased demand on the education sector, pressure on teaching staff increases proportionally. A study by Universities UK reported an increase of 0.4% between the academic years 2007-08 and 2016-17, a difference from 2.31 million to 2.32 million students across various levels and modes of study (Universitiesuk.ac.uk, 2019). Universities UK report also states the average salary for graduates in the UK was £10,000 more than non-graduates at £33,000 compared to £23,000 respectively. These facts alone are reason to suspect a continuation of this trend and without preparation it's not unreasonable to assume a situation where the quality of teaching decreases as a result of increasing student numbers and the need to accommodate for both them and the ever-evolving fields of study. This project seeks to make the first step to make available tools that can help to alleviate some of that pressure on professors and teachers alike by allowing them to better monitor their class without an increase in contact time so their time can be used as efficiently as possible.

# Literature Review

## What is statistical analysis?

The statistical analysis and AI design company SAS define statistical analysis as 'the science of collecting, exploring and presenting large amounts of data to discover underlying patterns and trends.' (Sas.com, n.d.). The aim of statistical analysis is to quantify the results of a task and ultimately rate the effectiveness of said task. The output data is often then displayed in a graph or table mainly for viewability purposes.

Statistical analytics can take many forms, including but not exclusive to: Correlation, the measure of association between variables, Paired T-test, the test of the differences between related variables, and regression, the prediction on if a change in one variable can predict the change in another. (Cyfar.org, n.d.). All specific implementations of analysis however, fit into one of two main groups: analysis of continuous data and the analysis of discreate data. Continuous data relates to data that cannot be counted but only measured. Common variables that are continuous include: time, speed, acceleration. Discrete data refers to data that can be counted such as the number of items in a bag or the count of people in a country.

Calculating and presenting underlying links between variables allows for statistical inference. Statistical inference is the ability to predict an outcome given certain circumstances. This method allow data to be created (although with less validity then collected data), to prove or disprove a given hypothetical given that it remains within the scope of the original investigation.

## What is machine learning?

TechEmergence released an article explaining their definition of machine learning:

"Machine Learning is the science of getting computers to learn and act like humans do, and improve their learning over time in autonomous fashion, by feeding them data and information in the form of observations and real-world interactions."(Faggella, 2018)

The definition itself is an amalgamation of definitions from top companies including: Nvidia, Standford, McKinsey & Co., University of Washington, Carnegie Mellon University. By amassing information from different area of expertise, both development and research, it can be concluded that the above definition of machine learning is reliable and even if some other definitions don't 100% agree with Faggella's definition, they will pull stong parallels between them.

Similar to statistical analysis, machine learning has many different forms of implementation. Machine learning algorithms can be subdivided into several major categories, each catering to different problems and approaching solutions uniquely. These divisions are:

- Supervised and semi-supervised learning
- Unsupervised learning
- Reinforcement learning

(En.wikipedia.org, 2018)

## Supervised and semi-supervised learning

Supervised and semi-supervised learning involves a system of mathematical formulae that are able to identify patterns within a sample test or training data. This training data includes both the expected inputs needed for future predictions and the measured outputs that has occurred from those inputs. This requires there to of been prior work done to generate the initial data with valid results. Semi-supervised learning differs slightly as not all of the initial training data requires an output for the model to train off. From training, that include multiple integrations, the algorithm will generate a formula that best describes the information provided.

Supervised learning includes classification and regression models for different situations. Classification methods are used when the output will fall within a predetermined set of values e.g. recognising a single digit or character will have a limit number of possibilities. Regression is used when the output values will fall within a range of numerical values.

## Unsupervised learning

Different from supervised learning, unsupervised learning does not require outputs for training. The information provided is processed to find structure within the data. When new data is added the algorithm reacts based on the presence or absence of the commonalities it identified in the original data.

## Reinforcement learning

“Reinforcement learning is an area of machine learning concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward.”

The cumulative reward references some in built mechanism to the specific implementation of reinforcement algorithm, where ‘good’ behaviour is rewarded whereas ‘bad’ behaviour is not. This pushes the machine to find and choose the best choice to maximise the reward.

## What is ensemble?

As machine learning techniques developed, more and more methods are created and improved. As new methods are created, commonly there are overlap in its intended use when compared to other methods and my both be suited to similar tasks. This is where ensemble is used; Ensemble is the method of compiling multiple predictive models together in the aim to obtain a better predictive model than any individual model. Ensemble is an example of a supervised learning algorithm and therefore must be trained to a specific problem.

## Machine learning in teaching

Machine learning is a powerful tool that, when used correctly can help find the underling links between inputs, map out complex data structures otherwise hard to find and define and create software agent capable of understand to a greater or lesser extend and identify the 'best' outcome. Machine learning is not limited to these ideas and can be engineered to solve many problems.

The area chosen by this study is teaching and how teaching and learning resources could be improved with the implementation of machine learning technologies.

A study in 2011, made use of six separate learning to evaluate student performance.(Kotsiantis, 2011). The algorithms they had chosen were:

- "Naive Bayes algorithm was the representative of the Bayesian networks"
- "The back-propagation algorithm with momentum was representative of the artificial neural networks"
- "The RIPPER algorithm was the representative of the rule-learning techniques"
- "The 3NN algorithm, with Euclidean distance as distance metric as instance-based learner"
- "From the decision trees, C4.5 algorithm"
- "Finally, from the SVMs we have selected the Sequential Minimal Optimization"

With accuracy as their intended goal, they combined the results of these six algorithms using a voting system, resulting in a more accurate representation of the datasets. The completed project resulted in the production of a JAVA application. The application asked several details of the user, including the personal data of the student, and two separate sections for each of the semester's grades the user had already received. From this a metric was calculated between zero and twenty was calculated and a response was given back to the user. The classification of these scores are as follows:

- “’Fail’ stands for student’s performance between 0 and 9.”
- "’Good’ stands for student’s performance between 10 and 14.”
- "’Very good’ stands for student’s performance between 15 and 17.”
- "’Excellent’ stands for student’s performance between 18 and 20.”

The conclusion of this study was positive as the study was able to “gain insights about student progress and recommend possible actions”. It goes on to mention that an extension to this project would include I more in dept review of the student data, looking to include other information sources or grading criteria in addition to those currently used.

The study provided a great beginning as to what machine learning can do in predicting student performance. By utilising multiple algorithms to improve accuracy allow for the end result to much more reliable in its predictions. However, the studies decision to utilise the “default

values of all learning parameters.” within the algorithms themselves mean the current completed project could be further fine-tuned to increase accuracy even further.

## Conclusions

Machine learning and statistical analysis are powerful tools for analysing large amounts of data. Kotsiantis study is 2011 had proven to a greater or lesser extent that this technology works when applied to the teaching sector. However, before a large-scale adoption of machine learning to help predict student performance, changes and refinements need to be made. The Kotsiantis study was designed as a proof on concept as they had made little to no adjustments when refining each model from its default parameters. This study seeks to continue on the Kotsiantis study by adopting its design architecture of multiple learning techniques coalescing into one predictive model as well as address some of the underdeveloped aspects of the study.

# Development

## Requirements

The requirements for the development of this project are simple to develop machine learning algorithms. These are the minimum requirements:

- An IDE or developer tool for creating the algorithms. Although not strictly required, an IDE is heavy recommended as to simplify the process of writing code and constructing the final product. In this project Spyder has been selected
- Other premade machine learning scripts and tools. Once again, although not required, many scripts have already been created with a high level of complexity and efficiency that can aid/quicken development. This project has selected to use Pandas and Anaconda as they link into Spyder well.

## Original Design Stage

The original design revolved around the development of a single algorithm that would intake student Moodle data for multiple modules and create a breakdown of the students' academic progress for each module.

## First Revision

After preliminarily research multiple features of past experiments highlighted potential issues and improvements on the initial design.

Firstly, the system created will use multiple algorithms and use ensemble to collate the outputs of the multiple algorithms at create a more accurate response.

Secondly, the system will focus on overall performance as opposed to the break down over multiple modules. The data provided consists of the marks and attendance of students over several computer science related subjects and as a result, specific module data is impossible to extract.

## Learning Models

## Linear regression

Linear regression is a learning model that is commonly used in predictive analysis. The design pattern for linear regression is to examine whether a single or set of variables can be used in predicting an outcome or dependant variable. The simplest implementation, the simple linear regression, involves attempting to correlate information between one dependant variable and one independent variable defined by the formula y=c + b^x , ”where y = estimated dependent variable score, c = constant, b = regression coefficient, and x = score on the independent variable.” (Statistics Solutions, 2019)

This model has been selected due to its high compatibility with the circumstances of this project. As the data provided gives attendance and marks for each student, these variables will be used as the independent and dependant variables respectively.

## K nearest neighbour

K nearest neighbour (KNN) is an essential example of a classification algorithm and belongs to the supervised learning domain. Its main application is in pattern recognition, data mining and intrusion detection. The main advantages of KNN that can be exploited and used for this project are as follows:

- KNN is intuitive and simple: KNN is easily understood and implemented, as the aim of this project is to develop tools for students and learning professionals, its ease to understand is of high importance.
- Constant evolution: KNN us a memory-based learner and therefore adapts well to new information and given that academics is a continuous sector with many students per year, a model that can improve itself as time goes on is highly desired
- Multi class implementation: Although slightly out of the scope of this iteration of the project, the ability to use multi class data points as easily as single class data points means that this model can be used effectively if the scope of the project was to expand, e.g. was to include online times or other Moodle information about the students without having to completely remake the system.

This model lends itself well to the current implementation in this project as well as any further development or refinement due to its versatility and has therefore been selected as a second model.

## Neural Networks

Neural networks are a type of learning algorithm that has been design to reflect how neurons work in the brain of humans. A neuron is the building block of a neural network and its job is simple signal processing from connected nodes. A large collection of these nodes, separated into layers, are connected together, their connections to other nodes are given weightings signifying their importance to the

analysation of a small portion of the overall problem. The first layer consisted solely of the inputs of the system often normalised between one and zero, the final layer represents the outputs of the system and can be a single node or multiple nodes each representing different things. In between these layers are several more layers of neurons that are responsible for finding patters in the data. The middle section of nodes can be any in number in both number of layers and number of nodes in each layer.

An example of this is in the image recognition of an animal such as a cat. A node or cluster of nodes may in theory be responsible in identifying ears or ear like shapes on a given image therefore the weights given to connecting nodes that are ‘seeing’ unrelated parts of the animal are given a low weight as they are unimportant in the definition of an ear however may be given higher weights elsewhere as they still constitute a significant portion of what makes up a cat. During training these weights are changed repeatedly in an effort to improve accuracy though a method called back propagation.

Back propagation works by calculating the error associated with an incorrect guess by a neural network and then it attempts to adjust the weights in between connecting nodes to minimise the error the next time the same input is ran. Back propagation calculates the magnitude of on incorrect guess by using a cost function, that is unique per implementation. The limitation to this method is that is it not guaranteed to find the global maximum (the best possible weightings and bias) of a system and may find itself stuck within a local maximum, this could mean that post training, the system may be significantly worse than it could be resulting in inaccurate system.

This learning algorithm was not selected for the final project this is because neural networks traditionally need lots of data for an accurate model to be produced and given the sub 200 samples given for this project, it is likely that the results given by such an algorithm will be skewed and not represent the data accurately. This is caused by neural networks tendency to overfit to training data resulting in the network only representing those initial samples.

## Support-vector machine

A support-vector machine is a supervised learning model specialised in classification and regression analysis. It works by attempting to plot and divide the data into two separate groups. This type of model is designed for data that can fall into one of two groups. The vector that is plotted to divide the data is called the hyperplane. The best hyperplane is one that represents that the largest separation between the two classes while still isolating them form each other.

This model was not chosen for this project as the data is required to be discrete and fall into one of two categories and is therefore unsuitable to predict the marks of a student. This model however could be used if the initial aim of the program was to only predict if a student was to pass or fail and therefore could see some application in a similar program.

## Ensemble techniques

### Average

Averaging is the simplest form of ensemble techniques and, as the name suggests, each model predict the outcome of a given input and then each of the model's outputs are averaged uniformly to produce a single output. This was the chosen method due to the low number of samples and low number of models selected.

### Weighted Average

Weighted average is one step further, each model is rated and given a weight defining the importance of each model and their outputs are multiplied by this weight before the average of all outcomes is produced

### Max Voting

Max voting is another simpler form of ensemble learning. It works by each model predicts the outcome of a given input and the output that is most predicted is the output of the max voting.

The issue with this model in respects with this project is that is predicts the outputs of discrete outputs and is therefore unsuited to this project as the outputs to each model is in a continuous format.

### Stacking

Stacking is a technique of building a new model from several other models. It achieves this by first splitting the training data into 10. A model is then fitted to 9 of the parts and predicts the $10^{th}$, this step is repeated until each section has been given a prediction. The original model is then fitted the entire dataset where it predicts the outputs of the test set. At this point, the model has a set of predictions for the entire training set as well as the test set. These steps are repeated for several other models and you are left with the collection of predictions of the training set and a collection of prediction of the testing set. This data then becomes the new inputs for a new model or models where further predictions can be made. Each time the above steps is ran, the outputs are considered a new level, the original models being level zero, the model used to predict off of that model is level one. Stacking can have any number of levels

## Blending

Blending is similar to stacking but with one major difference, the training set is only split into two sections, the training set and the validation set. The chosen model is then trained on the training set and used to predict the validation set. The second level models now only use the predictions made from the validation set and the test set. Other than this, blending is the same as stacking. The disadvantage of this method is that each iteration (level), the training set will but cut considerably as the training set used to train for the validation set will be lost.

## Bagging

Bagging is an approach of combining multiple results of multiple models to get a generalised response. The first step involves a process called bootstrapping which is a sampling technique that creates multiple subsets of the original data the same size of the original data. It does this by using a method called replacement where individual records have a possibility to be the same training set more than once. An estimated 63.2% of data is a given bootstrap sample will be unique, the rest being duplicates. (Aslam, Javed A.; Popa, Raluca A.; and Rivest, Ronald L. (2007)). After the creation of these subsets, a single base model is ran on each of the sets in parallel and are independent of each other producing several predictive models. The final predictions are then determined by combining the predictions from all the models

Bagging was not selected as the chosen ensemble technique due to the small number of learning algorithms used. Bagging is designed to be used for multiple learning algorithms to give the best results, since only two models were used bagging would likely not outperform simpler techniques.

## Boosting

Booting is a sequential process of refinement of models as each model tries to correct the incorrectly predicted values of the previous model. It does this by uniformly applying weights to all the input values prior to any model training on it. The first model is then trained of this data and the incorrectly predicted values are given higher weights and the model is reran on the output of the previous model as it tries to correct the previous model. This can be done multiple times is a chain of different or similar models. This process is repeated several times until multiple models are created with the desired level of precision. A final model then is the weighted mean of all the models and becomes the model where future predictions are done from.

# Implementation

## Linear Regression Implementation

```
1  import pandas as pd
2  import numpy as np
3
4  from sklearn import linear_model
5
6  data = pd.read_csv('dataset.csv')
7  data.head()
8
9  X = data.iloc[:,[3]].values
10 y = data.iloc[:,4].values
11
12 lm = linear_model.LinearRegression()
13 model = lm.fit(X,y)
14
15 score = lm.score(X,y)
16 print(score)
17 print(model.coef_)
```

Figure 1: Linear Regression Code

The implementation of linear regression is simple. After importing the necessary libraries, we import the csv file with pd.read_csv(). After reading the data, it is split onto the x axis, independent variable (attendance) and the y axis, dependant variable (marks), each represented as the third and fourth columns on the csv table respectively. The model is defined as linear regression on line 12 and the model is trained on line 13 to the data provided.

```
9  X = data.iloc[:,3].values
10 y = data.iloc[:,4].values
11
12 lm = linear_model.LinearRegression()
13 model = lm.fit(X,y)
14
15 score = lm.score(X,y)
16 print(score)
17 print(model.coef_)
18
```

```
IPython console
Console 1/A
ValueError: Expected 2D array, got 1D array instead:
array=[ 57.34  40.    52.45  76.19  26.21  21.38  40.    69.33  76.51  63.64
  70.55  63.95  66.67  76.39  69.23  61.64  43.84  68.92  56.94  39.73
  56.34  63.01  60.27  57.82  51.72  60.69  43.15  74.48  64.63  50.
  45.58  61.38  64.39  24.19  76.64  82.96  78.79  55.56 100.    60.9
  57.89  89.39  87.12  81.34  70.8   30.3   30.65  78.36  44.7   57.89
  56.39  31.58  53.03  30.66  87.12  75.    81.95  25.56  49.62  64.41
  37.12  65.15  90.91  90.91  48.63  65.22  79.7   50.74  74.07  64.79
  31.82  70.8   58.82  33.08  38.35  65.52  54.01  70.32  52.21  42.31
  66.18  49.33  78.2   66.15  48.53  20.9   52.59  51.02  33.08  34.56
  79.41  61.94  83.09  58.96  55.97  71.92  52.05  62.07  85.52  63.01
  72.6   70.55  45.52  72.11  59.31  74.48  73.29  73.79  72.79  34.93
  68.97  63.7   80.69  36.84  80.    51.37  80.14  80.82  34.12  82.88
  82.88  70.75].
Reshape your data either using array.reshape(-1, 1) if your data has a single feature or array.reshape(1, -1) if it contains a single sample.
```

Figure 2: Linear Regression Error Code

The first issue encountered here was the result of the regression model expected a two-dimensional array as an input at x and resulted in the program failing to run. The issue stemmed from the difference outputs from data.iloc(). As shown in figure 2, when given a single dimensional array the script cannot run. The fix for this was to enclose the '3' on line nine inside of square brackets ( [] ) which is the notation given for arrays, making it an array of size one.

## K-Nearest Neighbour Implementation

```
1  import pandas as pd
2  from sklearn.model_selection import train_test_split
3  from sklearn.neighbors import KNeighborsClassifier
4  from sklearn import preprocessing
5  from sklearn import metrics
6
7
8  # Read dataset to pandas dataframe
9  dataset = pd.read_csv('DataSet.csv')
10 dataset.head()
11
12
13 LabelEncoder = preprocessing.LabelEncoder()
14 for comName in dataset.columns.values:
15     dataset[comName] = LabelEncoder.fit_transform(dataset[comName])
16
17
18 X = dataset.iloc[:, [3]].values
19 y = dataset.iloc[:, 4].values
20
21 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state = 4)
22
23
24
25 knn = KNeighborsClassifier(n_neighbors=10)
26 knn.fit(X_train, y_train)
27 y_pred = knn.predict(X_test)
28 print( metrics.accuracy_score(y_test,y_pred))
29
```

Figure 3: K-Nearest Neighbour Code

Figure 3 shows the implementation for K nearest neighbour in this project. Once again, the needed libraries and csv document is loaded into the program and the x and y vales are isolated. Line 21 splits the data into test data and training data to be used to measure the accuracy of the models. The model is created as 'knn' and is given N_neighbours equal to 10. The reason that this value was chosen is shown in figure 4, where an analysis has been done to estimate the potential best values for it.

```
LabelEncoder = preprocessing.LabelEncoder()
for comName in dataset.columns.values:
    print(dataset.columns.get_loc(comName))
    dataset[comName] = LabelEncoder.fit_transform(dataset[comName])


X = dataset.iloc[:, [3]].values
y = dataset.iloc[:, 4].values

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state = 4)


k_range = range(1,26)
scores = {}
scores_list = []

for k in k_range:
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train, y_train)
    y_pred = knn.predict(X_test)
    scores[k] = metrics.accuracy_score(y_test,y_pred)
    scores_list.append(metrics.accuracy_score(y_test,y_pred))


plt.plot(k_range,scores_list)
plt.xlabel('value of K for KNN')
plt.ylabel('Testing accuracy')
#scaler = StandardScaler()
#scaler.fit(X_train)

#X_train = scaler.transform(X_train)
```

IPython console

Console 1/A

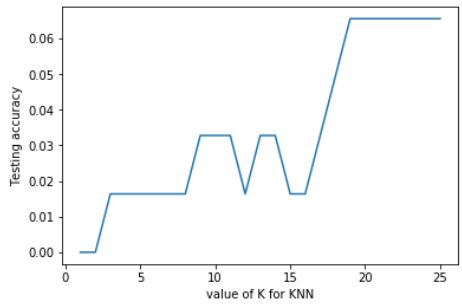


Figure 4: K-Nearest Neighbour N_Neighbours test

## Ensemble Implementation

The implementation of ensemble requires the previous models to be encapsulated into functions, named linearregression() and KNN(). They are each ran on the same data then an average is taken.

```
dataset = pd.read_csv('DataSet.csv')
dataset.head()


LabelEncoder = preprocessing.LabelEncoder()
for comName in dataset.columns.values:
    dataset[comName] = LabelEncoder.fit_transform(dataset[comName])


X = dataset.iloc[:, [3]].values
y = dataset.iloc[:, 4].values

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state = 4)
print("Average score:")
print((linearregression(X_test) + KNN(X_test)) / 2)
```

Figure 5: Ensemble Code

# Evaluation Of Product

The end result of this project represents an excellent start, but requires further development to practical use. Both the design and the process for development hold tremendous potential that this implementation has somewhat circumvented for one reason or another. The design ethos of multiple models ensembled into a single program shows great promise in predicting student performance based on other factors as it allows for an in-depth analysis of multiple factors that could affect a student's performance during their time studying.

This product however fails to capitalise on the main advantage of this design pattern as only a few models were chosen and its overall complexity kept simplistic. The main reason for this is due to the confines produced from the data set provided. Its small sample size as well as its low number of variables for testing excluded many training methods as they would have been overkill for this type of problem or would have easily resulted in overfitted models that ultimately are useless when used.

The availability of data is the biggest precursor to issues of this product. As said its small size limited the program making it unable to show off what the original design pattern could do. This is due to various issues such as data protection of student data as well as exact scope a designed product would be required to have to be useful to the teaching and academic sector. These issues can be solved though cooperation with universities as they hold huge collections of data of past students that could easily pass the thousands without issue, as well as strict adherence to data protection laws to ensure the anonymity of those involved.

With the size of the sample data with a solution, the models used can be changed. The exact models chosen will reflect the exact product that is intended to be developed. For example, a complete breakdown of individual performances, attendance and analysis of Moodle interaction would require a much more robust model such as neural networks or models that handle multi-class problems well such as k-nearest neighbour to be used. On the other hand, if the product was just to analyse a smaller subsection of a year such as a single module or the overall mark of a student over the year(s) they are in education then more computationally light models can be chosen to quicken development times and execution times for big sets of data such as a complete subjects students' performance as opposed to a singular or class based system. These instances and other situations require their own specific set-up to best exploit the ensemble model proposed in this project and as a result no exact recommendation for direction can be given those instances. For this implementation, if more data were available, the development of several other models would have been a fundamental goal of this project to produce the best product.

The ensemble method used falls for the same issues as the learning models that anything more complex seeks to overcomplicate the issue for no observable improvement. With the addition of a diverse collection of addition learning models, the implementation of a more complex ensemble technique would be required as

averaging the results could easily seek to over simplify the problem especially in a multi-class format where each model is producing results outside of just a predicted mark.

In conclusion, the product is a nice start and a good example at what this design model can do but lacks the necessary robustness required for a large-scale deployment. With the increase in data points, more in-depth models would have to be used to maintain accuracy. At the same time, more complex ensemble techniques would have to be used to prevent the new models from suffering any penalty to their scores. For future projects of a similar nature an in-depth analysis of what the end system is required to do as the scope that the system will be able to interact with will as well as the size of the data will become a major factor in the selection of models and ensemble techniques used.

# Conclusion

With the increase pressure on the academic sector to provide a solid learning environment for students and to educate millions of students across countless subjects the importance that tools are made available to students and teaching professionals alike are proportionally increasing. Issues such as class sizes growing larger or the decrease in teaching professionals could quickly become a massive problem in the education sector as the efficiency of teaching and learning would quickly decline and that is why these tools are necessary in the future.

This project aim was to partially develop tools that could be used be professionals to improve the efficiency that they can teach their classes. Although this product cannot eliminate the underlying problems that could render the education system inefficient, I can help to combat and slow down the progress of the problem until a more complete and robust solution can be developed and implemented. The end product of this project has limited real world application, it does on the other hand showcase a solid foundation for future development. By repeating this project with a multitude of different learning algorithms and ensemble techniques, accurate and useful tools could be developed for professionals.

Another use for this type of work extends outside the teaching sector. With little changes, an improved product could be used by business to monitor worker performance and can be built into current systems employed by business that reward above average contributions from staff or to help highlight potential pitfall in their business practises before they become too big an issue.

# References

Universitiesuk.ac.uk. (2019). [online] Available at: https://www.universitiesuk.ac.uk/facts-and-stats/data-and-analysis/Documents/patterns-and-trends-in-uk-higher-education-2018.pdf [Accessed 24 Apr. 2019].

Sas.com. (n.d.). Statistical Analysis - What is it?. [online] Available at: https://www.sas.com/en_gb/insights/analytics/statistical-analysis.html [Accessed 27 Nov. 2018].

Cyfar.org. (n.d.). Types of Statistical Tests | CYFAR. [online] Available at: https://cyfar.org/types-statistical-tests [Accessed 27 Nov. 2018].

Faggella, D. (2018). What is Machine Learning? - An Informed Definition. [online] TechEmergence. Available at: https://www.techemergence.com/what-is-machine-learning/ [Accessed 28 Nov. 2018].

En.wikipedia.org. (2018). Machine learning. [online] Available at: https://en.wikipedia.org/wiki/Machine_learning#Types_of_learning_algorithms [Accessed 29 Nov. 2018].

Kotsiantis, S. (2011). Use of machine learning techniques for educational proposes: a decision support system for forecasting students' grades. Artificial Intelligence Review, 37(4), pp.331-344.

Statistics Solutions. (2019). *What is Linear Regression? - Statistics Solutions*. [online] Available at: https://www.statisticssolutions.com/what-is-linear-regression/ [Accessed 15 May 2019].

Aslam, Javed A.; Popa, Raluca A.; and Rivest, Ronald L. (2007); *On Estimating the Size and Confidence of a Statistical Audit, Proceedings of the Electronic Voting Technology Workshop (EVT '07), Boston, MA, August 6, 2007. .*