# Questions and Report Structure

## 1. Classification vs Regression

Its a classification problem since we need to find out whether a student will pass the exam or not.

This is a classic case of Classification where we need to divide students in 2 classes : Pass or Fail

## 2. Exploring the Data

Can you find out the following facts about the dataset?

- Total number of students : **395**
- Number of students who passed : **265**
- Number of students who failed : **130**
- Graduation rate of the class (%) : **67.09**
- Number of features : **30**

Use the code block provided in the template to compute these values.

## 3. Preparing the Data

Execute the following steps to prepare the data for modeling, training and testing:

- Identify feature and target columns
- Preprocess feature columns
- Split data into training and test sets

Starter code snippets for these steps have been provided in the template.

## 4. Training and Evaluating Models

Choose 3 supervised learning models that are available in scikit-learn, and appropriate for this problem. For each model:

- What is both the theoretical space complexity to represent the model and the time for the algorithm to make a prediction? You can either provide the big-O notation, or list several of the major features that may affect the algorithm and state if the largest driving factor is constant, linear, logorithmic, polynomical, etc in nature.
    - 

| Complexity | Space | Time |
|---|---|---|
| DecisionTree | O(n^2) : n is the number of features. For decision tree classifiers, every node has a yes or a no and hence can branch out in 2 ways, which can further branch out in 2 ways for second feature. Hence space can be total number of nodes as 1+2+4….2^n = n^2. | O(n) : n is the number of features. For a decision tree in a worst case it needs to scan all the features to get to the output hence time complexity will be O(n) |
| KNN | O(n) : n is the number of training data or points | O(logn) + k where n is the number of total points or traning data and k is the number of neighbours we need in knn |
| SVC | 1 or constant ( it will be a line or a quadratic or polynomial equation which separates space into 2 parts | 1 or constant time to predict. Once the modeled is trained, it will take a constant time to predict. |

    - 
- What are the general applications of this model? What are its strengths and weaknesses?
    - 

| | **Applications** | **Strengths** | **Weaknesses** |
|---|---|---|---|
| DecisionTree | 1 Both regression | 1 Simple to | 1 Can create over |

| | | | |
|---|---|---|---|
| | and classification<br>2) Commonly used in data mining | understand and visualize<br><br>2 required little data preparation<br><br>3 Predicting data is logarithmic as per the number of data points. | complex trees (Overfitting)<br><br>2 Building optimal decision tree is a NP-complete problem<br><br>3 They create biased trees if classes dominate, therefore need to balance the dataset |
| Knn | 1 Both classification and regression<br>2 Handwritten digits, satellite image scenes | 1 Very simple algorithm. Lazy prediction therefore learning time is constant. Since it does not learn when training data is provided and do all the calculation when the prediction is required, it a very fast learner<br>2 Successful in classification where decision boundary is very irregular since its non parametric | 1 Its Lazy, suffers from noise and bias.<br>2 It can suffers from outliers and hence for randomly splitted data with similar features, does not work well |
| SVC | 1 Used for regression as well as classification analysis.<br><br>2 In medical science to classify proteins | 1 They are very efficient in high dimensional space<br><br>2 Memory efficient, use a subset of points in decision making. Since it needs to draw a line looking at the margin, it can do so using a subset of points so | 1 Large number of parameter tuning is needed so that the model do not overfit or bias. For non linear planes, difficult to visualize<br><br>2 Large number of features greater than the training data gives poor performance. |

| | | | |
|---|---|---|---|
| | | memory efficient<br><br>3 provides many parameters and kernel functions to fine tune the model | 3 SVM do not directly provide probability estimates |

- Given what you know about the data so far, why did you choose this model to apply?

| | Reason to choose this model |
|---|---|
| DecisionTree | 1) Simple to visualize. All features are divided into yes or no so suits the input labels perfectly with the DecisionTree |
| KNN | 1) Students with similar features tend to show similar behaviour and pattern, so finding nearest neighbours can be very efficient. |
| SVC | 1) SVC clearly divides the plane into 2 halves or classes. There are a lot of parameters that we can tune up to solve problems of overfitting. |

- Fit this model to the training data, try to predict labels (for both training and test sets), and measure the F1 score. Repeat this process with different training set sizes (100, 200, 300), keeping test set constant.
- Produce a table showing training time, prediction time, F1 score on training set and F1 score on test set, for each training set size.

| **DecisionTreeClassifier** | Training set size | | |
|---|---|---|---|
| | 100 | 200 | 300 |

| | | | |
|---|---|---|---|
| Training time (secs) | .001 | .003 | .004 |
| Prediction time (secs) | .000 | .000 | .000 |
| F1 score for training set | 1 | 1 | 1.0 |
| F1 score for test set | .71 | .75 | .68 |

| **KNearestNeighbour** | Training set size | | |
|---|---|---|---|
| | 100 | 200 | 300 |
| Training time (secs) | .001 | .001 | .002 |
| Prediction time (secs) | 0.002 | .004 | .010 |
| F1 score for training set | .78 | .85 | .86 |
| F1 score for test set | .76 | .79 | .80 |

| **SVC** | Training set size | | |
|---|---|---|---|
| | 100 | 200 | 300 |
| Training time (secs) | .001 | .005 | .009 |
| Prediction time (secs) | .001 | .003 | .006 |
| F1 score for training set | .84 | .88 | .86 |
| F1 score for test set | .76 | .78 | .808 |

- 

**Note**: You need to produce 3 such tables - one for each model.

## 5. Choosing the Best Model

Based on the experiments you performed earlier, in 1-2 paragraphs explain to the board of supervisors what single model you chose as the best model. Which model is generally the most appropriate based on the available data, limited resources, cost, and performance?

**Based on the available data, limited resources, cost and performance, SVC seems to be the best model as it takes constant time in space and prediction time and with the increase in training data it increases its F1 score. The F! score of SVC is comparable to KNN looking at the table above but SVC has low space and time complexity. It does take more time in learning than KNN but overall I would prefer SVC as KNN is pretty prone to outliers and bias**

In 1-2 paragraphs explain to the board of supervisors in layman's terms how the final model chosen is supposed to work (for example if you chose a decision tree or support vector machine, how does it make a prediction).

**I choose SVM. SVM linearly separates or divides the plane or data in 2 classes based on their labels. Once this separation is done it can predict data based on the boundary of separation. It is like you are dividing people into 2 halves. One half has same type of people and other half has other type of people. Now when a person arrives, looking at this features it can predict whether he needs to be moved in first half or other half**

Fine-tune the model. Use gridsearch with at least one important parameter tuned and with at least 3 settings. Use the entire training set for this.

**I have used GridSearchCV above in my code with cv as 5, mkscorer as F1-score and parameters based on the classifier.**

What is the model's final F1 score?

**Final F1 score for SVM model for training data is .86 and for testing data is .74**