## SVM – Amelia Gustave

### Predicting the success of students, depending on socioeconomic factors.

### Methodology

Support Vector Machine is a Supervised Machine Learning technique which is specialised in both regression, and classification – in this case. The aim of this supervised algorithm is to create a decision boundary in the n-dimensional space, such that a new data point can be added and correctly classified in the future. In the context of student success, SVM will be used to classify the binary result variables: graduate (1) or dropout (0).

As the relationship between the input data and output data is non-linear, Support Vector Classifier (SVC) will be used. The main hyperparameters in kernelized SVM are:

The kernel: determines the type of decision boundary to be used, for transformation into a higher dimensional space.

* Linear
* Polynomial
* Radial Basis Function
* Sigmoid

Gamma: decides the extent of influence of a single training example.

* Low values – smoother decision boundary.
* High values – More complex boundary.

C: Determines the bias to variance trade off, as penalty for misclassifications are inversely proportional to margin size.

* Low values; Larger margin between classes.
* High values; Emphasises correct classification of classes.

### Filtering, Balancing and Splitting Data

To begin data manipulation, the ’Target’ column which was initially comprised of ‘Graduate’, ‘Enrolled’ and ‘Dropout’ was filtered to only include ‘Graduate’ and ‘Dropout’ as it was found that these would give the most exact result. The two categories were then replaced with integer values of 1 and 0.

Further filtering of the dataset was then done. Through inspecting the columns of the chosen dataset, the conclusion that ‘GDP’, ‘Inflation Rate’ and ‘Unemployment Rate’ bore minimal significance, was drawn and thus they were removed.

Using a confusion matrix, it was found that there was not a need to under sample the majority class, since the ratio of minor class/ majority class was not less than 0.25.

Finally, the data points were split into 80% training set, and 10% validation and testing sets. The model was fitted on the training set to begin with. This is where the parameters were fitted. Then this fitted model was used to predict the responses for the results in the validation set. Finally, the test set was used to provide an unbiased evaluation of the final fitted model.

### Hyper Parameter Optimisation

#### Gamma, C and Kernel Parameter Optimisation

To find the optimal values for kernel, Gamma and C, code was run, in which the precision, accuracy and recall was taking into account. According to said algorithm, the optimal parameters were as follows:

*Table 1: Optimal Parameters according to Code*

|  |  |  |  |
| --- | --- | --- | --- |
| Parameters | Kernel | C | Gamma |
|  | ‘rbf’ | 100 | 0.001 |

#### Kernel Selection

The following figures show the accuracy, precision and recall scores of both the training and validation set, for the kernel types; linear, rbf, poly and sigmoid. Gamma and C were set as default values.

A graph of a graph

Description automatically generated with medium confidence*Figure …*

A graph with blue and orange lines

Description automatically generated*Figure…*

A graph with blue and orange lines

Description automatically generated*Figure..*

While the code suggested that kernel = rbf, through graphical plotting, it was found that with kernel = linear, the accuracy, precision and recall scores for the validation set were maximised. A table was then made to compare the scores when kernel was both linear and rbf, for the validation the set.

*Table 2: Validation Set Values*

|  |  |  |  |
| --- | --- | --- | --- |
|  | Accuracy | Precision | Recall |
| Linear | 0.9155 | 0.9223 | 0.9055 |
| rbf | 0.9086 | 0.9192 | 0.8961 |

The above table confirms that the scores of the validation set are peak when kernel = linear, and thus this parameter was chosen. The validation set was chosen as the primary criteria as this ensures the model performs well with new data, and thus better fulfils the aim of the algorithm.

#### C Value Selection

The figures below show the accuracy, precision and recall scores of both the validation and training set with varying C (from 1 to 7).

A graph with orange and blue lines

Description automatically generated*Figure*

*A graph with orange and blue lines

Description automatically generated Figure*

A graph with orange and blue lines

Description automatically generated*Figure*

From the above figures, it can be observed that accuracy, precision and recall peak at values of 4 and 6. While the previously ran code, suggested that C = 100, it was decided to go with the graphical result obtained of 4. This was due to the fact that with larger values of C, there is increased risk of overfitting.

### Results

*Table 3: Results of SVM Model*

|  |  |  |  |
| --- | --- | --- | --- |
|  | Training | Validation | Testing |
| Accuracy | 0.91 | 0.92 | 0.96 |
| Precision | 0.90 | 0.90 | 0.96 |
| Recall | 0.97 | 0.97 | 0.98 |

### Discussion

The SVM model’s performance was evaluated based on training, validation and testing sets, and 3 scoring metrics – accuracy, precision and recall. The model was able to accurately predict if a student dropped out with 96% accuracy and precision, which was a 5% and 6% improvement from the training set. This result can be perceived as abnormal and could suggest that the model is underfit. On the other hand, this could show that the model excelled in correctly classifying graduate instances. The recall on the testing set was 98%, which implies near perfect instance identification. To conclude, the SVM model portrays positive performance, however, further adjustments and precautions could be made to ensure no underfitting has occurred.