Student Number: 184514 & Kaggle Team Name: Bayley C-S

1. Approach

The task of developing a binary-class classifier, insists that a supervised learning method must be used, since classification is a supervised problem. We know this since the training data given has associated labels, and we are attempting to predict an output for unseen data. The approach I have adopted is to use a Support Vector Machine (SVM). SVMs are a set of supervised learning methods, which can be used for classification.

1.1. How it works & why I'm using it

The objective of the SVM algorithm in this case is to find a hyperplane in a 2-dimensional space, that classifies the data points. When separating two classes, there are many possible hyperplanes. The objective is to find the plane with the maximum distance between data points of both classes, the maximum margin. A hyperplane is a decision boundary, with data points falling either side and are attributed to the different classes [1]. Support vectors are the data points closest to the hyperplanes, influencing the position & orientation of the hyperplane. Good support vectors maximise the margin of the hyperplane [2]. Similar to logistic regression, in SVM, if the linear functions output is greater than 1, we assign it one class, and if the output is -1, identified as other class. These threshold values ([-1 +1]), which act as the margin [3]. We want to maximise the margin between data points & the hyperplane. The cost function is used to train the SVM. Minimising the value of theta, increases accuracy. [4].
The high dimenstionality of the training data allows for SVMs to be more effective, as for most algorithms, data of a high dimensionality causes the calculations to become extremely expenisive computationally. Since SVM is very memory efficient, relative to other algorithms, these calculations are not as expensive. This is because only a subset of the training points are used in the decision process, meaning only these points need storing in memory, to be calculated upon, when making decisions [5]. SVMs are also very versatile. As the class seperation is not linear, the ability to apply different kernels allows for flexibility for the decision boundaries. [6]

2. Methodology

2.1. Training and Testing Data

Due to using one of the classifiers from the SkLearn SVM library, the fit and predict methods were used to simply train and test the classifier. Through the use of cross validation, I was able to test the accuracy of the classifier. Another way of testing the accuracy of the classifier can be to compare the proportions of class predictions to the test proportions file provided, however this is not a good way to optimise the classifier (could lead to overfitting). This is suggested by a non-accurate classifier having similar proportions, meaning lots of misclassification occurred to get those proportions. This is evident in my code that my proportions are not similar, but do score well in the submission. Argument supported at [7].

2.2. Using the additional training data

By loading the additional training data from the CSV into another variable and using the concat() method to combine the variables holding training data, I was able to have one large collection of training data. Using additional training data is a good idea, it reduces the classifier having to rely on assumptions and weak correlations. Since the additional data included missing values, it reduces the accuracy of a model or leads to a biased model, leading to inaccurate predictions. Through imputation of missing values, these inaccuracies can be reduced, in turn creating a more accurate model [8]. This was implemented in my code using a simple imputer from the SkLearn library. All missing values, in this NaN values, were replaced with the mean of the known values.

2.3. Resampling the Data

Originally, I intended to use under sample the training data to reduce the imbalance of memorable & non-memorable samples. Having an imbalance in samples can lead to overfitting and bad cross-validation. To implement this, I would use the ClusterCentroids library from imblearn. In practice, by undersampling, we solve class imbalance issue, and increase the sensitivity of our model. However, the accuracy decreased. A reason could be that we trained our classifiers using too few samples. The more imbalanced the dataset, the more samples discarded, therefore throwing away possible useful information. [9]

2.4. Pre-processing

I plan to use Scaling as my method of pre-processing the training data. I decided to implement the MinMaxScaler due to the default range being between 0 & 1, the values of a binary classifier. It transforms features by scaling each feature given the range. Normally, a dataset will contain features highly varying in magnitudes, units and range. It is important features are scaled as algorithms use Euclidian distance between two data points in their computations, this is a problem. A disadvantage of it is that it's sensitive to outlier. In my case, using MinMax did only slightly increase the accuracy [10].

2.5. Feature Selection (PCA)

To implement feature selection, I decided to use Principal Component Analysis. To do this, I used the SkLearn decomposition library. PCA is an unsupervised machine learning method. It's a common technique used for dimensional reduction while keeping the patterns in the data, with the goal to maximise variance [5]. A PCA advantage is that in addition to the low-dimensional sample representation, it provides a synchronized low-dimensional representation of the variables. Even though this reduces dimensionality, it increased the accuracy of classifier when ~800 components were discarded.

2.6. Grid Search

A Support-Vector Classifier requires many parameters, and I need to find the best combination which produces the best accuracy. To do this I used GridSearch from the SkLearn library. I decided to use this over Random Grid Search, as this is an exhaustive search over the data, meaning its more likely to produce optimal parameters as it would try every possible combination provided. [11]

Due to time-constraints, I was unable test more combinations, so this model could be improved, yet, I believe the parameters used were sufficient. The parameters used were: C value of 0.05 & a linear kernel. All other parameters default values. Kernel parameter selects the type of hyperplane used to separate the data. Parameter tuning and the affects of changing values available at [12].

3. Results & Discussions

3.1. Cross Validation (CV) Score

CV is a technique used to test the effectiveness of a machine learning model, when we have limited data. Advantages of CV include; Allows for parameter fine tuning, allows for more metrics, uses all of your data (avoiding under-fitting) and more [13]. GridSearchCV, I was able to test how the accuracy (CV score) fluctuates against a change in C value (Figure 1). The regularization parameter (C), where the strength of regularization is inversely proportional to C. The standard deviation is included to show max-and-min scores possible when not using the mean test score. The results in (Figure 1) suggest that as C increases past approx. 0.05, the CV score decreases, reducing accuracy of classifier. In addition, as C increases past 0.1, the size of the standard deviation 'error bars' get wider, meaning using the mean-score is no longer more beneficial to the model. The code also further explains that the maximum CV score is achieved at a C value around 0.03. To increase the CV score I could test the CV score against other parameters like gamma.

3.2. Learning Curve

A learning curve is a plot of model learning performance over experience or time, in this case learning efforts (experience). The metric used to evaluate learning in this case would be classification accuracy. The shape of the learning curve tells whether performance is improving, declining, stagnating or fluctuating [14]. From the results in (Figure 2), it would suggest that the plot of learning curves shows a good fit. This is because; the plot of training loss decreases to a point of stability & the plot of validation loss decreases to a point of stability and has a small gap with the training loss. It's not an overfit, as the plot of training loss doesn't continue to decrease with experience. I further don't believe its an underfit as it doesn't show a flat line or noisy values of relatively high loss. However, could be from an unrepresentative validation dataset. This is where the validation dataset doesn't provide enough information to evaluate the ability of the model. This possibly occurred as the validation dataset has too few examples as compared to the training dataset. In addition, it is identified by a validation loss that is lower than the training loss, indicating that the validation dataset is easier for the model to predict than the training dataset, possibly hinting at the domain adaption problem.

4. Figures

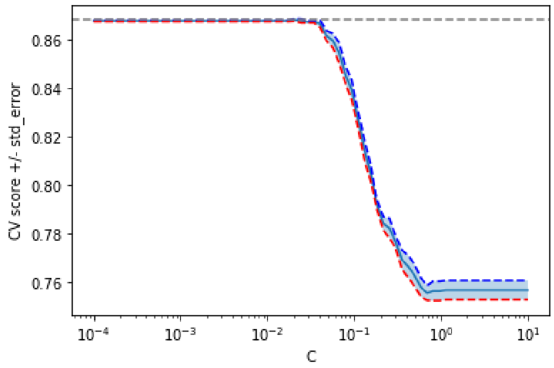

Figure 1 – Cross Validation Score against C value

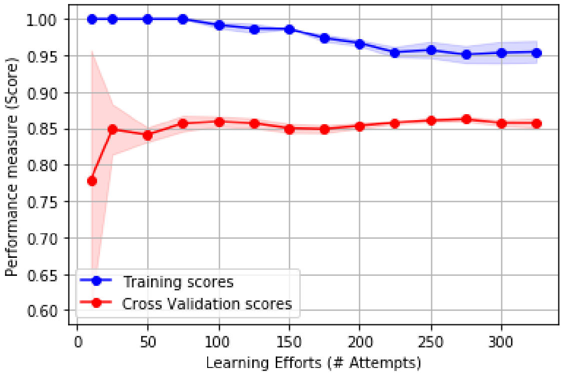

Figure 2 – Learning Curve Diagram

5. References

[1] R. Gandhi, “Introduction to Machine Learning Algorithms”, Towards Data Science, [Online]. Available: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47, 2018

[2] “Support Vector Machine - Classification (SVM)”, [Online]. Available: https://www.saedsayad.com/support_vector_machine.htm

[3] B. Stecanella, “An Introduction to Support Vector Machines (SVM)” Monkey Learn, [Online]. Available: https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/, 2017

[4] “SVM Machine Learning Algorithm Explained”, Free Code Camp, [Online]. Available: https://www.freecodecamp.org/news/support-vector-machines/, 2020

[5] M. Solirzano, “Working with high dimensional data”, Medium, [Online]. Available: https://medium.com/working-with-high-dimensional-data/working-with-high-dimensional-data-9e556b07cf99, 2019

[6] K. Dhriaj, “Top 4 advantages and disadvantages of Support Vector Machine or SVM”, Medium, [Online]. Available: https://medium.com/@dhiraj8899/top-4-advantages-and-disadvantages-of-support-vector-machine-or-svm-a3c06a2b107, 2019

[7] F. Harwell, “Classification vs. Prediction”, Statistical Thinking, [Online]. Available: https://www.fharrell.com/post/classification/, 2019

[8] S. Ray, “8 Proven Ways for improving the “Accuracy” of a Machine Learning Model” Analytics Vidhya, [Online]. Available: https://www.analyticsvidhya.com/blog/2015/12/improve-machine-learning-results/, 2015

[9] M. Altini, “Dealing with imbalanced data: undersampling, oversampling and proper cross-validation”, Marco Altini Blog, [Online]. Available: https://www.marcoaltini.com/blog/dealing-with-imbalanced-data-undersampling-oversampling-and-proper-cross-validation, 2015

[10] S. Asaithambi, “Why, How and When to Scale your Features”, Medium, [Online]. Available: https://medium.com/greyatom/why-how-and-when-to-scale-your-features-4b30ab09db5e, 2017

[11] S. Chan & P. Treleaven “Handbook of Statistics”, 2015

[12] M. Fraj, “In Depth: Parameter tuning for SVC”, Medium, [Online]. Available: https://medium.com/all-things-ai/in-depth-parameter-tuning-for-svc-758215394769

[13] D.Shulga, “5 Reasons why you should use Cross-Validation in your Data Science Projects”, Towards Data Science, [Online]. Available: https://towardsdatascience.com/5-reasons-why-you-should-use-cross-validation-in-your-data-science-project-8163311a1e79, 2018

[14] J. Brownlee, “How to use Learning Curves to Diagnose Machine Learning Model Performance”, Machine Learning Mystery, [Online]. Available: https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/, 2019