Student Number: 181345 & Kaggle Team Name: Ares

1. Approach

Before committing to a final machine learning approach, multiple models were implemented with their corresponding default values with the training data without the additional dataset or confidence labels. These included a Linear Perceptron Classifier, Random Forest Classifier and an ensemble voting classifier. This was done prior to implementing any pre-processing, under sampling, feature selection or cross validation as to set a control. This provided a clear comparison of the performance and accuracy of each model to commit to. Although each model exceeded the expectation set, the SVC surpass the other models, which lead to the final decision of choice.

The support vector machine approach is a supervised learning model that utilizes learning algorithms to tackle a given classification problem. A variant of kernels could be implemented within the parameters of the SVC, including Gaussian functions [1]. When the support vector machine is passed with provided training data, it will map it to a higher dimensionality space to draw a clear distinction between classifications. In addition, if the model is given unseen testing data, it will place these values also within the sane feature space so it could map them to one of the given classifications, in this case 1 or 0 (sunny or not sunny). Support vector machines also allows the manipulation of the kernels in a flexible manner [2], proving them to be a strong candidate for the approach as these available parameters can be optimized to solve the present task. Furthermore, support vector machines are well suited for dealing with high dimensionality when paired with the appropriate kernel, thus appropriate for the given dataset.

2. Methodology

2.1. Pre-processing

Normalization was used in the pre-processing for the support vector machine to change the values within the dataset to a common scale. This was simply applied with the sklearn toolkit using the 'Normalizer' fit function to pass in the training and test features. Pre-processing such as binarization, minmax and scaling were also tested prior to using normalization, although standardizing the data prior to the model training would theoretically be suitable, Given the variation of range between the CNN an d GIST features, normalizing the data yielded the most robust model. Due to the basis of the support vector machine classification approach, once the data is normalized, it will be able to distinguish the training data lying on the class decision boundary for discrimination. The rest of the pre-processing consisted of utilizing the additional dataset, test proportions and annotation confidence provided. This will be further developed later in this section.

2.2. Data undersampling

Before training the final model, it is imperative to handle any unbalancing. When utilizing the additional training data, it is evident to observe a great imbalance between the positive (1) and negative (0) classes. Specifically, the additional training data contained more negative classes compared to the existing training data, which had classes that are more positive. In order to balance the classes out before training the model, a simple random under sampling was implemented using 'RandomUnderSampler' from imblearn toolkit. This would randomly shuffle the training data and scale the input vectors individually to uniform norm.

2.3. Feature Selection

Given the high dimensionality of the features, it seemed necessary to implement a principal component analysis algorithm to reduce the dimensionality of the features by transforming them into smaller sets containing most of the information of the original set. However, it is important to note that reducing the dimensionality comes at the expense of accuracy, yet it will provide computational ease. This was implemented with the sklearn toolkit 'PCA' that can take multiple parameters including the 'n_components' which represents the number of components. An exhaustive grid search was implemented to find the most optimal 'n_components' value which was used for the final model.

2.4. Model Selection

Multiple parameters are available when using the support vector machines that will affect the overall model. Notably, the choice of kernel, gamma and C value and tuning will affect the final performance of the classifier. In order to generate the most accuracy support vector machine, a combination of random grid searches and exhaustive grid searches were utilizes to obtain the best hyperparameters for the final model. The different possible kernels, gamma and C values were passed through initially a random grid search which outputted the best parameters. These parameters were then passed through an exhaustive grid search this time, which also yielded an optimized hyperparameter. Finally, the previously obtained parameters were passed through a

second exhaustive grid search to obtain the most optimized parameters pass through the final model. This long but stable process returned these sets of parameters: {'C': 1.5, 'gamma': 0.001, 'kernel': 'linear'}.

2.5. Dealing with NaN values

The supplementary dataset contained a large portion of missing values (NaN) across the 4096 dimensional features. Whilst some of the GIST and CNN features were missing, the prediction values were intact. To expand the training data and take advantage of the increase, the NaN values within the additional dataset were imputed using the mean of all the values across the dataset for the given feature. This was implemented at the pre-processing stage using the 'fillna' function in python and taking the mean value. Thus, the entire dataset, including the additional training and original, was used which is a better representation of the feature as a whole when making the classification.

2.6. Annotation confidence

The confidence of the annotation given to each vector (image) was labelled either 1 or 0.66. The former values were taken into account and the latter ones were removed. This increased the overall performance of the final model.

3. Results and Discussion

3.1. Most optimal model

In comparison to one grid search, optimizing through multiple grid searches yielded the highest performance with the support vector machine, which returned a Kaggle accuracy of 0.67%. The best model constitutes the combination of all the methodologies described above.

3.2. Optimizing Hyper-Parameters

Initially, the learning curve obtained after the first optimization the training score is quite low whilst the cross-validation score is increasing gradually over time yet plateaus towards the end. This suggests a high bias thus increasing the complexity of the model would likely improve the quality of the classifier.. 'C': 1.0, 'gamma': 0.1, 'kernel': 'linear'} model fitted with these parameters.

The kernel parameter selects the type of hyperplane used to separate the data. The gamma value is the parameter for non-linear hyperplanes and the C value is the penalty parameter of the error term. Therefore, the goal was to find the most optimal hyper-parameters to implement in the final model. Figure 1 depicts the final optimization parameters obtained after 1 randomized grid and 2 complete grid searches.

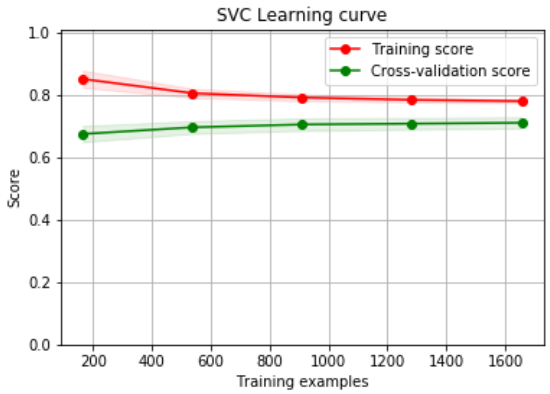

Figure 1. Learning curve of SVC with most optimized hyperparameters. {C = 1.5, kernel = linear, gamma = 0.001}

3.3. Validation

Cross-fold validation is used in the support vector machine and cross validation score where cv = 5 is calculated to help evaluate the performance of the classifier. This validation ensures that the classifier is not overfit to ta subset of the data.

Figure 2. Cross validation accuracies of SVC

	SVC	SVC with additional data
Accuracy	0.6330039525697%	0.71642512077294%

As is clear from figure 2, using the additional data during the pre-processing of the data produces a better classifier. A similar trend exists when using the confidence labels as sample weights.

3.4. Discussion

The support vector machine could be improved by increasing the instances of "grid-search method in cross validation to select the best parameters [3]. Furthermore, the hyperparameter optimization could be applied to different models which could yield a higher accuracy and robust model. Addressing the domain adaption problem could have also increase the overall performance of model. It is noted that the training data has been collected from the sea-side whereas the testing data has been collected from urban environments. This problem refers to the "dissimilarity between the source and target domains" [4]. Hence, the disjoint between the training data and testing data could lead to a unreliable classifier. A possible solution for this would be to use a multi-domain model instead of a single domain one where the training data and testing data are from different distributions. However, one of the major risks present whilst implementing a multi-domain model is the covariate shift [5], which can lead to unreliable predictions

References

[1] Srivastava, Durgesh, and Lekha Bhambu. "Data Classification Using Support Vector Machines"..

[2] Auria Laura, and Moro Rounslan A. "Support Vector Machines (SVM) as a Technique for Solvency Analysis". Deutsches Institut fur Wirtschaftsforschung. August 2008. Pg 7.

[3] Srivastava, Durgesh, and Lekha Bhambu. "Data Classification Using Support Vector Machines".

[4] Krijthe, Jesse, and Wouter Kouw. "Feature-Level Domain Adaptation." *Feature-Level Domain Adaptation*, Sept. 2016.

[5] Sugiyama, Masashi, et al. "Importance-Weighted Cross-Validation for Covariate Shift." *Lecture Notes in Computer Science Pattern Recognition*, 2006, pp. 354–363., doi:10.1007/11861898_36.