Support Vector Machines (SVM)

Madhu Sujan Paudel
Business Division
Caldwell University
Caldwell 07006, USA
mpaudel@caldwell.edu

***Abstract*—** Computer and the ways of computing have always been fascinating. In many big-data systems large amount of information are recorded and stored for myriad purposes. Extracting the information from these bulks of data can be a challenging task. Classification can be a good way of extracting unless it is done by using human effort. So we let the machines do this tasks by teaching them certain algorithms. The whole process of teaching a machine to learn something from the data and using that experience in an efficient way is called machine learning. Using machine learning for classification purpose is an effective way of identifying information from the data. In this project my main task is to classify the data and predict the outcome through the use of machine learning. For that purpose I will use Support Vector Machines for my purpose.

Keywords: SVM, unsupervised, machine learning, classification problems

1. Introduction

Data mining is the extraction of implicit, previously unknown and potentially useful information from data. The main purpose of data mining is to classify the raw data and uncover relationship in data to predict an outcome. Classification of data is the process of organizing data into categories for its most effective and efficient use. In data classification a given set of data is given, called a training set where each record consists of several fields or attributes.

One way of classifying a dataset using machine learning model is use of SVM (Support Vector Machine). Since Support Vector Machine were first purposed by Vapnik, they have been shown to build accurate models with practical relevance for classification, regression and novelty detection. [**1**]. As a result SVM is capable of performing linear or nonlinear classification, regression, and even outlier detection. There has been successful use of SVM for classification, facial recognition, bioinformatics, and text categorization. As a classification tools, SVMs using the idea of kernel substitution has become more popular.

SVMs is a class of data driven machine learning approach that deals with predictive binary classification, i.e. the assignment of class labels to unlabelled data. Using a large set of observations with known labels, SVM finds the maximum margin function that separates the observations into two classes where each observation is a multidimensional space of feature measurements. New unlabelled data are then assigned a class based on their geometric position relative to the classifier function. Given the vast amount of complex features that modern systems use, finding the classifier function often requires the simplification of the features space by identifying dimensions that have the most distinguishing power. [**2**]

In this paper we classify a dataset using SVM and classify that dataset and predict the outcome from the test datasets. The aim of this study is to develop an efficient method that can provide more successful classifications especially in the datasets containing multiple features. For this purpose, a non-linear SVM classifier with the combined kernel is used.

2. Literature Review

I will be taking refrence of the following articles, journals and the book for my guidance throughout the project.

My main study material through the whole semester for completing this project has been the the book on *Hands-On Machine Learning with Scikit-Learn and Tensor Flow* by *Aurélien Géron.* This book has been primary source of knowledge on using the scikit learn and learning more about the support vector machines.

Shrivastha et.al. have used different datasets to get an accurate precision from their model. They have xperimented with a number of parameters associated with the use ofthe SVM algorithm that can impact the results. These parameters include choice of kernel functions, the standard deviation of the Gaussian kernel, relative weights associated with slack variables to account for the non-uniform distribution of labeled data, and the number of training examples.

Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin shows a guide to support vector classification. Usually beginners who are not familiar with SVM often get unsatisfactory results since they miss significant steps. This guide shows procedure which are simple and give reasonable results. It also discusses the use of LIBSVM and LIBLINEAR and ends with the end result that LIBLINEAR is more efficient than LIBSVM as it consumes less memory as spaces are allocated to store recently used kernel elements [3].

Bissan et.al., have applied SVMs by input for classifying medical dataset and predicting the diseases from those data. On their paper High Dimensional Data

Classification and Feature Selection Using Support Vector Machines, they have used SVM along with various kernels to solidify their results. They have used SVM to classify the cancer based on gene expression. This paper was helpful for me to understand the working of SVM model and its kernel.

3. SUPPORT VECTOR MACHINE

3.1 OVERVIEW

SVM is a linear classifier that learns a function from the feature space examples. In order to make a good classification, it is needed to constitute a good decision surface. A special property of SVM is, SVM simultaneously minimize the empirical classification error and maximize the geometric margin. So SVM is called Maximum Margin Classifiers. [4] The basics of Support Vector Machine and how it works can be best understood with a simple example below in Figure 1.

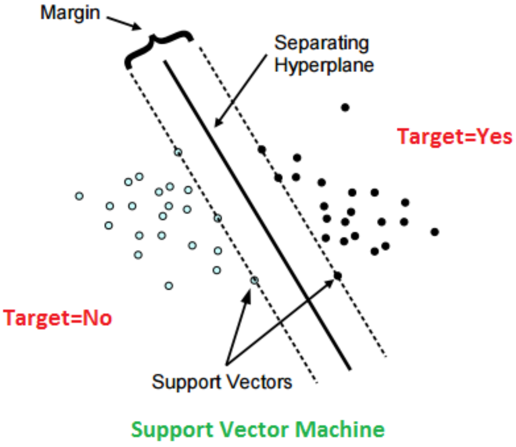


Figure 1.Classificaiton of linearly separable sample using SVM

The two classes can be separated easily after plotting them in the graph, with a straight line (they are linearly separable). The plot shows the decision boundaries of three possible linear classifiers. The solid line in the plot represents the decision boundary of an SVM classifier, this line not only separates the two classes, but also stays away from the closest training instances as possible. SVM classifier can be called as fitting the widest possible street (represented by the parallel dashed lines) between the classes, called large margin classification. [5]. Adding more training instances "off the street" will not affect the decision boundary at all, it is fully determined by the instances located on the edge of those line (street). These instances are called *Support Vectors*.

When all the instances are off the street and on the right side, this is called Hard Margin Classification. This type of classification works only with the linearly separable data and it is quite sensitive to the outliers. Selecting a more flexible model and limiting the margin violations can be a good way to avoid these issues.

Another way to maintain this balance in Scikit-Learn SVM classes is by using the C hyperparameter: a smaller C value leads to a wider street but more margin violations, and larger value leads to fewer margin violations but end up with a smaller margin.

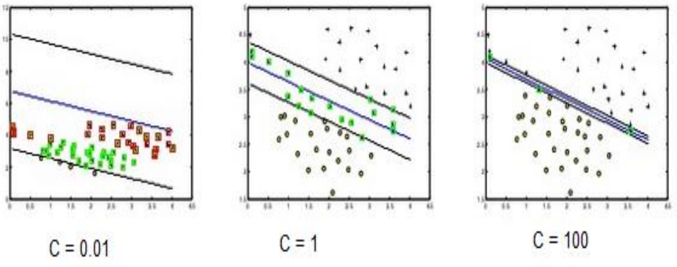


Figure 2.The effect of the using C hyperparameter. Decision boundaries (blue lines), two planes (black lines), support vectors are market in green swuares, misclassified smaples are marked with red squares.

Although linear SVM classifier are efficient, many datasets are not even close to being linearly separable. By adding more polynomial features in some cases can result in linearly separable dataset. [5]. Using a kernel trick makes it possible for linear classifier to solve a non-linear problem.

3.2 KERNEL SELECTION

Training vectors are mapped into a higher dimensional space by a function. Then SVM finds a linear separating hyper plane with the maximal margin in this higher dimension space. C > 0 is the penalty (hyperparameter) parameter of the error term. [4] Kernel function selection helps to map the nonlinear dataset into linear dataset that can help to classify the dataset. Some of the popular kernel function are:

- Linear kernel : $K (x_i , x_j) = x_i^T x_j$
- Polynomial kernel
  $K (x_i , x_j) = (\gamma x_i^T x_j + r)^d$, $\gamma > 0$
- RBF kernel:
  $K (x_i , x_j) = \exp(- \gamma \| x_i - x_j \|^2)$, $\gamma > 0$
- Sigmoid kernel:
  $K (x_i , x_j) = \tanh(\gamma x_i^T x_j + r)$

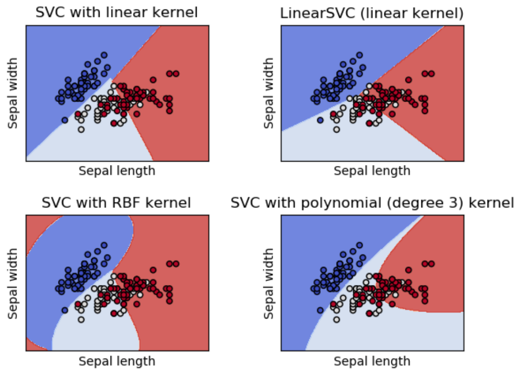


Figure: Graphical illustration of different types of kernel.

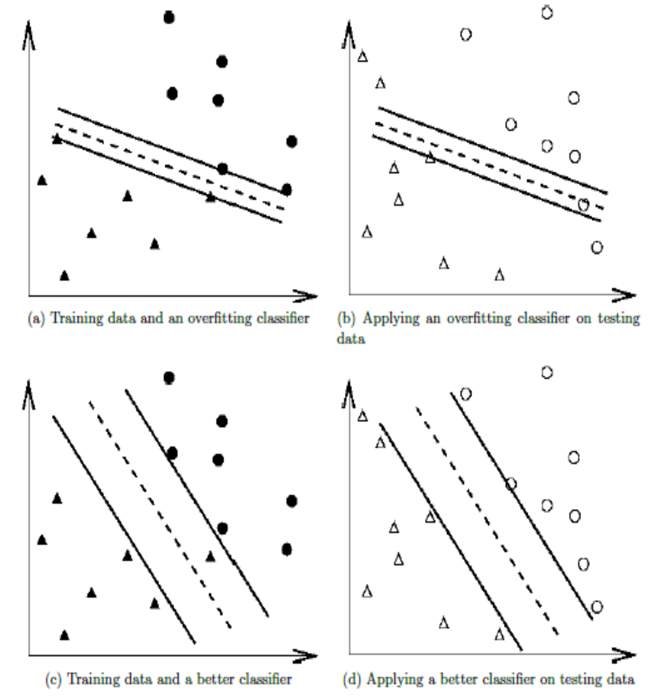
Here, x_i & x_j are training vectors and γ, r and d are kernel parameters. In these popular kernel functions, RBF is the main kernel function beacouse of the following reasons.

- The RBF kernel nonlinearly maps samples into a higher dimensional space unlike to linear kernel.
- It has less hyper parameters than the polynomial kernel and less numerical difficulties. [4]

When training an SVM with the *Radial Basis Function* (RBF) kernel, two parameters must be considered: C and gamma. The parameter C, common to all SVM kernels, trades off misclassification of training examples against simplicity of the decision surface. A low C makes the decision surface smooth, while a high C aims at classifying all training examples correctly. Gamma defines how much influence a single training example has. The larger gamma is, the closer other examples must be to be affected. [6]

3.3 MODEL SELECTION

Selection a good model for SVM is key to have a higher prediction accuracy for the SVM model. The success of the model depend on the tuning of several parameters which affect the generalization error. We often call this parameter tuning procedure as the model selection. [4] Using a grid search on C and γ is the best approach to find higher accuracy for the SVM model. Various pairs of C and γ values are tries and the one with the best cross-validation accuracy is picked.


(a) Training data and an overfitting classifier

(b) Applying an overfitting classifier on testing data

(c) Training data and a better classifier

(d) Applying a better classifier on testing data

Figure 3: An overfitting classifier and a better classifier (● and ▲: training data; ○ and △: testing data).

Though doing grid-search can be tiem consuming to find the optimal values of the parameters it is well suited approach to use for the SVC model. And the computational time required to find the good paramteres by grid-search is not much more than that by adavanced methods since there are onlu two paramteres. Many other advance methods are iteraice processes, which can be hard to parallelize. [3]

This method for selecting SVM model can work for problems with thousands or mre data points. It is also the best available method to classify the our given dataset.

4. SVC MODEL

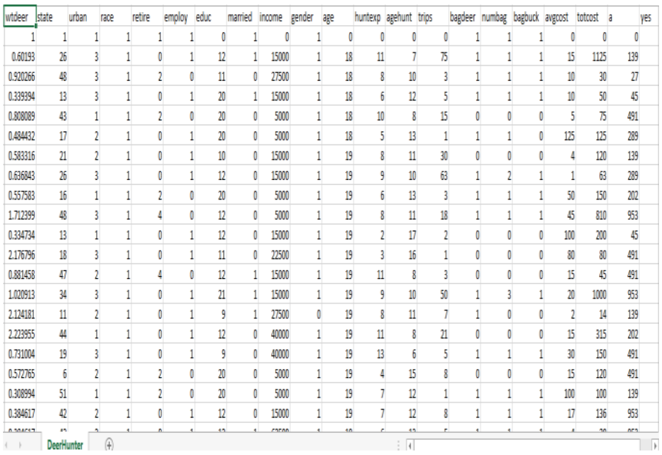
Among the different SVM models to train the dataset, we use the SVC model to train our dataset. Unlike the other classifiers that used for the linearly separable datasets, SVC classifiers takes as input two arrays: an array X of size [n_samples, n_features] holding the training samples, and an array y of class labels (strings or integers), size [n_samples]. SVC implement the “one-against-one” approach for multi-class classification. If n_class is the number of classes, then n_class * (n_class - 1) / 2 classifiers are constructed and each one trains data from two classes. To provide a consistent interface with other classifiers, the decision_function_shape option allows to aggregate the results of the “one-against-one” classifiers to a decision function of shape [6]

5. PRELIMINARY DATA

Dataset are the primary source for the execution and test of the machine learning algorithms. The use of extensive data makes the ML/AI algorithm more feasible and we can have a better understanding about the perfomance of our algorithms. For any machine learning algorithms a part of the data is used to feed the information to our model alos called as the training dataset. This dataset is used to train our model and other part of the data is used to find out the precision of our model called as testing dataset.

For the SVM model the dataset model that is to be used should always be with the numeric values. Among the given datasets the deer hunter dataset is the one that seems more relatable to feed the model as it is all numeric. The other reason to use this dataset is the sheer volume of data available that can be used to better train and test the SVM model. Since this model can be applied to other datasets as well, I worked in Breast Cancer datset imported from Scikit learn just to test the accuracy of model for a different class of data.

The DeerHunter dataset contains various information about various factors involved while hunting deer and by analyzing all of those, the program is to predict whether the person is to hunt or not. There are 20 different features in this dataset and one label.

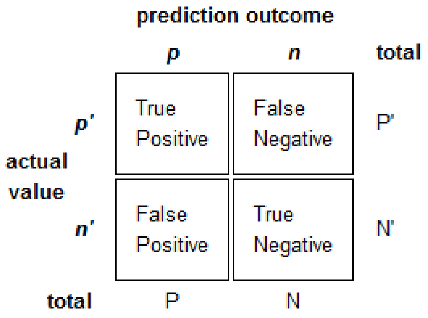


wtdeer	state	urban	race	retire	employ	educ	married	income	gender	age	huntexp	agehunt	trips	bagdeer	numbag	bagbuck	avgcost	totcost	a	yes
1	1	1	1	1	1	0	1	0	1	0	0	0	0	1	1	1	0	0	0	
0.60193	26	3	1	0	1	12	1	15000	1	18	11	7	75	1	1	1	15	1125	139	
0.920266	48	3	1	2	0	11	0	27500	1	18	8	10	3	1	1	1	10	30	27	
0.339394	13	3	1	0	1	20	1	15000	1	18	6	12	5	1	1	1	10	50	45	
0.808089	43	1	1	2	0	20	0	5000	1	18	10	8	15	0	0	0	5	75	491	
0.484432	17	2	1	0	1	20	0	5000	1	18	5	13	1	1	1	0	125	125	289	
0.583316	21	2	1	0	1	10	0	15000	1	19	8	11	30	0	0	0	4	120	139	
0.636843	26	3	1	0	1	12	0	15000	1	19	9	10	63	1	2	1	1	63	289	
0.557583	16	1	1	2	0	20	0	5000	1	19	6	13	3	1	1	1	50	150	202	
1.712399	48	3	1	4	0	12	0	5000	1	19	8	11	18	1	1	1	45	810	953	
0.334734	13	1	1	0	1	12	0	15000	1	19	2	17	2	0	0	0	100	200	45	
2.176796	18	3	1	0	1	11	0	22500	1	19	3	16	1	0	0	0	80	80	491	
0.881458	47	2	1	4	0	12	1	15000	1	19	11	8	3	0	0	0	15	45	491	
1.020913	34	3	1	0	1	21	1	15000	1	19	9	10	50	1	3	1	20	1000	953	
2.124181	11	2	1	0	1	9	1	27500	0	19	8	11	7	1	0	0	2	14	139	
2.223955	44	1	1	0	1	12	0	40000	1	19	11	8	21	0	0	0	15	315	202	
0.731004	19	3	1	0	1	9	0	40000	1	19	13	6	5	1	1	1	30	150	491	
0.572765	6	2	1	2	0	20	0	5000	1	19	4	15	8	0	0	0	15	120	491	
0.308994	51	1	1	2	0	20	0	5000	1	19	7	12	1	1	1	1	100	100	139	
0.384617	42	2	1	0	1	12	0	15000	1	19	7	12	8	1	1	1	17	136	953	

Since the SVM model works for binary datfields. Our main operation is to dimensional reduce the 20 features into one single feature and assign the other dimesional value for the single label data.

6. EVALUATION METHOD

After the data has been manipulated and run through the model, the next task is to find the model accuracy. There are different ways to evaluate the results like Confusion Matrix, Precision Call, Cross-Validation to analyze the performance of a classifier, but I have used Confusion Matrix as it can be easily evaluated and analyzed. The general idea of confusion matrix is to count the number of times instances of class A are classified as class B. [5]



		prediction outcome		
		p	n	total
actual value	p'	True Positive	False Negative	P'
	n'	False Positive	True Negative	N'
	total	P	N	

Figure 4. A confusion matrix

Here, p' is the number of instances where predicted label was True Positive and so was the actual label, n is the number of instances where the predicted was positive but actual label was negative, n' is the number of instances where the predicted label was Negative but the actual was Positive and N' is the number of instances where the predicted and the actual label is True Negative. By looking at this, we can see that p' and N' are predicted correctly but P' and n' are wrongly predicted.

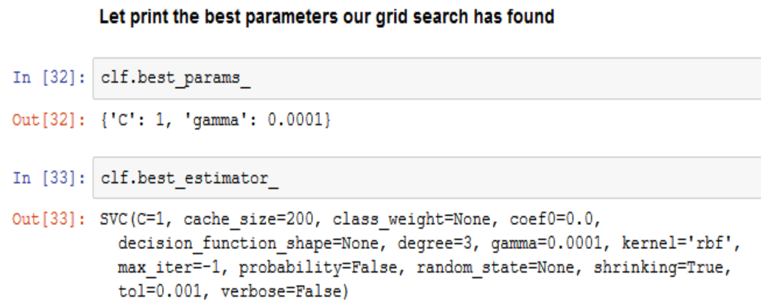
7. EXPERIMENTATION

Instead of writing the code from the scratch, I imported most of my contents from scikit learn. After importing the required from the scikit learn we need to assign certain parameters and attributes to get the optimal prediction for our datasets.

7.1 PARAMETERS

There are certain parameters that need to be set for our model to get the optimal result:
C: float, optional (value = [0.1,1,10,100,1000])
Penalty parameter C of the error term.
kernel**:** string, optional (default='rbf')
Specifies the kernel type to be used in the algorithm. It must be one of 'linear', 'poly', 'rbf', 'sigmoid', 'precomputed' or a callable. If none is given, 'rbf' will be used. If a callable is given it is used to pre-compute the kernel matrix from data matrices; that matrix should be an array of shape (n_samples, n_samples).
gamma**:** float, optional (value = [1,0.1,0.01,0.001,0.0001])
Kernel coefficient for 'rbf', 'poly' and 'sigmoid'. If gamma is 'auto' then 1/n_features will be used instead.
class_weight : {dict, 'balanced'}, optional

Set the parameter C of class i to class_weight[i]*C for SVC. If not given, all classes are supposed to have weight one. The "balanced" mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data as n_samples / (n_classes * np.bincount(y))
verbose : bool, value =3 ,
This setting takes advantage of a per-process runtime setting in libsvm.



Let print the best parameters our grid search has found

```
In [32]: clf.best_params_

Out[32]: {'C': 1, 'gamma': 0.0001}

In [33]: clf.best_estimator_

Out[33]: SVC(C=1, cache_size=200, class_weight=None, coef0=0.0,
           decision_function_shape=None, degree=3, gamma=0.0001, kernel='rbf',
           max_iter=-1, probability=False, random_state=None, shrinking=True,
           tol=0.001, verbose=False)
```

7.2 ATTRIBUTES

support_**:** array-like, shape = [n_SV] Indices of support vectors.
support_vectors_**:** array-like, shape = [n_SV, n_features] Support vectors.
n_support_**:** array-like, dtype=int32, shape = [n_class] Number of support vectors for each class.
dual_coef_**:** array, shape = [n_class-1, n_SV] Coefficients of the support vector in the decision function. For multiclass, coefficient for all 1-vs-1 classifiers. The layout of the coefficients in the multiclass case is somewhat non-trivial. See the section about multi-class classification in the SVM section

of the User Guide for details.
coef_**:** array, shape = [n_class-1, n_features] Weights assigned to the features (coefficients in the primal problem). This is only available in the case of a linear kernel.*coef_* is a read only property derived from *dual_coef_* and *support_vectors_*.
intercept_**:** array, shape = [n_class * (n_class-1) / 2] Constants in decision function.
[6]

7.3 FEATURE SCALING

Machine learning don't perform well when the input numerical attributes have very different scales. For our dataset of multiple features we need to feature scale for better input. Among the two feature scaling i.e. Standardization and min-max scaling we use the latter one. The main purpose of using minimax-scaling is to shrink the range of the feature between 0 and 1 (or -1 if there are negative values). This scaling works better for cases in which the standard scaler might not work so well. Scikit learn provides a transformer called MiniMaxScaler for this. This transformation is given by:

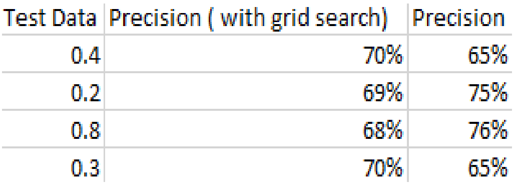
```
X_std    =    (X    -    X.min(axis=0))    /
(X.max(axis=0) - X.min(axis=0))
X_scaled = X_std * (max - min) + min
```

where min, max = feature_range. [6]

8. RESULTS

SVM is unsupervised learning so it is necessary to separate the labels at the beginning. For the dataset with multiple features, it is necessary to specify the required label. After the classification of features and labels from the dataset it is necessary to split the data into train and test data subsets. For this purpose I imported the test-train method from the scikit learn. Using this method to split the data into any percentage and shuffle it. It is necessary to shuffle the data to have different range of data for each batch of training the model. Shuffling data serves the purpose of reducing variance and making sure that models remain general and over fit less.

Initially I used 60% of the data for training purpose and 40% of the data for testing purpose. Without the use of grid search, the SVC model classified the data with an accuracy of 65% . To find the optimal values of all the features, I used grid search along with the SVC model and increased my accuracy to 70%. In order to find the effect of size of testing data, I decided to change the testing data size to 20% and got an accuracy of 75% (not using the grid search) and an accuracy of 69% using the grid search. The change in testing size does not make a significant change in results.



Test Data	Precision (with grid search)	Precision
0.4	70%	65%
0.2	69%	75%
0.8	68%	76%
0.3	70%	65%

Using the same model for the breast cancer dataset, I found the accuracy under grid search to be 95% which is pretty high for this model.

SVM model is ideal for small datasets and that might be the reason for low precision rate. Also, the high number of features in the dataset make it harder to classify it.

9. STATEMENT OF LIMITATION

SVM is ideal to use for small set of data and other advantages also include:

- Effective in high dimensional spaces and cases where number of dimensions is greater than the number of samples.
- It is also memory efficient due to the use a subset of training points in the decision function (support vectors)
- Different Kernel functions can be specified for the decision function, thus making it versatile to use.

Though SVM is effective to use in dataset classification particularly small dataset, classifying a large dataset can be time consuming for this type of model. And other disadvantages include:

- If the number of features is much greater than the number of samples, it might take a long time in training dataset.
- SVMs do not directly provide probability estimates which are calculated using an expensive five-fold cross-validation.

10. CONCLUSION

Support Vector Machine is widely used machine algorithm mainly for the classification purposes. It has been put effectively to work by combining with other machine learning algorithms. Though, SVM is an ideal model for classifying smaller datasets, it can be used to classify higher volume of data by using it with kernel function. If combined with other machine learning models, SVM can be highly efficient. Mostly SVM data classification is becoming popular classifier for medical purposes.

Though this paper was not able to get a high precision accuracy using the SVM model, it provide an insight in how SVM works, and what can be done to have higher accuracy with the models. The goal of this project was to use the SVM model to train the dataset and find the precision using the train and test data labels. Upon looking the results and completing the project goals, though a small flaw in finding higher precision this model works fine.

11. REFRENCES

[1] Do, Thanh-Nghi, and Jean-Daniel Fekete. “Large Scale Classification with Support Vector Machine Algorithms.” *Sixth International Conference on Machine Learning and Applications (ICMLA 2007)*, 2007, doi:10.1109/icmla.2007.25

[2]Ghaddar, Bissan, and Joe Naoum-Sawaya. “High Dimensional Data Classification and Feature Selection Using Support Vector Machines.” *European Journal of Operational Research*, vol. 265, no. 3, Sept. 2017, pp. 993–1004., doi:10.1016/j.ejor.2017.08.040.

[3] Hsu, Chih-Wei. *A Practical Guide to Support Vector Classification.* A Practical Guide to Support Vector Classification, 15 Apr. 2015, www.researchgate.net/publication/272039161_Evaluating_unsupervised_and_supervised_image_classification_methods_for_mapping_cotton_root_rot.

[4] Shrivastha, Durgesh K., and Lekha Bhambhu. "Data Classification Using Support Vector Machine."(2005): n. pag. Journal of Theoretical and Applied Information Technology. JATIT, 2009. Web. 20 Apr. 2018.

[5]Géron, Aurélien. *Hands-on Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. OReilly, 2017.

[6]“1.4. Support Vector Machines¶.” *1.4. Support Vector Machines - Scikit-Learn 0.19.1 Documentation*, scikit-learn.org/stable/modules/svm.html.