**Handwritten Character Recognition Using Machine Learning**

**Introduction:**

Handwriting analysis, character recognition, and graphology are all synonyms for “the study of printed and written symbols and writing systems [1]”. These topics have numerous commercial and private applications. The most well-known application of commercial handwriting analysis is documented within the criminal justice system as a subdivision of forensic science. However, private citizens can also employ handwriting analysis in day to day life, albeit on a more superficial scale. For example, students decoding a professor's handwritten lecture notes or an executive officer reading a message receipt provided by a secretary. Each of these scenarios, both commercial and private, rely on “the principle that no two individuals can produce the same handwriting and that an individual can not exactly reproduce their own handwriting, due to natural deviations otherwise known as variation. [2]”

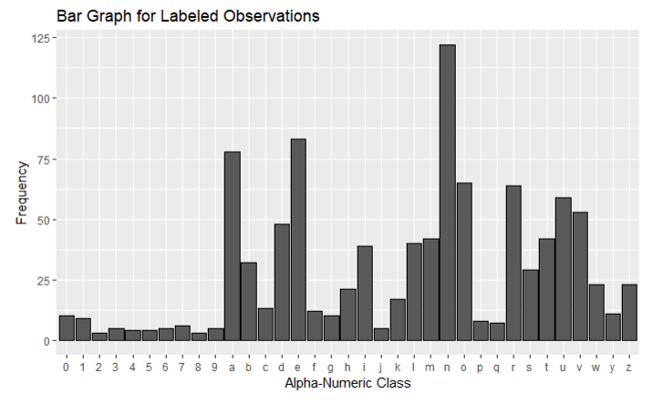
**Background:**

The purpose of this paper is “to construct a classiﬁer that will predict the letter of a handwritten character based on a 56 by 56 bitmap [3]”. To do so, two data sets are given, defined as “Labeled” and “Unlabeled”.

The Labeled data set contains 10,440 rows and 3,138 columns. The first column is the observation number, which will be ignored throughout the paper. This is because each observation is contained within a single row and isn’t a meaningful predictor. The second column is the character observation, denoted “Letter”. Characters are defined to be single digit alpha numeric. The remaining 3,136 columns represent individual pixels in a 56x56 bitmap. The column values are binary, denoting “1” for shaded pixels, and “0” for non-shaded or empty pixels.

The Unlabeled data set contains 2000 rows and 3,137 columns. The first column is the observation number, which is also ignored for the same reason stated above. The main difference between the Labeled and Unlabeled data is that, as the name suggests, there is no character observation column for each observation in the Unlabeled data set. Instead, there are only the remaining 3,136 columns to represent the individual shaded or non-shaded pixels.

After reviewing the Labeled observations, it was found that of the 26 alphabetic characters assumed to be included in the data set only 25 were present, since the character “X” was omitted. 10 numeric characters were included in the Labeled data set (0-9). *Figure 1* below shows the approximate distribution of the Labeled observations.



*Figure 1: Bar Graph of the Alpha-Numeric Classes*

**Principal Component Analysis:**

As discussed earlier, there are 3,136 pixels in the 56x56 grid of an image under study resulting in 3,136 variables in the dataset. If all the variables are to be used in building the model, it can be problematic in the sense that it would attain too much computational complexity. The other disadvantage that comes along with high dimensional data is that there would be too many pairwise correlations between the variables to consider. Principal Component Analysis can be useful in such scenario as it is a technique for feature extraction and eventually, dimensionality reduction. It combines the input variables in a specific way to create new features, after which the least important features can be dropped while still retaining the most valuable information in the data. The added benefit is that the new features generated are all orthogonal or totally uncorrelated to each other.

PCA is an unsupervised learning method in that it finds patterns without reference to the prior knowledge about the dataset. It reduces data by geometrically projecting them onto lower dimensions called principal components (PCs), with the goal of finding the best summary of the data using a limited number of PCs. This transformation process is defined in such a way that the first principal component has the largest possible variance, and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. For more information on detailed workings on Principal Component Analysis, please refer to Principal Component Analysis, Series [4].

For the given dataset, principal component analysis is performed, and the loadings are extracted. On observing the variance explained by each principal component, it is seen that the first principal component explains 6.67% variance within the dataset while the second principal component explains 5.34% variance within the dataset, and it goes on decreasing for the remaining principal components. 80 principal components explained 70% of the variance within the data which is why 80 principal components were chosen among the 3,136 principal components generated. It is interesting to see that the first 1000 principal components explain almost 96% variation within the data which means that 2136 principal components could be dropped without losing any significant amount of information in the data. Considering the time constraint and the computational limitation for this project, 1000 principal components is too many to consider. So, being aware about the fact that the information on 30% variance within the data will be lost, 80 principal components are taken into consideration for further analysis of this project because with this chance of losing data also comes the benefit of greatly reducing the computational complexity.

Once the principal component analysis is performed in the training dataset, the loadings for the training dataset, the test dataset, and the validation dataset are generated. Now, these features are ready to be fed into the model or to be predicted from the model, as per the requirement.

**Support Vector Machines:**

Support vector machines are a class of supervised learning methods that can be used for classification of both linear and non-linear data. As noted above, the data contains 35 different classes of characters, ranging from both letters and numbers. SVM’s have been successfully applied to several applications ranging from face detection [5], object recognition [6] and even detection of moving vehicles [7]. It was decided to use SVM to classify the predictions of letters and numbers based on SVM’s ability and prior reputation to be able to distinguish more complex data.

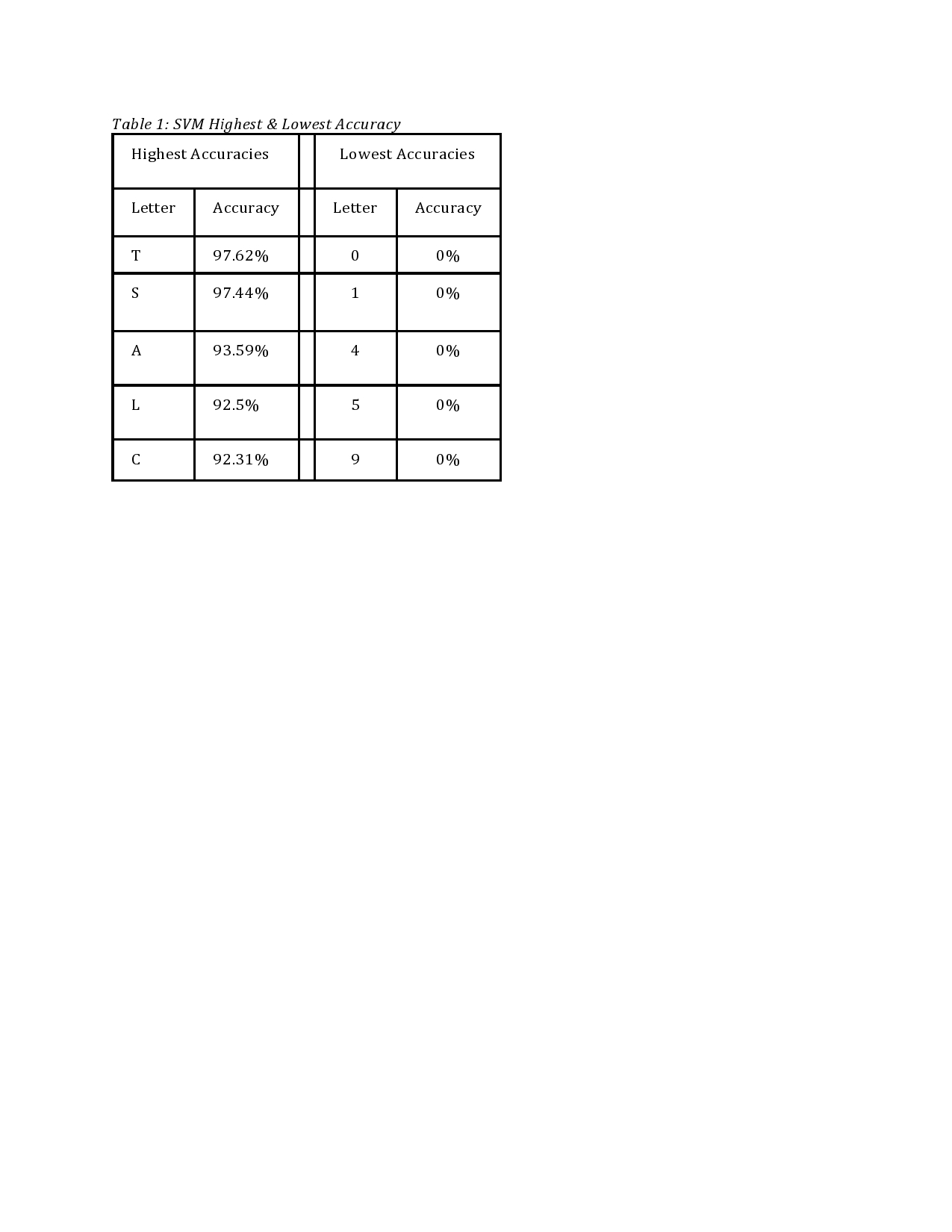
The basic idea of the SVMs is to construct a line, plane or “hyperplane” as the decision plane, which separates classes with the largest margin. An optimal hyperplane is a hyperplane with the maximum margin of separation between the two classes, where the margin is the sum of the distances from the hyperplane to the closest data points of each of the two classes. These closest data points are referred to as Support Vectors.

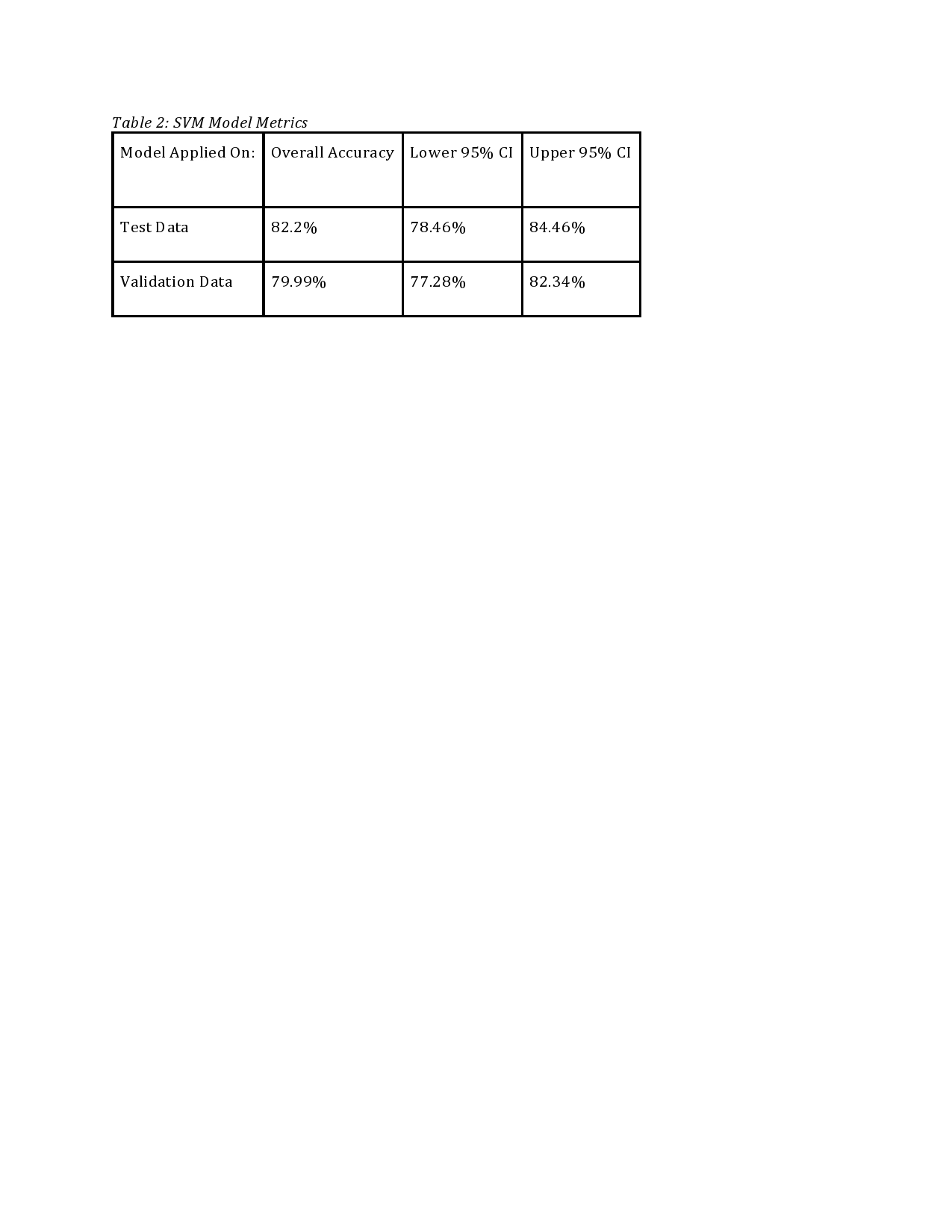
Although this may be impractical for implications on real-scenario data, sometimes two classes are not linearly completely separable. However, a hyperplane that maximizes the margin while minimizing a quantity proportional to the misclassification errors can still be achieved by introducing positive slack variables. For more information on detailed workings on Support Vector Machines, please refer to SVM Classification in Multiclass Letter Recognition [8].

After conducting a train/split ratio above, SVM was built using the package caret; and tuning function was used to find the optimum number of slack variables(C) to reduce the error. It is also noted that Caret’s uses a One Versus One ensemble. The OVO ensemble would be composed of 595 binary classifiers. The first will classifier would distinguish A from B, the second A from C, and the third A from D. etc. If another letter is to be classified, that letter would be presented to each binary classifier of the ensemble to create a vector of individual classifications. Based on majority voting, that letter would be assigned.

Also 10-Fold cross-validation was used to test the robustness of the model. The model averaged about 80% accuracy while attempting to predict the test set using a default value of C =1. C was the parameter chosen by us that controlled the tradeoff between the margin and the misclassification errors. A higher C means that a higher penalty to misclassification errors is assigned whereas, for tiny values of C, we should get misclassified examples. It was decided to use a smaller C with a value of 0.05, as there were letters that were bound to be misclassified regardless due to their similarity in writing structure. For example, I's and 1's, 5's and S's and O's and 0's. Choosing a small value of C would more than likely maximize the linear separation between classes that are different such as A's and F's, A and O's, etc.

The highest accuracy is achieved at C = 0.05 before the values tend to decrease as the value of C is higher. The model averaged about 82.2% accuracy while attempting to predict the test set after tuning the value of C. The model was then used to predict cases on the original 1000 letters. For example, the model was able to perform well predicting A's, C’s and E's but had more unsatisfactory performance in differentiating between 1's and I's, D's and O's & O's and 0's. Table 1 is a summary of the top 5 lowest and highest accuracies. As we can see from these tables, although the model does extremely well in classifying variables, it fails to differentiate similar letters to numbers. The overall accuracy on the original labeled dataset is around 79.9% having a 95% confidence interval at around 77.28-82.34%. An overall summary on the accuracy metrics can be seen in Table 2.

**



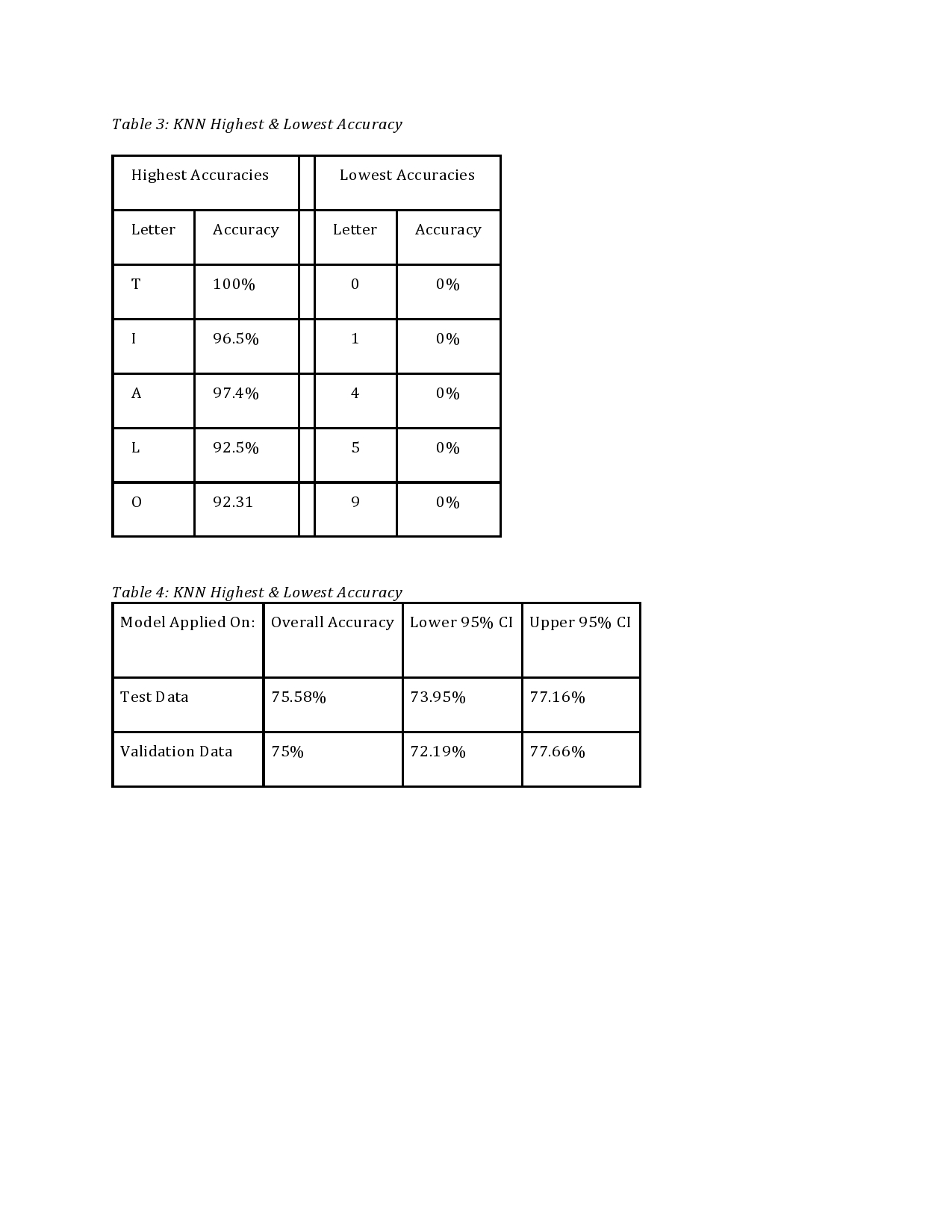
**k-Nearest-Neighbors (KNN):**

The k-Nearest-Neighbors (KNN) is a non-parametric classification method. By non-parametric it means that the KNN classification method does not assume a distribution of the data. For a specific object to be classified, its “k” nearest neighbors are retrieved and based on majority voting amongst the data in the neighborhood family; the classification of the specific object is decided. Each object is classified based on an independent trial. For this paper, Euclidean distance-based weighting for identifying the nearest k- neighbors was used. In order to apply the KNN method an appropriate value for k should be chosen, and the success of classification is largely dependent on this value. This is because results may change depending on how many nearest neighbors are being classified to the respective class.

KNN is often used in text mining [9], finance and agriculture [10]. The decision to use KNN arose from the assumption that a single class of characters should not have a large variance across each observation. Therefore, KNN may be an ideal model for classifying objects properly into their respective character class. For more information on the KNN process, please refer to Distance Metric Learning for Large Margin Nearest Neighbor Classification [11].

The process for choosing the best “k” value, was to run the algorithm on a 10-Fold cross-validation with different k values and choose the one with the best performance. The range of values tested on KNN spanned from 3-11. As it can be seen, the minimum error rate occurs at k=5, and that is the value of K chosen. The performance of the test set yielded about 75.44%. When the model was used to predict the accuracy of the original set of variables (validation set), the accuracy rate was about 75.8% with a confidence interval between 73.02-78.43%. *Table 3* shows us how well the letters were predicted individually using the K-NN algorithm.

The overall accuracy on the validation test with KNN is about 75.8% with an upper confidence interval at around 78.4% and a lower confidence interval at 73%. *Tables 3 and 4* provide a summary of this information.



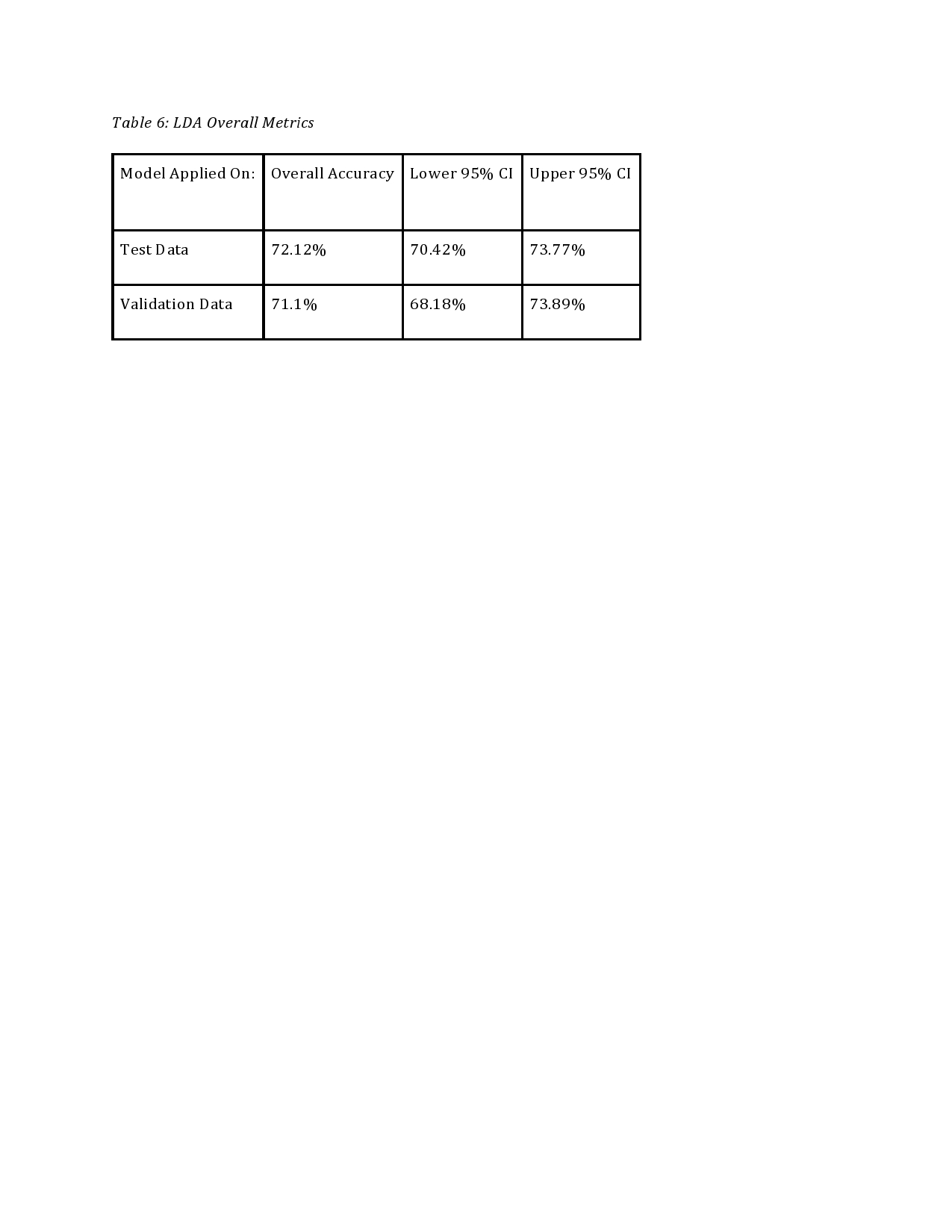
**Linear Discriminant Analysis:**

Linear discriminant analysis (LDA) “is a well-established machine learning technique for predicting categories. Its main advantages, compared to other classification algorithms such as neural networks and random forests, are that the model is interpretable, and that prediction is easy [12]”. This is partially why LDA was chosen as a applicable model for the project, because it is quickly run and easily implemented.

However, the technical goal of LDA is to take a single observation and to predict the likelihood of the observation belonging to a single class, versus any other set of classes. The class that gets the highest probability is the class that is assigned or in this case “predicted” for the observation. Since the goal of this paper is to correctly classify or “predict” an observation into 1 of 36 bins (36 bins would be assuming X’s are introduced in the 30,000 observations later released), it would appear that LDA is not only easily applicable, but also well suited to the task of classifying the observations as one single character over another character.

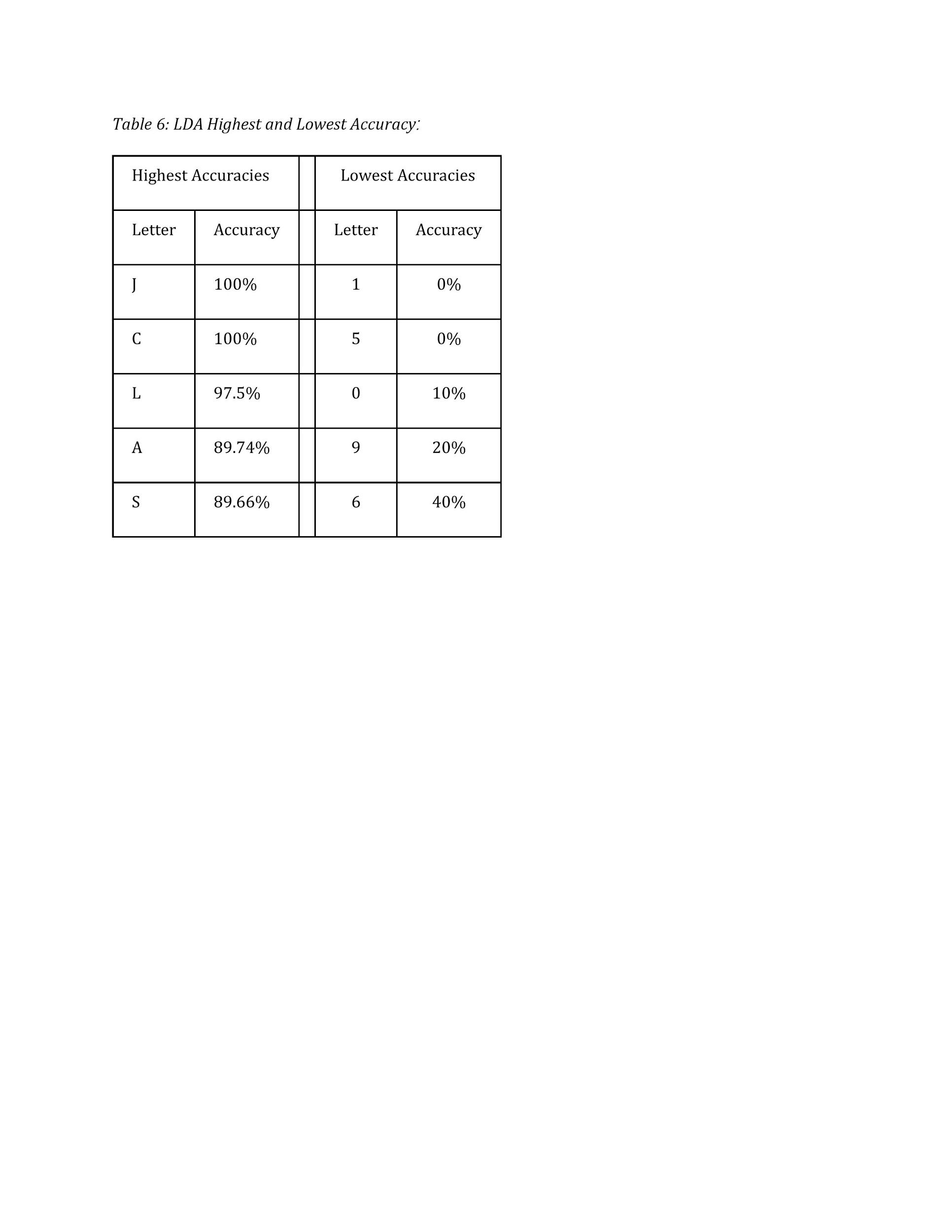
There are some important assumptions and limitations to using LDA. The first and potentially most relevant assumption is that the data is independent and normally distributed [13]. Since the data is binary, this assumption is violated. However, this does not invalidate the accuracy of the model since the data has been transformed from the original predictor variables to principal components. Additionally, there is the limitation of multicollinearity, which states that “predictive power can decrease with an increased correlation between predictor variables” [13]. Since PCA was previously performed to enhance the variable selection, multicollinearity is essentially a non-issue. Lastly, an LDAs potential to overfit depends largely on the number of predictor variables relative to the number of samples. In this case, since there are 80 principal components and approximately 10,000 samples, the model does not appear prone to overfitting.

The overall model performance is 72.12% accuracy as seen in *Table 6* below.

**

However, the accuracy rate does vary significantly by character class. The highest class performance is for characters: J, C,L,A,S. Each of these classes has an accuracy rate at or above approximately 90%. It is worth noting that these characters tend to appear more frequently in the data distributions defined previously in the paper. The worst class performance is for characters: 1,5,0,9,6. The digit 6 has the highest accuracy rate at approximately 40%. The remaining digits (1,5,0,9) fall significantly below the 40% mark, ranging to the minimum accuracy of 0%. It should be noted that each of the worst performing characters are noticeably digits, so the total count of observations is low to begin with and each subsequent misclassification ultimately represents a large portion of the overall accuracy rate. Each of these character class accuracy rates can be seen in *Table 7 below.*

*Table 7: LDA Highest and Lowest Accuracy*



**Conclusion:**

SVM is a popular classification method and has been used in different areas, such as classifying a letter of handwritten alphanumeric character into predefined categories. Using an optimum number of components determined after applying the PCA technique, and also choosing the best candidate kernel for building the model will reduce the computation time. In addition, better results are generated as an effect of using hand labeling the unlabeled data. The results show that the proposed method has an accuracy rate of 79.99%.

Comparing the results of the chosen model (SVM) with the other candidate models; KNN, and LDA shows that SVM has a higher accuracy rate for the same input. When comparing SVM (with accuracy rate of 79.99%) with KNN (with accuracy rate of 75.64%), highest accuracy of classes in both SVM and KNN are for alphabetic characters and their lowest accuracy for the numbers, but overall accuracy of the trained SVM model is higher than trained KNN model [14]. When comparing SVM with LDA, the results show that SVM as a higher accuracy compared to LDA (with accuracy rate of 72.12%). Highest accuracy of classes in both SVM and LDA are for alphabetic characters, and LDA has better accuracy rate for classifying numbers, but overall accuracy rate of the trained SVM is higher than LDA [15]. Thus, the it can be concluded that SVM has a higher accuracy for classifying the handwritten alphanumeric characters and thus is a best candidate among the investigated models.

**Reference:**

[1]: “Graphology”. Definition 3 (linguistics). <https://en.oxforddictionaries.com/>

This source was used to provide a formal definition of handwriting analysis and relevant applications.

[2]: <https://www.robsonforensic.com/articles/forensic-handwriting-analysis-expert>

This source was used to determine the assumptions of handwriting analysis.

[3]: D2L - Letter recognition.pdf

This was the original source given with the problem statement.

[4]: Jolliffe I.T. Principal Component Analysis, Series: Springer Series in Statistics, 2nd ed., Springer, NY, 2002, XXIX, 487 p. 28 illus. [ISBN](https://en.wikipedia.org/wiki/International_Standard_Book_Number) [978-0-387-95442-4](https://en.wikipedia.org/wiki/Special:BookSources/978-0-387-95442-4)

[5]: K. Jonsson, J. Kittler, and Y.P. Matas, Support vector machines for face authentication, Journal of Image and Vision Computing, vol. 20. pp. 369-375, (2002).

[6]: F. Seraldi and J. Bigun, Retinal vision applied to facial features detection and face authentication, Pattern Recognition Letters, vol. 23, pp. 463-475, (2002).

[7]: D. Gao, J. Zhou, and Leping Xin, SVM-based detection of moving vehicles for automatic traffic monitoring, IEEE Intelligent Transportation System, pp. 745- 749, (2001).

[8]: Kaushik, Aarti, et al. *SVM Classification in Multiclass Letter Recognition System*. Global Journal of Computer Science and Technology Software & Data Engineering (2013).

[9]: Baoli, L., Shiwen, Y. & Qin, L. (2003) "An Improved k-Nearest Neighbor Algorithm for Text Categorization, ArXiv Computer Science e-prints.

[10]: Bauer, M.E., Burk, T.E., Ek, A.R., Coppin, P.R. Lime, S.D., Walsh, T.A., Walters, D.K., Befort, W. & Heinzen, D.F. (1994) "Satellite Inventory of Minnesota’s Forest Resources", Photogrammetric Engineering and Remote Sensing, 60(3): 287–298.

[11]: K.Q. Weinberger and L.K. Saul. Distance metric learning for large margin nearest neighbor classification. Journal of Machine Learning Research, 10(Feb):207–244, 2009.

[12]: <https://www.r-bloggers.com/linear-discriminant-analysis-in-r-an-introduction/>

This source was used to define the pros of the LDA model.

[13]: <https://en.wikipedia.org/wiki/Linear_discriminant_analysis>

This source was used to define the assumptions of the LDA model.

[14]: <https://www.ijraset.com/fileserve.php?FID=11852>

[15] <https://stats.stackexchange.com/questions/243932/what-is-the-difference-between-svm-and-lda>