2.2 US CENSUS

1. (4 points) For each of the five algorithms list key strength and key weakness. Use no more than 250 words in total (+- 50 per algorithm).

Naïve Bayes:

When assumption of independent predictors holds true, a Naive Bayes classifier performs better as compared to other models. Naive Bayes requires a small amount of training data to estimate the test data and so is also easy to implement.

Main imitation of Naive Bayes is the assumption of independent predictors. Naive Bayes implicitly assumes that all the attributes are mutually independent. It is almost impossible that we get a set of predictors which are completely independent.

Decision Trees:

Decision Trees require less effort for data preparation. Moreover, it does not require normalization of data as well as scaling of data.

The disadvantages include more time required to train the model. Also, in decision trees a small change in the data can cause a large change in the structure of the decision tree causing instability.

K-Nearest Neighbor Classifier:

K-NN is a non-parametric algorithm which means there are assumptions to be met to implement K-NN. It does not explicitly build any model and simply tags the new data entry-based learning from historical data.

K-NN might be very easy to implement but as dataset grows efficiency or speed of algorithm declines very fast. K-NN algorithm is very sensitive to outliers as it simply chose the neighbors based on distance criteria.

SVM:

SVM works well when there is a clear margin of separation between the classes. It is more effective in higher dimensions and is relatively more memory efficient.

SVM is not suitable for large data sets and does not perform well when the data set has more noise.

Logistic Regression:

Logistic Regression performs well when the dataset is linearly separable. It is easier to implement, interpret and very efficient to train.

A disadvantage of it is that we can't solve non-linear problems with logistic regression since its decision surface is linear. It also requires a large dataset and also sufficient training examples for all the categories it needs to identify.

2. (3 points) Carefully read the Scikit-learn hyper-parameter documentation for each of the five algorithms. Based on this documentation explain how the previously mentioned hyper-parameters effect the algorithms and their performance. Express yourself clearly and provide your reasoning. Use no more than 300 words in total (+- 75 per algorithm). Note: You don't have to write anything about the Naive Bayes since it has not hyperparameters of interest.

Decision Trees:

Max-depth: The maximum depth of the tree. The deeper you allow your tree to grow, the more complex your model will become because you will have more splits and it captures more information about the data. Setting the max depth prevents overfitting.

Min-samples-leaf: The min number of samples required to be at a leaf node. It ensures that the tree cannot overfit the training dataset by creating a bunch of small branches exclusively for one sample each, which are always pure.

Random state: Controls the randomness of the estimator. The features are always randomly permuted at each split. Ensures that the results are not too different if the decision tree algorithm is rerun.

K-Nearest Neighbor Classifier:

N-neighbors: Number of neighbors to use around a point. If k is too small, the algorithm would be more sensitive to outliers. If k is too large, then the neighborhood may include too many points from other classes.

Weights: weight function used in prediction. You could give more weight to the points which are nearby and less weight to the points which are farther away. This can be more robust against variations in distances of the k-nearest neighbors which may lead to wrong decision.

SVC:

C: Large values of C mean low regularization which in turn causes the training data to fit very well but may cause over fitting. Lower values of C mean higher regularization which causes the model to be more tolerant of errors which can cause lower accuracy

Kernel: Specifies the kernel type to be used in the algorithm. The kernel, is selected based on the type of data and also the type of transformation

Random state: Controls the randomness of the estimator.

Logistic Regression:

C: Large values of C mean low regularization which in turn causes the training data to fit very well but may cause over fitting. Lower values of C mean higher regularization which causes the model to be more tolerant of errors which can cause lower accuracy

Penalty: Used to specify the norm used in penalization

random state: Controls the randomness of the estimator.

1. (1 point) Explore the features and target variables of the dataset. What is the right performance metric to use for this dataset? Clearly explain which performance metric you choose and why. Use no more than 125 words.

The training data on observation shows that there is not the same number of observations for the two classes. The number of observations for a person with a salary less than $50,000 are much more in number as compare to the number of observations for a person with a salary of greater than $50,000. Using classification accuracy as a performance metric is not a good choice for such a skewed data. We make a choice of using F1 score as a performance metric for this reason.

2. (1 point) Algorithmic bias can be a real problem in Machine Learning. So based on this, should we use the Race and the Sex features in our machine learning algorithm? Clearly explain what you believe, also provide us with arguments why. Note this question will be graded based only on your argumentation. Use no more than 75 words.

Observing the salaries shows that gender and sex does in-fact play a part in determining whether a person has a salary above $50,000 or not. As we are training the classifier to predict the salary of a particular person, the race and gender of that person is an important feature for the prediction and should be used. If we were however training a classifier to assign a salary to a person given his attributes, we would not use the gender and race attributes.

1. (2 points) This dataset hasn't been cleaned, yet. Do this by finding all the missing values and handling them. How did you handle these missing values? Clearly explain which values were missing and how you handled them. Use no more than 100 words.

We used Imputer functionality provided by sklearn library to Identify the missing values in every column. The missing values in a column were replaced by the most frequent value in that particular column. The column's with missing values were education-num, workclass, occupation and native-country.

2. (2 points) All Scikit-learn's implementations of these algorithms expect numerical features. Check for all features if they are in numerical format. If not, transform these features into numerical ones. Clearly explain which features you transformed, how you transformed them and why these transformations. Use no more than 75 words. (You might want to read the preprocessing documentation of Scikit-learn for handy tips.)

We converted the native-country attribute to have two values (United-States and Outside US). This was done as X_test contains native Country values which are not seen in the native-country attribute of X_train. Then transformed the non-numerical data into numerical data. The features we encoded were workclass, education, marital-status, occupation, relationship, race, sex, native-country. The data was transformed as sklearn expects numerical values.

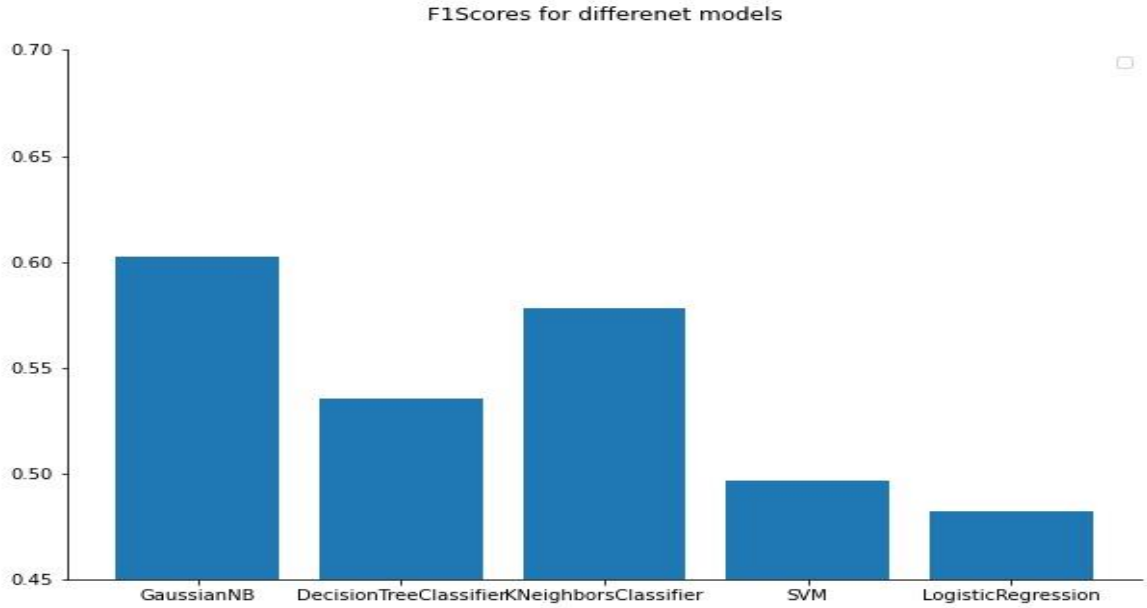
3. (Bonus 2 point) Have you done any other data preprocessing steps? If you did, explain what you did and why you did it. Use no more than 100 words.

We converted the native-country attribute to have two values (United-States and Outside US). This was done as X_test contains native Country values which are not seen in the native-country attribute of X_train.

1. (1 point) Now set up your experiment. Clearly explain how you divided the data and how you ensured that your measurements are valid. Use no more than 100 words.

Data was divided into 2 sets, the training set and the validation set. The training-validation ratio was kept at 80:20. The data was shuffled before it was divided into training and test sets. The data was split in a stratified fashion using Y_train as the class labels. Stratification which will lock the distribution of classes in train and test sets.

2. (2 points) Fit the five algorithms using the default hyper-parameters from section 2.1. Create a useful plot that shows the performances of the algorithms. Clearly explain what this plot tells us about the performances of the algorithms. Also, clearly explain why you think some algorithms perform better than others. Use no more than 150 words and two plots (but 1 is sufficient).

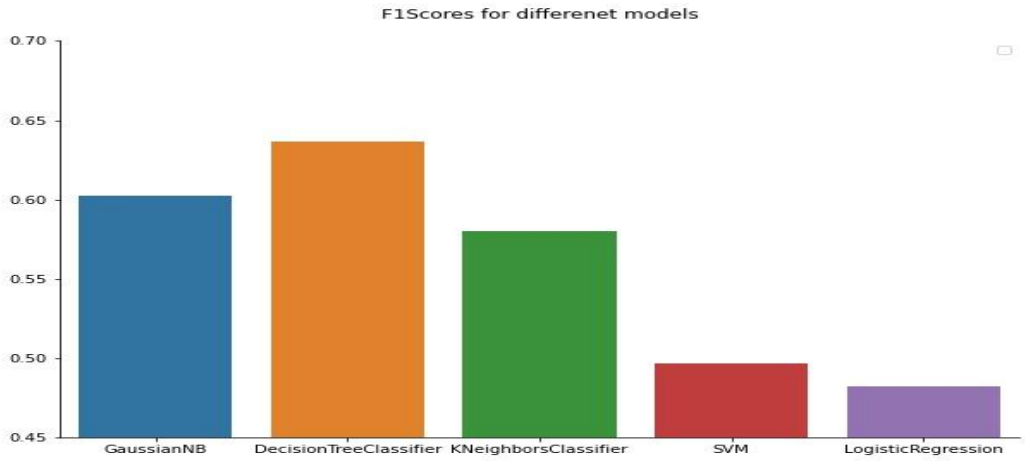


The F1 score plotted as a bar graph shows that Gaussian Non-Bayes classifier works the best followed by K-Neighbors classifier, Decision-Tree classifier, SVM while the worst classifier is the Logistic Regression classifier. The skewed data could be the reason of some algorithms performing better than others. The general performance of all the algorithms is low however.

3. (2 points) Now perform hyper-parameter tuning on the key hyper-parameters you have previously identified. Clearly explain what you did to be systematic, what you did to get fair results, what trade-off accuracy vs resources trade-off, etc. Use no more than 200 words. Note: First focus on tuning the default hyper-parameters, this should be sufficient. Only look at others if time permits it.

We used GridSearchCv for every model. Each of the default hyper-parameters of the model was given a list of values. GridSearchCV performs the classification using combinations for all these values of hyperparameters and returns the hyperparameter combination which gives the greatest F-score. Using a lot of values for each hyper parameter leads to a lot of combinations which in turn leads to greater computation. We balanced the number of hyper parameters in such a way that we get somewhat of an accurate result, while keeping the computation needed for all the combinations of hyperparameters reasonable as well. This was done by only using hyper parameter values which are known to perform well.

4. (2 points) Compare the performance of the algorithms with and without hyper-parameter tuning. How did the tuning affect your result? Clearly explain the results and the differences. Use no more than 100 words and two plots (but 1 is sufficient).



The graph above shows that the performance has increase for Decision Tree classifier while it has remained the same for the others. This is because GridSearchCV takes the combination of hyper-parameters which are the most efficient and give the best result for the data.

5. Select your best algorithm for this dataset and use it to make your predictions for the unknown samples. Please note in you algorithm which algorithm you chose.

The best algorithm for this data-set is Decision Tree classifier as it gives the best F-1 score for the validation set.

2.3 MNIST

2.3.1 DATA EXPLORATION

1. (1 point) Explore the dataset by plotting the same image from both datasets side by side. How do these images compare? Which dataset do you expect to perform better? Clearly explain why you suspect that. Use no more than 75 words.

Looking at the images from both data sets it seems like the data-set storing 8x8 images will perform better. The digits pixels in the image are stored in the center of the image, which is why there are many redundant pixel values for 28x28 images.8x8 images gets More pixels mean more larger dimensionality of the feature space. Which means more data will be required for proper classification (curse of dimensionality).

3.2 DATA PREPARATIONS

1. (3 points) Examine the features of both the datasets and decide if you need to do any data cleaning or preprocessing. If not, clearly explain why not. If yes, clearly explain why and what you did. Use no more than 100 words. (You might want to read the additional reading materials).

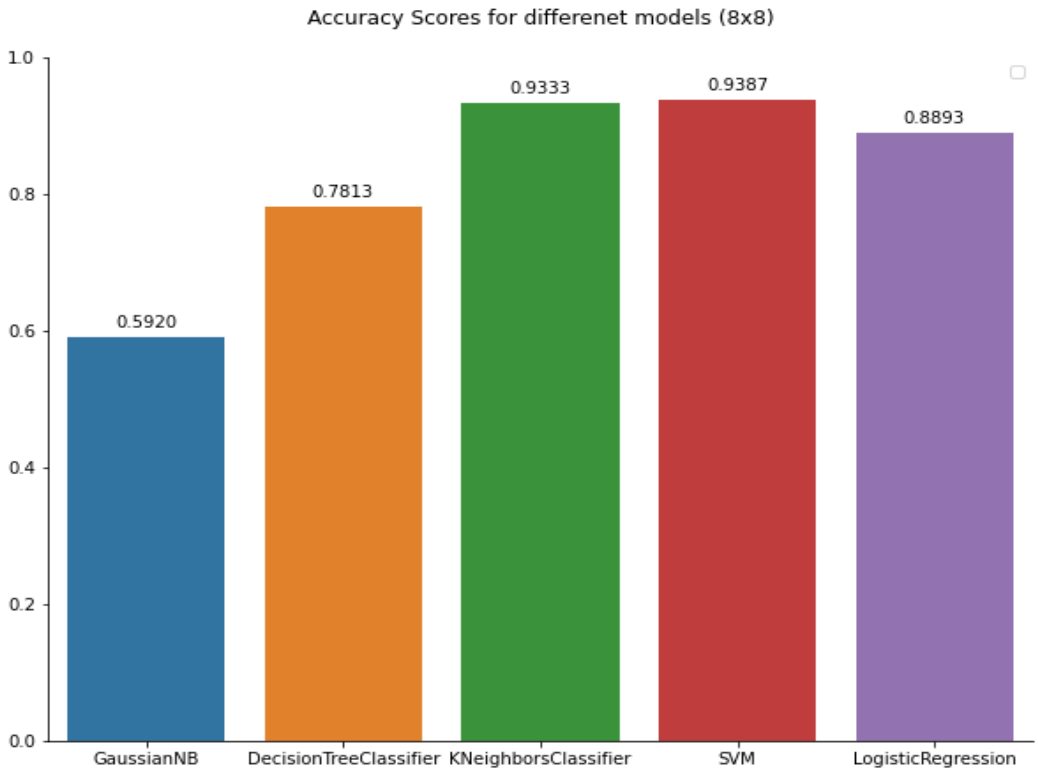
We processed the data before using it. The values of each pixel were normalized such that a pixel had a value between 0 and 1 and not between 0 and 255. Every pixel value was divided by 255 for this reason. The reason for normalizing the pixel values was to reduce the variance of the data. We also reshaped the image data array from 2 dimension to 1 dimension as the functions expect the data array to be 1 dimensional.

2.3.3 EXPERIMENTS

1. (1 point) Now set up your experiment. Clearly explain how you divided the data and how you ensured a valid measurement. Use no more than 100 words.

Data was divided into 2 sets, the training set and the validation set. The training-validation ratio was kept at 80:20. The data was shuffled before it was divided into training and test sets. The data was split in a stratified fashion using Y_train as the class labels. Stratification which will lock the distribution of classes in train and test sets.

2. (2 points) Fit the five algorithms using Scikit-learn's default hyper-parameters. Create a useful plot that shows the performances of the algorithms. Clearly explain what these plots tell us about the performances of the algorithms. Also, clearly explain why you think some algorithms perform better than others and why some of them perform better on one dataset than the other. Use no more than 200 words and two plots (but 1 is sufficient).



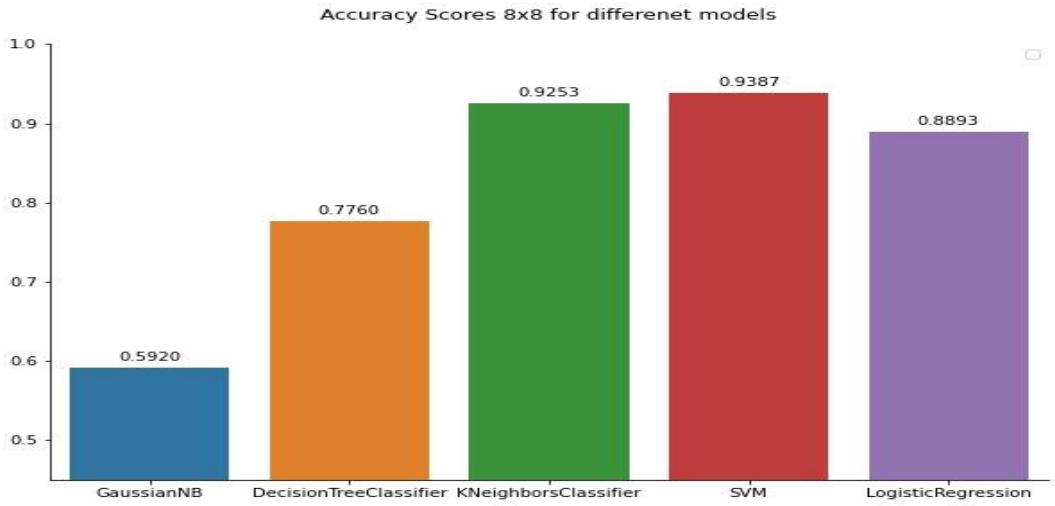
The bar graph shows that SVM performs the best out of all the algorithms while GaussianNB performs the worst. Naive Bayes implicitly assumes that all the attributes are mutually independent, so the reason for it underperforming could be that that the attributes are not independent. The type of data determines which algorithm performs better. If the data is linearly separable then Linear classifiers like Logistic Regression will flourish while if the data is non-linearly separable then non-linear classifiers like KNN perform better.

3. (2 points) Now perform hyper-parameter tuning on the key hyper-parameters you have previously identified. Clearly explain what you did and how you did this. Use no more than 200 words.

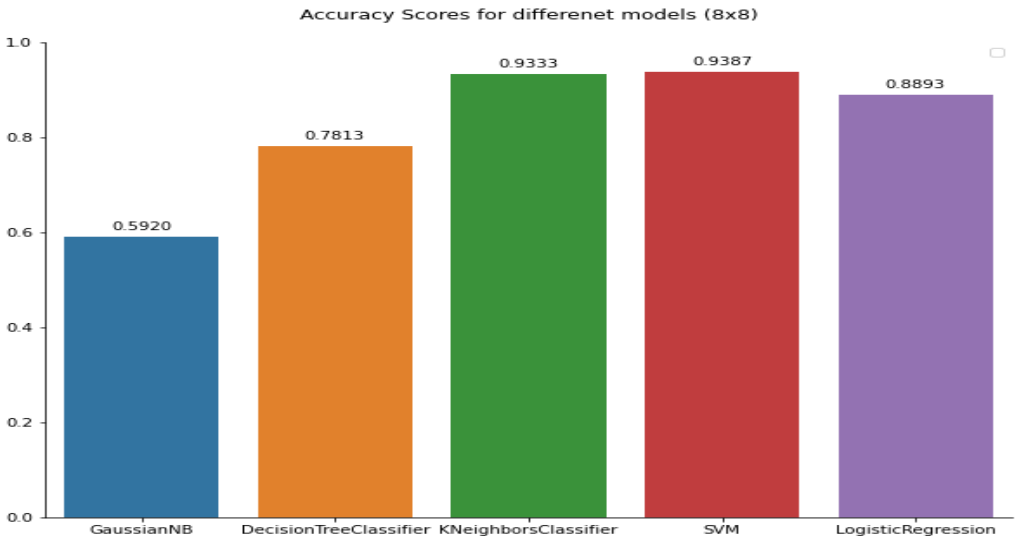
We used GridSearchCv for every model using both data sets. Each of the default hyper-parameters of the model was given a list of values. GridSearchCV performs the classification using combinations for all these values of hyperparameters and returns the hyperparameter combination which gives the greatest accuracy-score. Using a lot of values for each hyper parameter leads to a lot of combinations which in turn leads to greater computation. We balanced the number of hyper parameters in such a way that we get somewhat of an accurate result, while keeping the computation needed for all the combinations of hyperparameters reasonable as well. This was done by only using hyper parameter values which are known to perform well.

4. (2 points) Compare the performance of the algorithms with and without hyper-parameter tuning. Also, make a comparison with your original baseline. How did the tuning affect your result? Clearly explain the results and the differences. Use no more than 200 words and two plots (but 1 is sufficient).

After Tuning:



Before Tuning:



The performances of the algorithms remain almost the same after hyper parameter tuning. The performance of Decision Tree increases while that of KNN decreases slightly. These differences are minimal however. The performances the other classifiers remain the same and do not change. These minimal changes may be because we did not try out all the combinations of hyper-parameters.

5. (1 points) Compare the performance of the algorithms with the 8x8 and 28x28 features. What effect do the additional features have? Also, state what you think causes this effect. Use no more than 75 words.

The performance of the algorithms for both the data sets is almost similar with the performance for 8x8 data set a bit better for the majority of the algorithms

6. Select your best algorithm for this dataset and use it to make your predictions for the unknown samples. Feel free to use either the 8x8 or 28x28 features. Please note in your report which algorithm and feature set you chose.

The best algorithm we chose for this data set is SVC. We used the 8x8 feature set.

2.4 CONCLUSION

1. (3 points) Which conclusions can we draw about the five algorithms examined during this assignment? For each algorithm briefly discuss the key thing you noticed about it during this assignment. Use no more than 250 words in total (+- 50 words per algorithm).

We noticed that the type of data distribution determines how well a classifier works. Some classifiers would for example flourish if the data was easily separable while others work better when the data non linearly separated. We learned that tuning the hyper parameters increases the accuracy of the classifiers. Naiive Bayes tends to have lower accuracy for both examples. Decision Tree classifier performed the best for the first while had a below par performance for the second dataset. KNN, SVM and Logistic regression performed really well for the image data set. We also found out that classifiers tend not to perform well when the data set is skewed as can be seen in the first data set where the F1-score for all the classifiers was on the lower side.