Machine Learning Final Report

Radu Constantinescu: 5043654
and Luca Pantea: 5040582
*~ **Group 40** ~*

Assignment 2.1

Question 1

GaussianNB

One key strength is that it models a vast number of real cases (e.g. document filtering), while only requiring a relatively small amount of sample data to estimate parameters for classification. A weakness is that under the "naive" assumption of conditional independence between the features, the posterior probability estimate will be affected if a class label frequency is low.

DecisionTreeClassifier

One main advantage of Decision Tree Classifiers is that nonlinear relationships between the underlying parameters do not affect the overall efficiency of the tree. On the other hand, a key disadvantage is that overly explicit decision trees are susceptible to cause overfitting if algorithms such as post-pruning are not implemented.

KNeighboursClassifier

One main advantage of K-Neighbours Classifiers is that by being an instance-based learning algorithm, training a model for generalisation is no longer a concern - thus, the computational cost for the training is kept at a minimum. Nonetheless, the performance decreases rapidly as the feature space increases (curse of dimensionality).

SVC

One of the biggest advantages of the SVC algorithm is that it performs well even without having too much prior knowledge about the data while also being accurate in higher dimensions. However, the downside is that the algorithm does not perform well if the data set is too large or there is a lot of noise in the data.

LogisticRegression

One Key use of the algorithm is that given its relatively easy implementation it can provide useful information about the correlation of features. On the downside, the algorithm does not fit well all scenarios, being able to solve only linear problems, and giving accurate results only when large datasets are provided.

Question 2

DecisionTreeClassifier

By leaving the max depth parameter to be none, we let the decision tree grow and become more complex, capturing more detailed information about the structure of the data. However this can have a downside, the model can overfit on the training data and this would produce bad results on the test sets.

By using a small number of samples leaf, the tree has a high probability it will overfit, meaning that the outcome will have a high bias towards the training data.

KNeighborsClassifier

Firstly, choosing an odd number for the KNN classifier is good because it avoids ties, however, if the data is not sufficient, meaning there is no large amount of training, the value 3 can be too less and result in a lot of variance and instability around the decision boundary.

Secondly, using the distance as the weight measurement means that the data within the closest distance of the new point will have the most impact on the classification.

SVC

A lower C will result in a higher error threshold (miss-classifications are accepted), and thus, a higher chance of underfitting, while a higher C will result in overfitting.

The kernel hyperparameter indicates the type of transformations the data undergoes to minimize cost. For nonlinear data, a linear kernel is outperformed by an RBF kernel. Random_state sets a seed to the random number generator, for the train-test split to be deterministic. The value will have an effect on the reproducibility of the returned result.

LogisticRegression

"C" (inverse of regularization strength) combats against high variance in the model by increasing the bias - lower C values reduce overfitting, at the expense of adding bias to the estimations (which can result in underfitting). The penalty hyperparameter is used to regulate the bias of the features' wights. For instance, "L1" penalty reduces the less important features to 0 and thus reduces overfitting. The random_state value determines the reproducibility of the results (same results on different calls).

Assignment 2.2 - US CENSUS

2.2.1 Data exploration

Question 1

Considering that in the training dataset, the target variable has an unbalanced distribution of samples belonging to each class (mentioned in the Jupiter file: ~75.92% for < $50k and ~24.08% for > $50k), having the classification accuracy as a performance measure would yield rather erroneous predictions. Therefore, a performance metric that could accurately evaluate the chosen model and would account for the uneven class distribution is the F1 score, which uses the harmonic mean between precision and recall to penalize outliers and high variance in data.

Question 2

Given the early 1990' situation regarding gender and race wage gap in the US, which consisted of notable discrepancies in the salaries of minorities and between women and men (30.51 % of men >$50k, 11.12 % of women >$50k), it can be presumed that the algorithm will inherit an implicit bias from the underlying training set. Thus, I firmly believe that race and sex would not make good choices for features and should not be used.

2.2.2 Data preparation

Question 1

As shown in the Jupyter file, there are 1576 rows with missing values. Removing those rows, which account for ~ 9.68% out of the training dataset would possibly result in underfitting, followed by accuracy tradeoffs. Since the numerical data (age, education hours and work hours) behaves mostly like a normal distribution, missing values will be replaced by the feature's mean. The majority of the categorical features, as analysed in the notebook, have most of their data within 2 or 3 categories (with the exception of education type), and thus, the most frequent imputation technique will be used.

Question 2

The categorical features (workclass, education, marital-status, occupation, relationship, native-country) were encoded using the OneHotEncoder, which represents those features in a binary fashion. This, while being more computationally expensive, allows the model to learn in an unbiased fashion, being that for the underlying attribute values, there is no ordinal relationship (e.g workclass: Private and Self-emp-inc are not comparable and nor are Married-AF-spouse and Married-civ-spouse).

Question 3

To prevent overfitting and improve the model's performance, the categorical features "native-country", "marital-status" and "education" were categorically binned. Firstly, North and South Americans make up for ~ 89.66% of the dataset; Thus binning all countries per continent will aid the performance of the classifier while combating overfitting. Secondly, the categories of marital-status can be reduced to married, not married, widowed and separated. Thirdly, the education feature contains all educational stages. Hence, these can also be grouped to further improve the model's robustness. Finally, L2 normalization is applied to numerical features to adjust the values' ranges and to increase the model's stability.

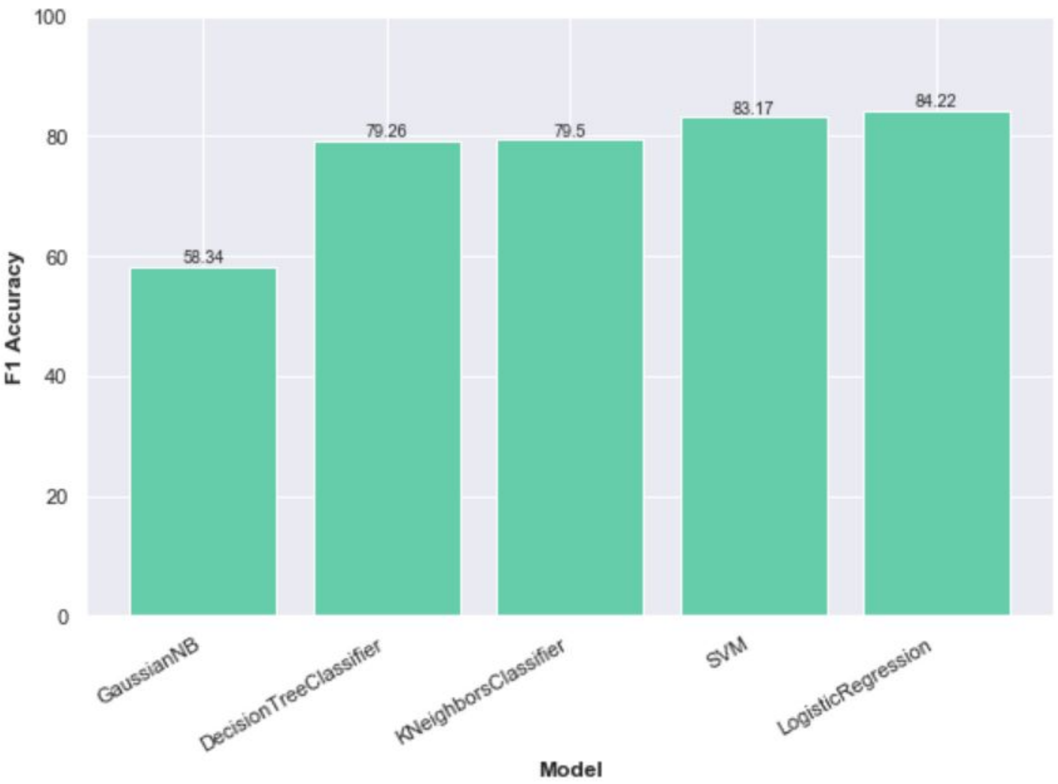
2.2.3 Experiments

Question 1

The sample data was divided and evaluated using K-fold cross-validation which makes use of resampling to shuffle and separate the data in k groups, evaluating the model on the first group, while training it on the remaining. This process combats overfitting in the training phase (through shuffling and splitting the training data). One downside of this procedure is that albeit resourceful, it is computationally more expensive than standard methods (train_test_split). As mentioned previously, the f1 scoring method was applied in order to account for the unbalanced distribution target labels.

Question 2

The plot indicates that the algorithms perform relatively acceptable, even in the absence of hyperparameter tuning (LogisticRegression's f1 score being around 84.2%). Two possible factors that make the LogisticRegression classifier outperform the others is that the underlying dataset was normalised in the pre-processing stage, which allows the gradient descent to converge in a more controlled manner, and that the data appears to be linearly separable (factor related to the nature of the dataset). The Naive Bayes Gaussian classifier has a rather low performance, which can be explained through the assumed conditional independence between features of this dataset (e.g. education-num is highly correlated to education, and thus, dependent; age and marital-status). Furthermore, the substandard performance of the DecisionTreeClassifier can be justified by one of its main drawbacks - the algorithm underperforms due to bias towards features with a greater number of categories (e.g. occupation: 14 unique categories).

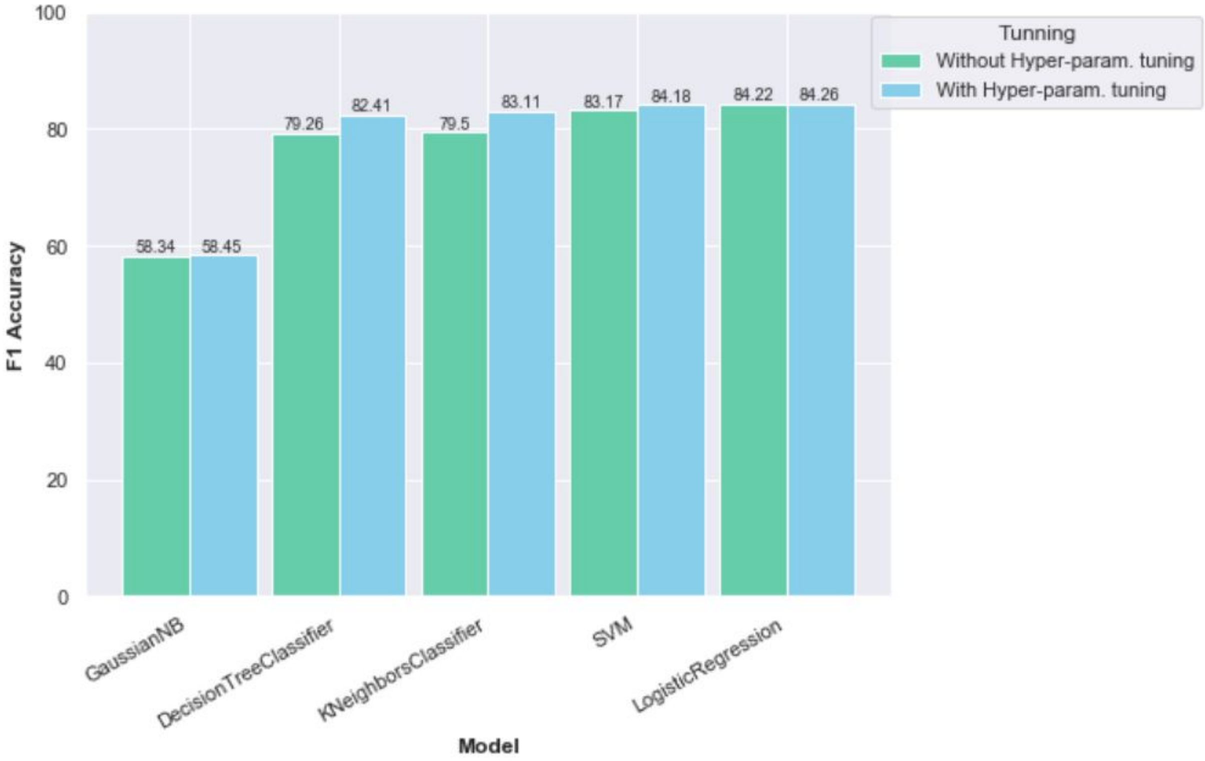


Question 3

For performing hyperparameter optimization, sklearn's GridSearchCV algorithm was used, which exhaustively searches through specified parameters, using cross-validation to select the optimal ones. Since finding the algorithms' best parameter combination would be computationally expensive, in-depth tuning will be performed only on the **SVM** and **LogisticRegression** classifiers, as those are promising to yield better results; the other algorithms' parameter space will be reduced. The approach used was guided by 2 core principles: firstly, allow for a tradeoff between variance and bias, in which models would be susceptible to increased variance, while bias would be minimised; secondly, combating unnecessary computations through reasoning about the nature of the dataset before setting the hyperparameter values. Through experimentation (setting a strict regularization parameter and a linear kernel for SVC, while training and testing on the same data) it was found that the data appears to be mostly linearly separable. Thus, to save computational resources, SVM's linear kernel was the only kernel used in the optimization process. Finally, to have an adequate assessment of the model's performance, Kfold cross-validation was applied to tune the parameters. This methodology, although computationally expensive, provides a resource-efficient alternative, in which the training data is reused to fit and evaluate the model.

Question 4

All algorithms (except GaussianNB, which had no tuning done), as expected, performed better, on average, rising to about 83%. In particular, KNN and DecisionTreeClassifier performed much better after hyperparameter tuning since their parameters required scaling to account for the size of the dataset (e.g. KNN: n_neighbours: 10; DecisionTreeClassifier: max_depth: 10, min_samples_leaf: 7). On the other hand, SVC and LinearRegression did not increase their performance substantially, possibly because both models are fitted with mostly binary data (obtained from OneHotEncoding), predicting a binary target variable, which makes both classifiers behave similarly.



Question 5

The algorithm chose is **Logistic Regression**, with parameters:

- C = 10
- penalty = “l2”
- max_iter = 1000
- random_state = 42

Assignment 2.3 - MNIST DATASET

2.3.1 Data exploration

Question 1

After plotting the images side by side from both datasets, I can say that my prediction is that the dataset with more pixels 28x28 will perform much better in recognizing handwritten digits. In my view, in problems that require recognizing patterns the more data (pixels) you provide for each image in the dataset the better and more clear the result will be, however, there is a limit to how many pixels you provide since the amount of memory is limited. Albeit one thing that I may suspect is the fact that the 28x28 dataset has a higher chance of overfitting.

2.3.2 Data preparations

Question 1

After examining both datasets, I can say that there is no need for data cleaning since there are no missing values. However, both datasets require the same way of preprocessing: firstly, reshaping them from 3d to 2d in order to make them a more suitable input for the algorithms. Secondly, the most important change was bringing the values of the pixels into the 0,1 range using the sklearn Minmax function, in order to make them fit more easily into the standard classifiers.

Lastly, the datasets were transformed into panda DataFrames for easier visualization.

2.3.3 Experiments

Question 1

For splitting the data we decided to use Kfold cross-validation in order to get the most of our initial dataset. We also decided to take things further and make a comparison of the performance of the algorithms when using the classic Kfold and StratifiedKfold (which makes sure that each split in the data has a balanced number of labels). As you can see on the graphs there is not a huge performance improvement, the reason being that the initial dataset had a somewhat equally distributed data in the training label. Given this information, we decided to use the accuracy metric for comparing performances since the data is balanced.

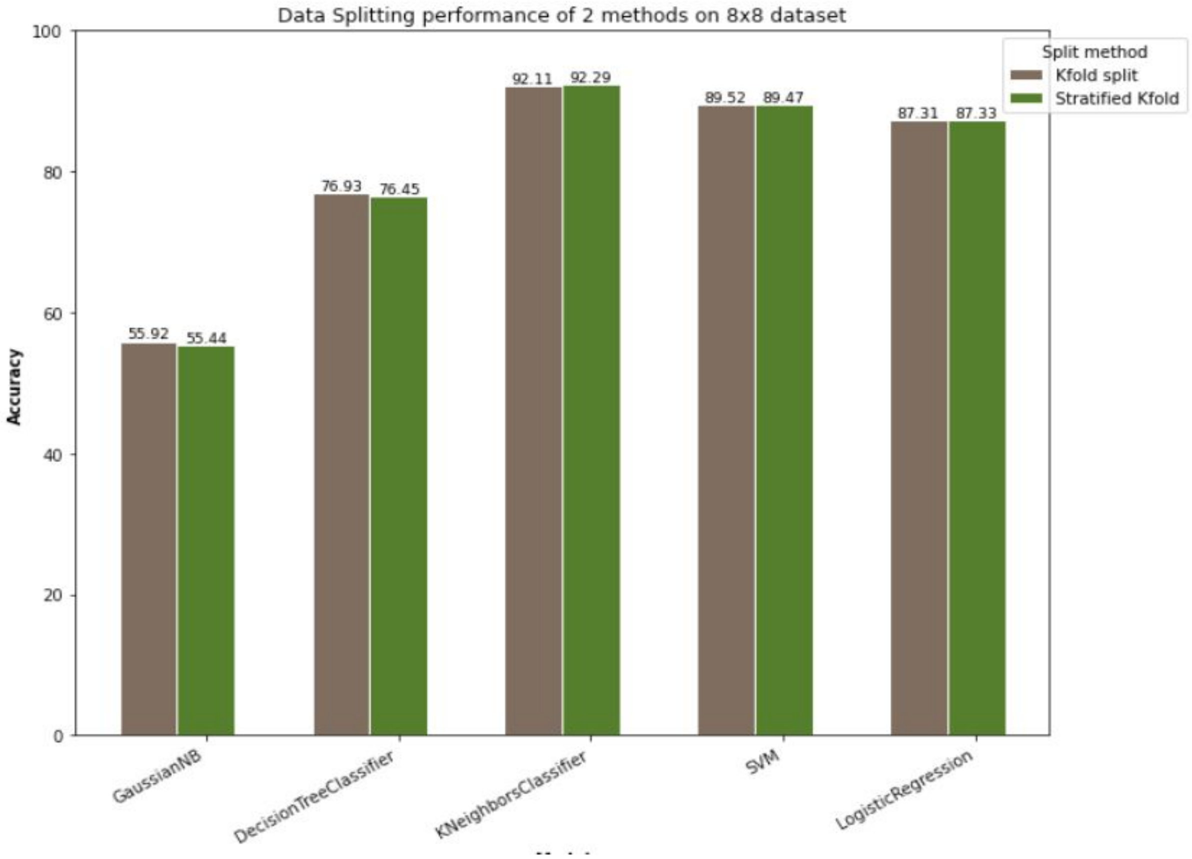
Question 2

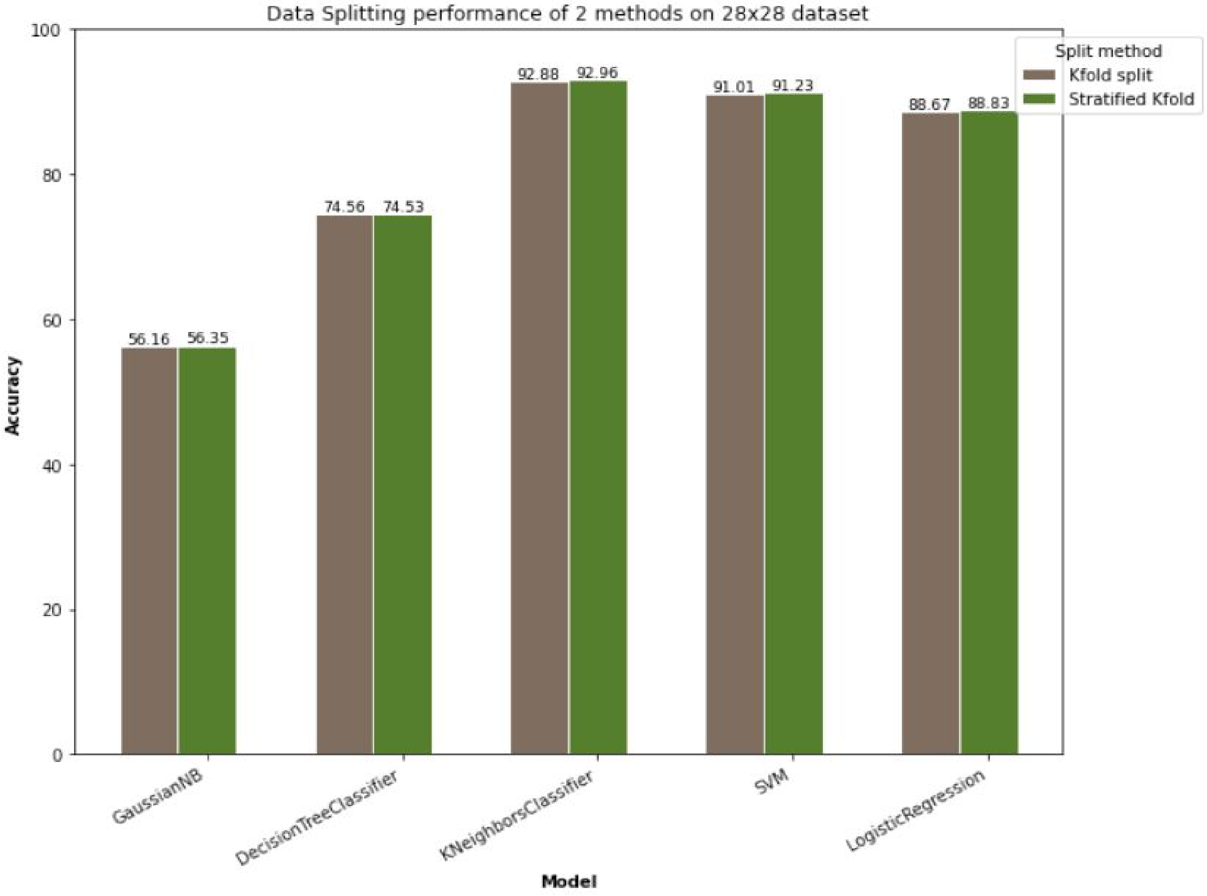
From the data plotted we can see that on both datasets there are three algorithms which perform considerably better, given no hyperparameter tuning (KneighboursClassifier, SVM and Logistic Regression with scores of 92% ,85% and 87% respectively). One valid reason for the KneighboursClassifier to perform well even without tuning is that the data has been preprocessed and scaled into a range and equally balanced. In the Logistic Regression case, the algorithm performed reasonably well also because there was no feature that was 'redundant', also the fact that the algorithm was trained using Kfold cross-validation, creates good results for a logistic classifier since it is able to distinguish between every digit individually (0/1,0/2..etc)

An interesting remark can be done when comparing the performance of 2 algorithms: DecisionTreeClassifier and SVM.

The DecisionTreeClassifier seems to perform better on the 8x8 dataset by a 2.6% margin (77%) this can be explained by the nature of the classifier which is very sensitive to the data, more exactly the noise that is far more predominant in the 28x28 dataset combined with the fact that small variances in the dataset can have a significant impact on the outcome of the classification.

The SVC performs better than the others since this classifier gives good results when the data is mostly not linearly separable.


Data Splitting performance of 2 methods on 8x8 dataset
Split method
Kfold split
Stratified Kfold
Accuracy
100
80
60
40
20
0
55.92
55.44
76.93
76.45
92.11
92.29
89.52
89.47
87.31
87.33
GaussianNB
DecisionTreeClassifier
KNeighborsClassifier
SVM
LogisticRegression


Data Splitting performance of 2 methods on 28x28 dataset
Split method
Kfold split
Stratified Kfold
Accuracy
100
80
60
40
20
0
56.16
56.35
74.56
74.53
92.88
92.96
91.01
91.23
88.67
88.83
GaussianNB
DecisionTreeClassifier
KNeighborsClassifier
SVM
LogisticRegression
Model

Question 3

Just as for the US CENSUS dataset, the same method was used for performing the hyperparameter tuning, sklearn GridSearchCV combined with KFold cross-validation. However this time the focus was specifically on the KNN and SVM classifier. Firstly, after trivial research (finding the non-linearity in the data samples) and experimentation, our intuition was that given the right tuning, the SVM classifier would perform best on the MNIST dataset. The hyperparameters that can create the most impact on an SVM classifier are: (Regularization - C; Type of kernel and the Gamma representing the kernel coefficient). The regularization parameter mainly focuses on the model's acceptance to misclassification and the right values could combat overfitting given the noise of the provided sample data. The c parameter cannot be too low (as initially 0.5) but not too high either because it can overfit the data, and therefore resulting in low accuracy, so identifying the right values is a necessity.

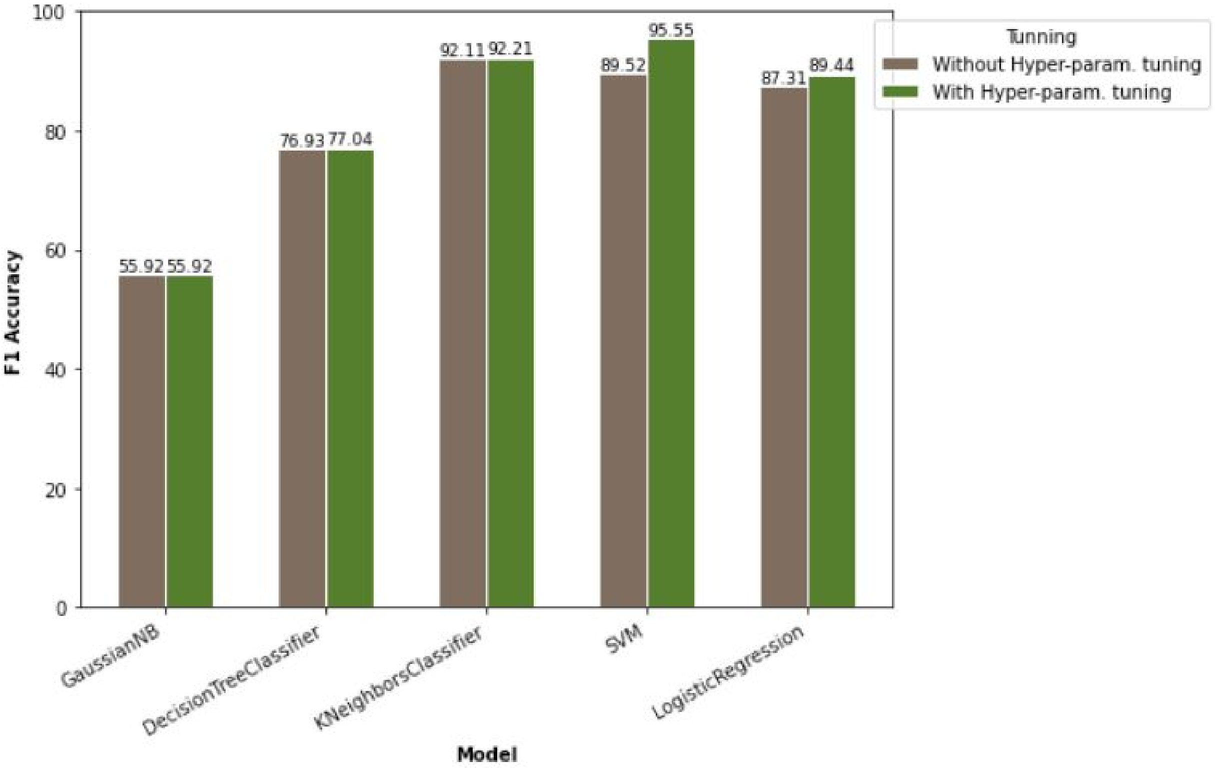
The type of kernels that we put into test were: polynomial and RBF kernels, which we believed to be more effective given the nonlinearity of the training data. Additionally, the gamma range exercised was [0.001 to 100].

Question 4

After inserting all the previously stated inputs of the hyperparameters in the GridSearch we managed to get a performance improvement of 5,4% reaching an accuracy of 95.5% on the 8x8 dataset using the SVM classifier.

Algorithms such as GNB and DecisionTreeClassifier did not get huge importance when regarding tuning because it was quite clear that in this scenario they would outperform the other algorithms.

The KNN classifier got almost no improved performance since the GridSearch concluded that the only difference would be to choose k=4 neighbours instead of 3, with 'uniform' distance to all neighbours, parameter which makes sense considering that a pixel can have max 4 neighbours and the distance to them is always equal. The Logistic Regression received a 1.74% improvement, due to the change from 0.5 to 7 of the penalty parameter. Below we inserted a plot where we show the effect of tuning in the 8x8 dataset.



Question 5

In conclusion, contrary to what we initially believed, the dataset which managed to perform the best given the most suitable algorithm was the 8x8 dataset. Overall, tuning had an impact on the algorithms we expected to perform well, mostly on this matter the nonlinear, discriminative ones. However, on the other dataset, the KNN managed to outperform the SVM classifier, due to the amount of sparse data and noise in the bigger dataset.

Question 6

The dataset chosen is 8x8 mnist, with the **SVM classifier**, with parameters

- C = 4
- gamma = 0.7
- kernel = rbf
- random_state = 42

2.4 CONCLUSION

GaussianNB
The GaussianNB classifier underperformed as expected since the sample datasets characteristics (correlation between features and therefore close to no independence among them; categorical features, encoded with 0 and 1, do not fit a normal distribution) fight the core assumptions made by this classifier.

DecisionTreeClassifier
Even though in our case the classifier did not have the best performance, we found it rather interesting that in the absence of normalization and tuning, performance remained somewhat constant, suggesting that with minimal preprocessing and computational resources, a good overall performance can be achieved for non-complex classification tasks.

KNearestNeighbours
Given its instance-based learning nature, the KNearestNeighbours algorithm performed quite well, even under no hyperparameter tuning done and given a large dataset, all without consuming large amounts of computational resources. One drawback though is that without standardization and normalization of the dataset, the classifier is inclined to give erroneous predictions.

SVM
One particular conclusion drawn from our experiments with the SVM is that it is one of the classifiers from which you can expect to have decent results given the right parameter tuning such as the kernel, regularization parameter and ‘gamma’ kernel coefficient, given that the nature of the data is assumed beforehand (in the case of the US CENSUS, linearity, MNIST - RBF kernel).

LogisticRegression
At last, the LogisticRegression classifier performed well on average, being the go-to classifier in the case of the US CENSUS. An important point that we concluded upon is the fact that with enough preprocessing to reduce the feature space, and given mostly linearly separable data, the classifier not only yields good results but is computationally efficient.