COL341 Assignment-3 Report

Amaiya Singhal (2021CS50598)

3.1 Binary Classification

Overview

- In this part, the task was to classify 32x32x3 dimensional images into Persons and NOT Persons using **Decision Trees**.
- Part 3.1a contains my implementation of Decision Trees, rest parts have used the `scikit-learn` library.
- The confusion matrices on the train and validation data have been placed with each part and not separately in 3.1g.
- After experimenting with different hyperparameters in each of the subparts, the best parameters obtained have been used in the `main_binary.py` file.

a) Decision Tree from scratch

- In this part I have implemented a decision tree from scratch.
- I have used **Gini Index** and **Information Gain** in two separate implementations as splitting criteria.
- The halting criteria was `max_depth` = 10 and `min_samples_split` = 7.
- Since the model has been implemented completely from scratch, it is much slower than the `scikit-learn` implementations.

Information Gain

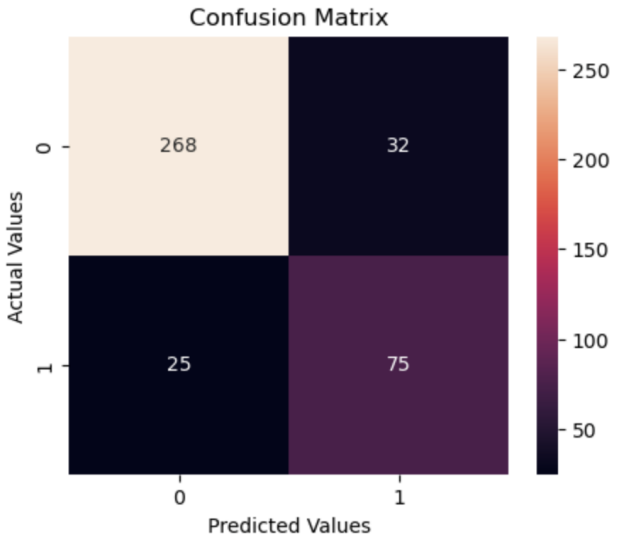
`max_depth`	`min_samples_split`	Time Taken	Accuracy on Train	Precision on Train	Recall on Train
10	7	26 min	94.7%	86.3468 %	93.6%

`max_depth`	`min_samples_split`	Time Taken	Accuracy on Val	Precision on Val	Recall on Val
10	7	26 min	85.75%	70.093458 %	75.0%

Observations

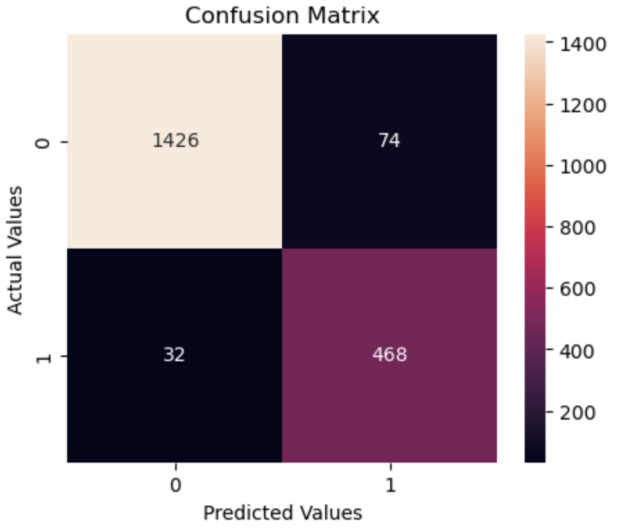
- The accuracy on the train and validation sets were quite high.
- This shows that **Information gain** is a very good splitting criteria in this case.

Confusion Matrix on the Validation Set



Grid Search best parameters

Confusion Matrix on the Training Set



Grid Search best parameters

Gini Index

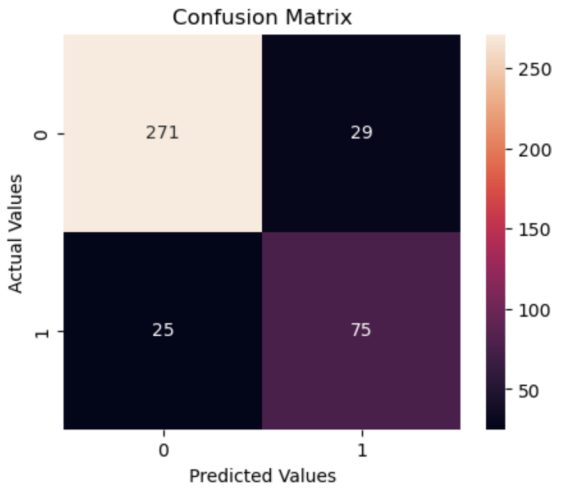
max_depth	min_samples_split	Time Taken	Accuracy on Train	Precision on Train	Recall on Train
10	7	22 min	94.5%	86.380597 %	92.6%

max_depth	min_samples_split	Time Taken	Accuracy on Val	Precision on Val	Recall on Val
10	7	22 min	86.5%	72.115385%	75.0%

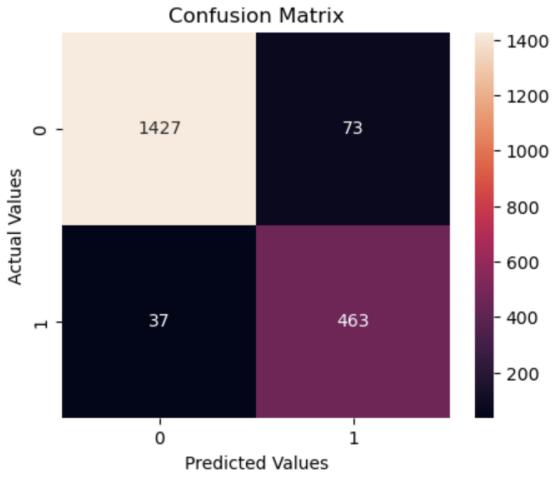
Observations

- The accuracy on the train and validation sets were quite high.
- This shows that **Gini Index** is also a very good splitting criteria in this case.

Confusion Matrix on Validation Set



On Validation Set



On Training Set

b) Decision Tree sklearn

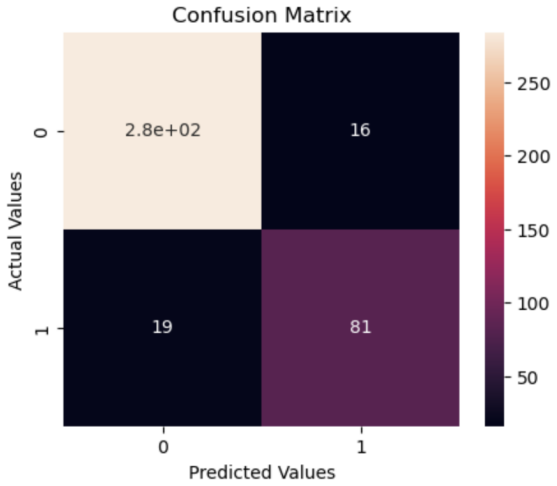
max_depth	min_samples_split	Time Taken	Accuracy on Train	Precision on Train	Recall on Train
Default	Default	3.461752s	100%	100%	100%
10	Default	2.498956s	99.25%	100.0%	97.0%
Default	7	3.3435s	99.35%	98.7975952 %	98.6%
10	7	2.465323s	98.85%	98.773 %	96.6%

max_depth	min_samples_split	Time Taken	Accuracy on Val	Precision on Val	Recall on Val
Default	Default	3.461752s	91.25%	83.505155%	81.0%
10	Default	2.498956s	92.0%	85.416667%	82.0%
Default	7	3.3435s	93.0%	89.130435%	82.0%
10	7	2.465323s	93.0%	90.0 %	81.0%

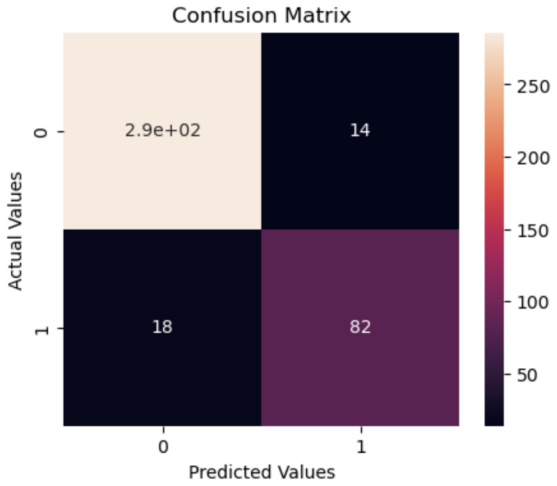
Observations and Comparison

- The best results were obtained in the 2 cases where the `min_samples_split` was set to 7 among all the 4 combinations tried.
- Setting `max_depth` to 10 also resulted in slightly better accuracy, precision and recall on the validation set.
- Though, restricting the growth of the decision tree on either of the 2 parameters leads to a smaller tree grown which consequently takes less time to build.
- The reduction in time is more noticeable when we restrict the `max_depth` to 10 which tells us that the size of the original tree was larger than 10 and hence took more time to train compared to the tree with restricted depth.
- There is also a slight reduction in the train accuracy when restricting the free growth of the tree, however this tells us that the original tree had overfitted on the train data and this reduction in train accuracy actually has led to an increase in the validation accuracy.
- The best parameters in this case were observed to be
  1. `max_depth` = 10
  2. `min_samples_split` = 7
- The accuracy on validation set for this case was 93.0% and time taken to train was 2.46 seconds.

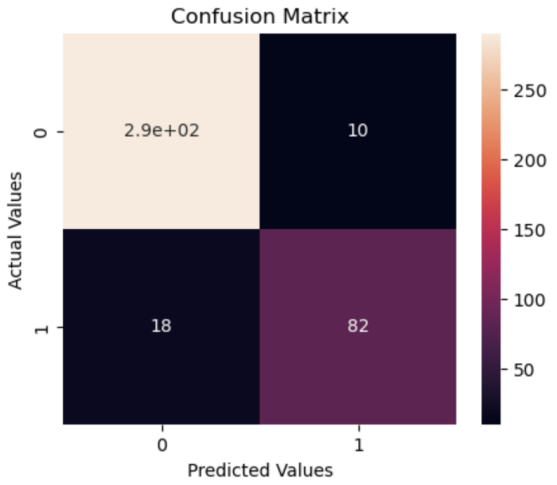
Confusion Matrix on Validation Set



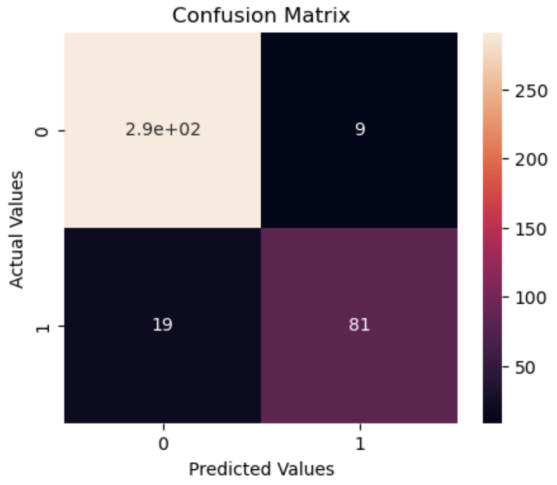
Default Parameters



max_depth = 10

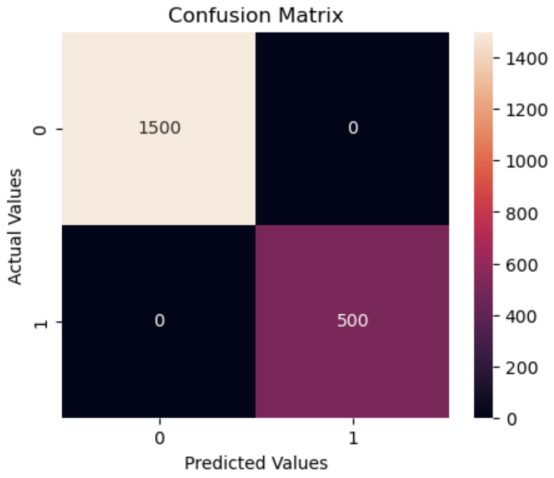


min_samples_split = 7

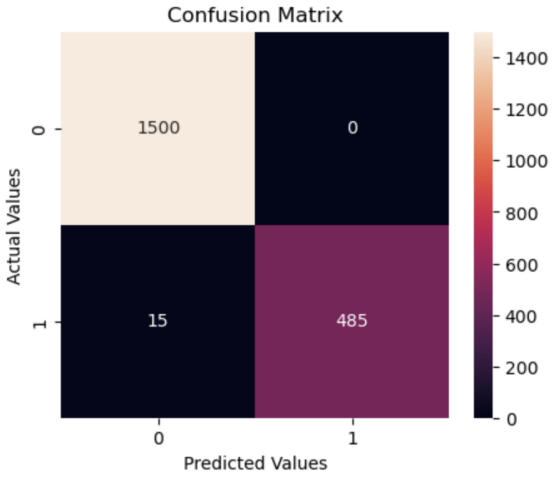


depth = 10, split = 7

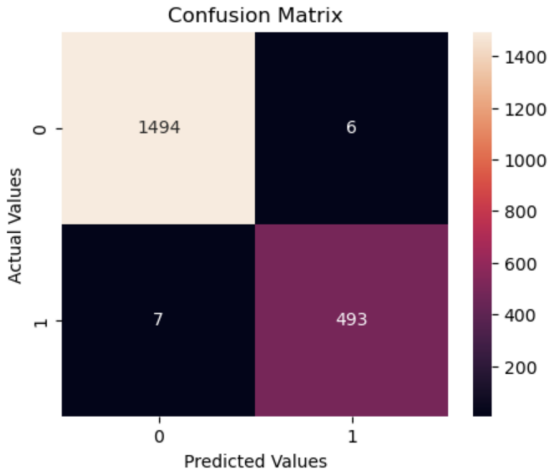
Confusion Matrix on Training Set



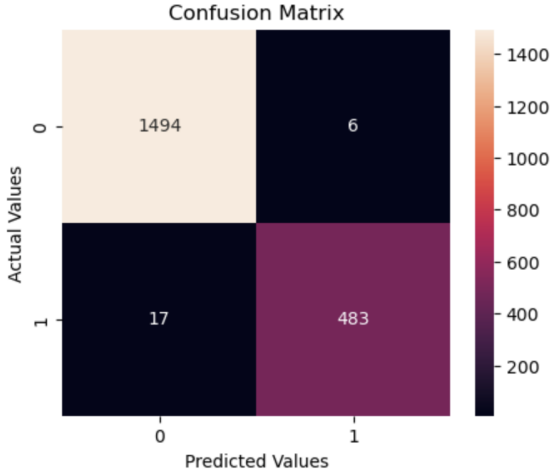
Default Parameters



max_depth = 10



min_samples_split = 7



depth = 10, split = 7

c) Decision Tree Grid-Search and Visualisation

Top 10 feature selection

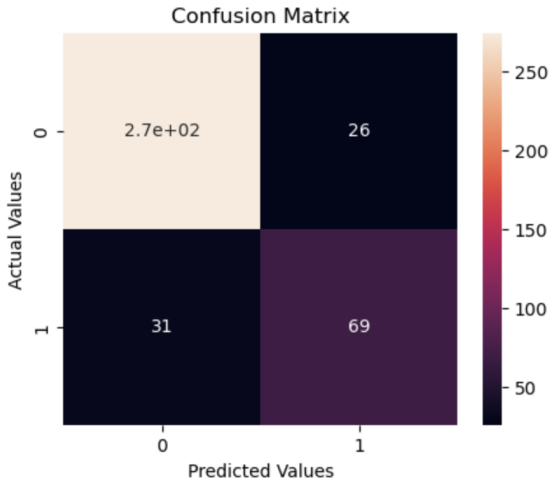
max_depth	min_samples_split	Accuracy on Train	Precision on Train	Recall on Train
Default	Default	100%	100%	100%
10	Default	94.0%	82.312925%	96.8%
Default	7	97.2%	96.8344 %	91.8%
10	7	93.0%	81.80212 %	92.6%

max_depth	min_samples_split	Accuracy on Val	Precision on Val	Recall on Val
Default	Default	85.75%	72.63158%	69.0%
10	Default	89.0%	75.926%	82.0%
Default	7	86.0%	73.9130%	68.0%
10	7	89.0%	76.9231%	80.0%

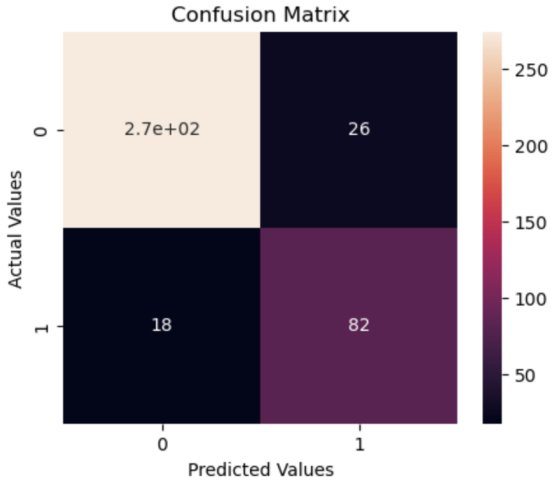
Observations and Comparison

- The `SelectKBest` function (a part of `sklearn`) was used in this case to obtain the top 10 features and the decision tree was trained using these 10 features.
- The best results were obtained in the 2 cases where the `max_depth` was set to 10.
- Setting `max_depth` to 10 also resulted in similar accuracy, precision and recall on the validation set.
- There is also a reduction in the train accuracy when restricting the free growth of the tree by specifying the `max_depth` and `min_samples_split` parameters.
- At the same time there is an increase in the validation accuracy when the growth of the tree is restricted. This tells us that the tree with default parameters had overfitted the train data (hence the 100% accuracy on train).
- The overall accuracy is less compared to the tree built with all the features because the features are now much less (10 instead of 3072), however the decrease in accuracy is small $\leq 7\%$ for the given parameters.
- Since there are much less features to train on in this case, the training time was also much less (in hundredths of a second compared to a few seconds in 3.1b).
- The best parameters in this case were observed to be the same as 3.1b and are as follows
  1. `max_depth` = 10
  2. `min_samples_split` = 7
- The accuracy on validation set for this case was 89.0%.

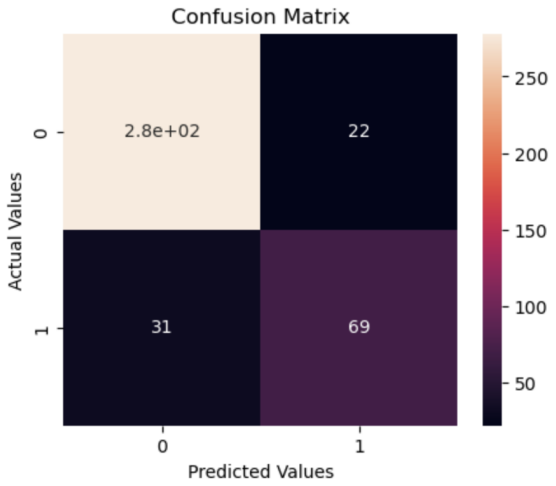
Confusion Matrix on Validation Set



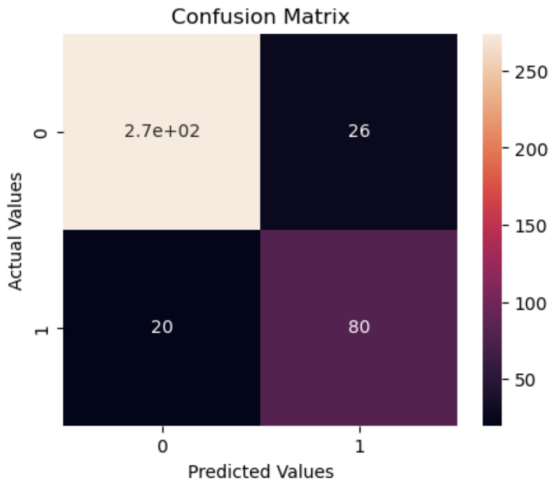
Default Parameters



max_depth = 10

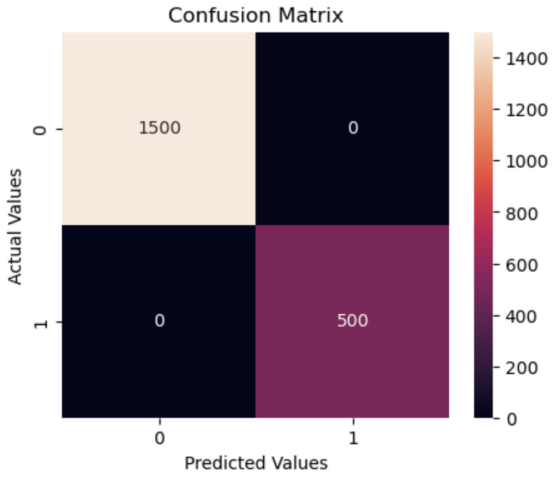


min_samples_split = 7

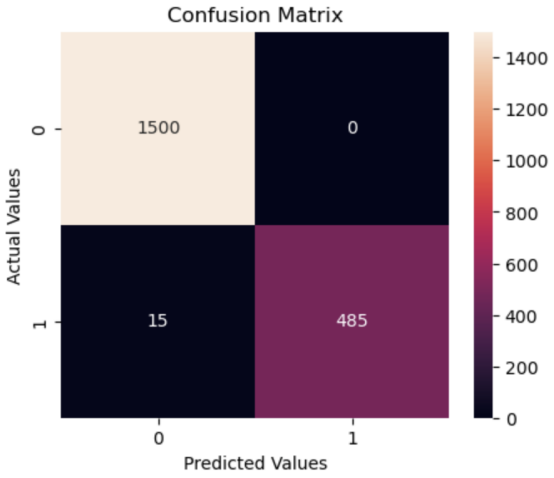


depth = 10, split = 7

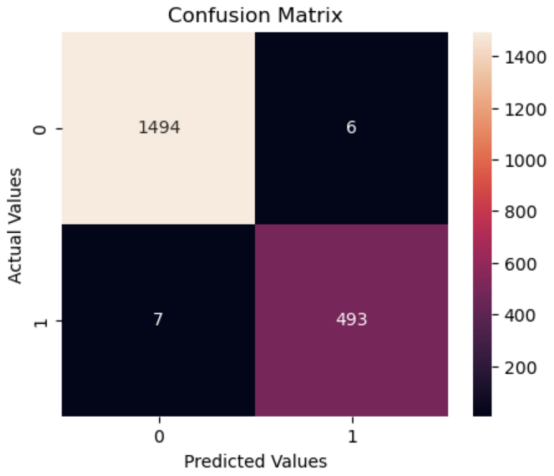
Confusion Matrix on Training Set



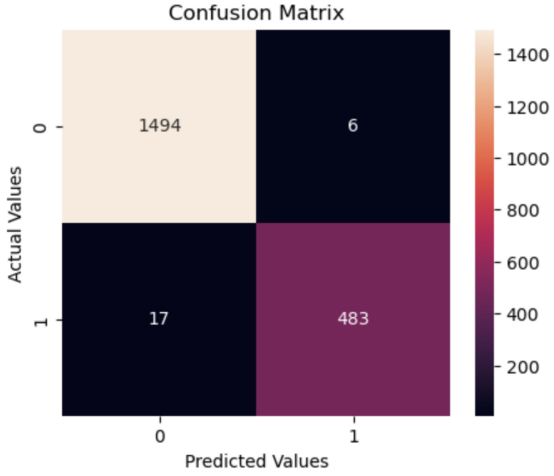
Default Parameters



max_depth = 10



min_samples_split = 7



depth = 10, split = 7

Visualizing the Tree

The trees built are very large and hence cannot be properly viewed here. The links for the visualised trees are provided below:

- Default Parameters
- Best Parameters

Grid Search

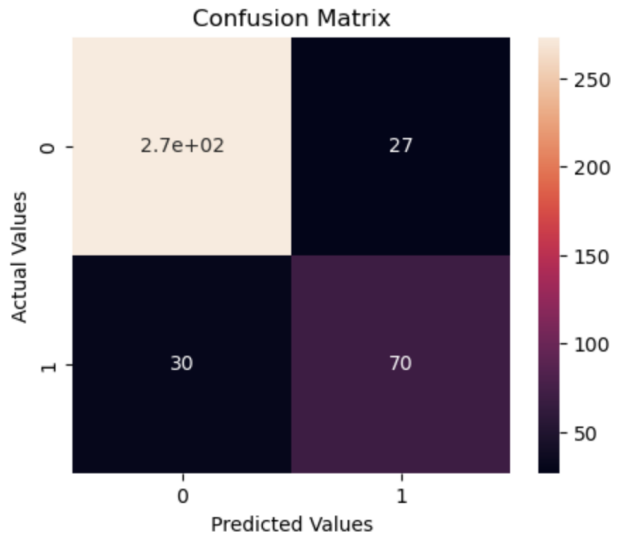
- In this part Grid Search was performed using the `GridSearchCV` implementation of `sklearn` on the decision tree classifier (with default hyper parameters) to find out the best set of hyperparameters.
- The best parameters came out to be
  1. criterion: entropy
  2. max_depth: 5
  3. min_samples_split = 4
- The accuracy on the train and validation sets for these parameters is as follows

Accuracy on Train	Accuracy on Validation
87.75%	87.75%

Observations

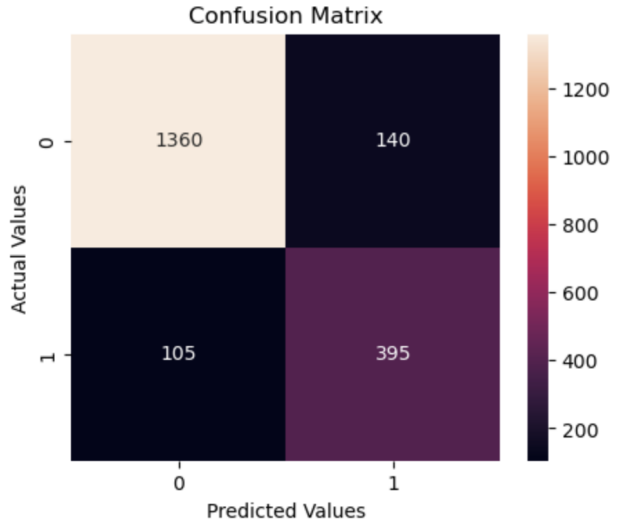
- On running the GridSearch multiple times, there was a lot of variation in the value of the parameter `min_samples_split`. It took values 2,4,7 and 9, however on calculating the accuracies on the validation set, the value 4 yielded the best results by a small margin and hence was chosen.
- This is not very different and in fact not better than any of the set of hyperparameters we had tried previously in c). All of them had yielded equally as good if not better results on the validation set.
- However the train accuracy in this case is significantly less than the train accuracy obtained when manually experimenting with the hyperparameters.
- This tells us that these parameters have clearly prevented overfitting but might just have underfitted the tree because we had observed a slightly higher validation accuracy with the last set of hyperparameters previously.
- But since Grid Search calculates the best parameters using cross validation, it is likely that the parameters returned are a better set.
- Compared to b) where we used all the features, the accuracy is less by about 6-7% (93.0% accuracy was obtained in 3.1b).
- Compared to a) it produces similar results.

Confusion Matrix on the Validation Set



Grid Search best parameters

Confusion Matrix on the Training Set

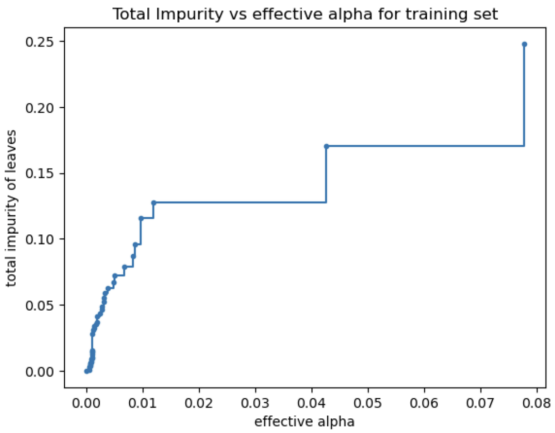


Grid Search best parameters

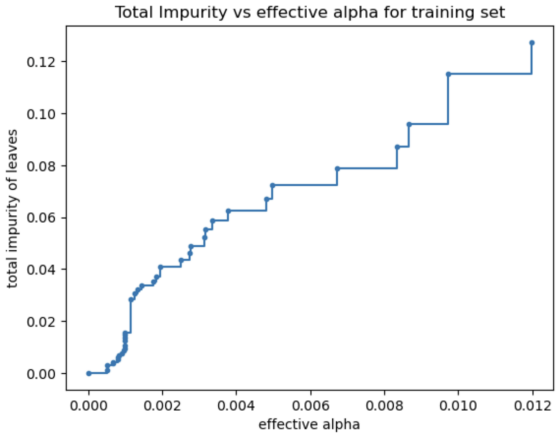
d) Decision Tree Post Pruning with Cost Complexity Pruning

- We used the `DecisionTreeClassifier.cost_complexity_pruning_path` (a part of `sklearn`) to obtain the total leaf impurities and effective alphas.
- Below are the various plots showing the variation of different parameters with the effective alphas which were asked.

Total Impurity v/s Effective Alphas

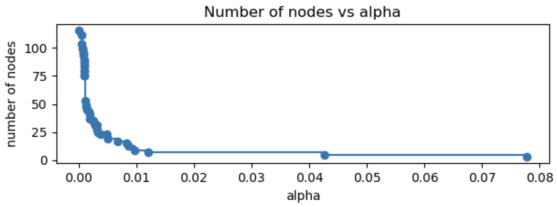


For all effective alphas

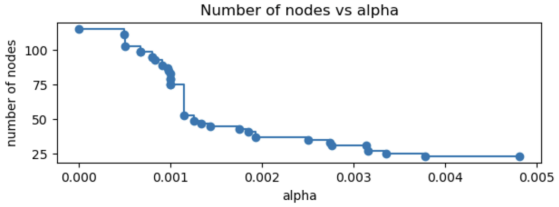


Excluding some alphas from the end

Number of Nodes v/s Effective Alphas

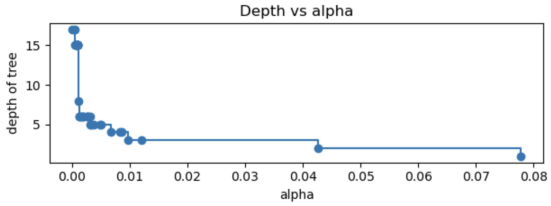


For all effective alphas

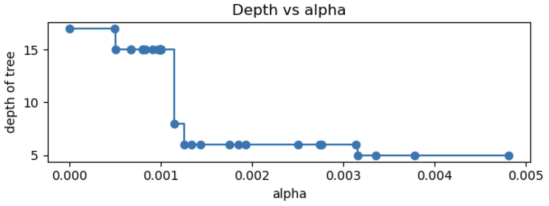


Excluding some alphas from the end

Depth v/s Effective Alphas

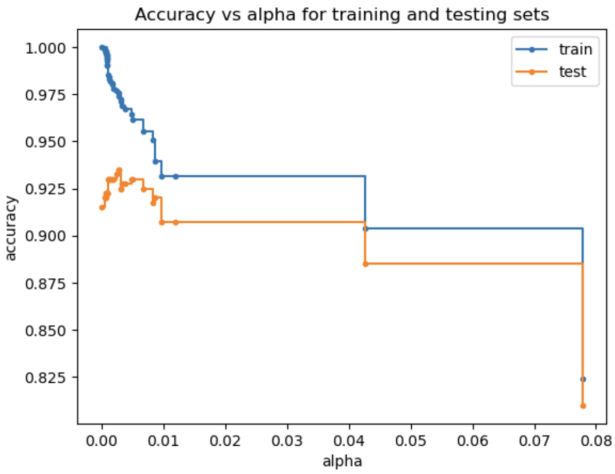


For all effective alphas



Excluding some alphas from the end

Accuracy on Train and Validation v/s Alphas



For all effective alphas

Observations

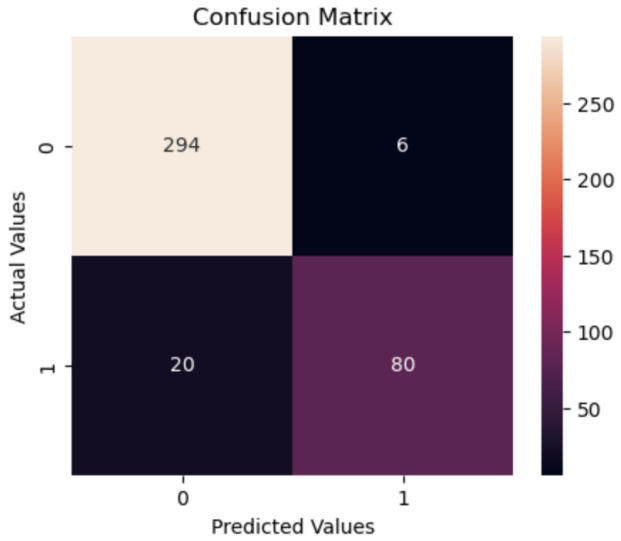
- As alpha increases, more and more of the tree is pruned, which increases the total impurity of the leaves as more misclassifications will occur.
- As more and more of the tree is pruned, the number of nodes reduces (pruning means removing nodes from the tree).
- As the number of nodes in the tree decreases, the depth of the tree also consequently decreases.
- As alpha increases, the validation set accuracy initially increases but soon reaches a maxima and starts decreasing.
- The maximum validation set accuracy of 93.5% is achieved at alpha = 0.0027352941176470593. The training accuracy at this alpha is 97.55%.
- The training accuracy decreases with increase in alpha. Because as the number of nodes reduces, there is a tradeoff between the training accuracy and the validation accuracy.
- Initially the tree has overfitted to the train set and produces almost 100% accuracy on the train set, however it performs poorly on the validation set.
- Upon pruning the train accuracy decreases slightly however there is a significant improvement in the validation accuracy.

Visualizing the best-pruned tree

The tree would not be clearly visible here hence I am attaching a link of the best-pruned tree that was obtained in this part.

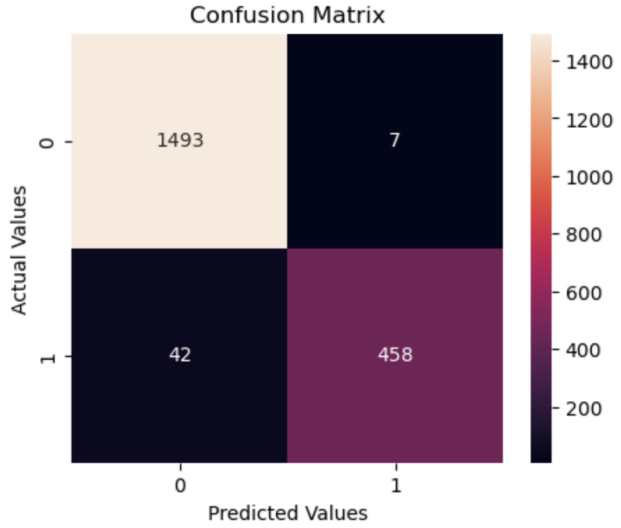
- Best Pruned Tree

Confusion Matrix on the Validation Set



Grid Search best parameters

Confusion Matrix on the Training Set



Grid Search best parameters

e) Random Forest

- We have used the `RandomForestClassifier` from `sklearn` in this part to use the **Random Forest** Classifier.
- The accuracy, precision and recall on the train and validation sets for default parameters is shown below.

Accuracy on Train	Precision on Train	Recall on Train
100%	100%	100%

Accuracy on Validation	Precision on Validation	Recall on Validation
97.75%	100%	91.0%

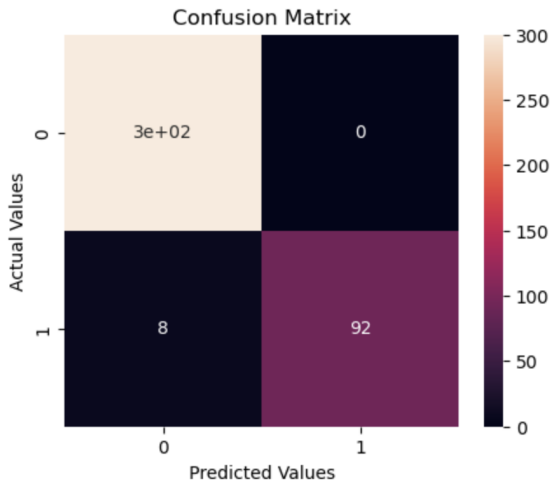
Grid Search

- In this part Grid Search was performed using the `GridSearchCV` implementation of `sklearn` on the Random Forest classifier (with default hyper parameters).
- The best parameters came out to be
  1. criterion: entropy
  2. max_depth: None
  3. min_samples_split = 10
  4. n_estimators = 150
- The accuracy, precision and recall on the train and validation sets for these best set of parameters is shown below.

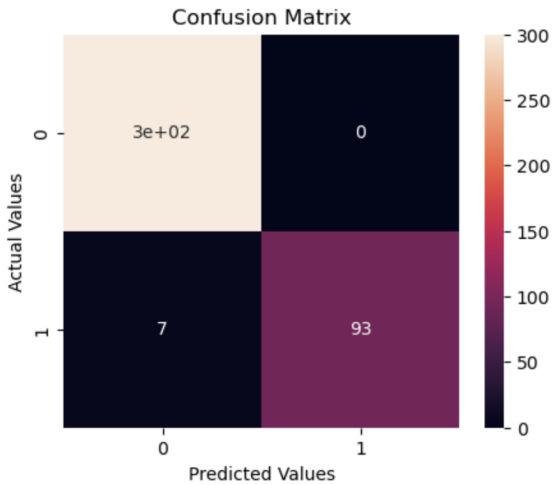
Accuracy on Train	Precision on Train	Recall on Train
100%	100%	100%

Accuracy on Validation	Precision on Validation	Recall on Validation
98.25%	100%	93.0%

Confusion Matrix on Validation Set

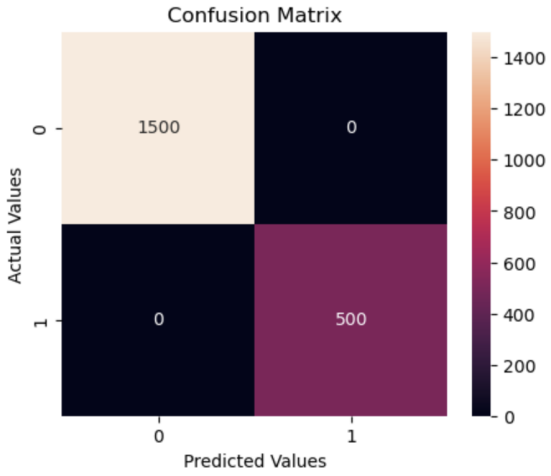


Default Parameters

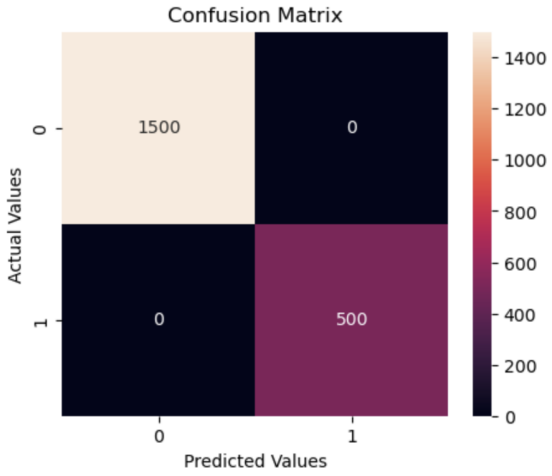


Best Parameters

Confusion Matrix on Training Set



Default Parameters



Best Parameters

f) Gradient Boosted Trees and XGBoost

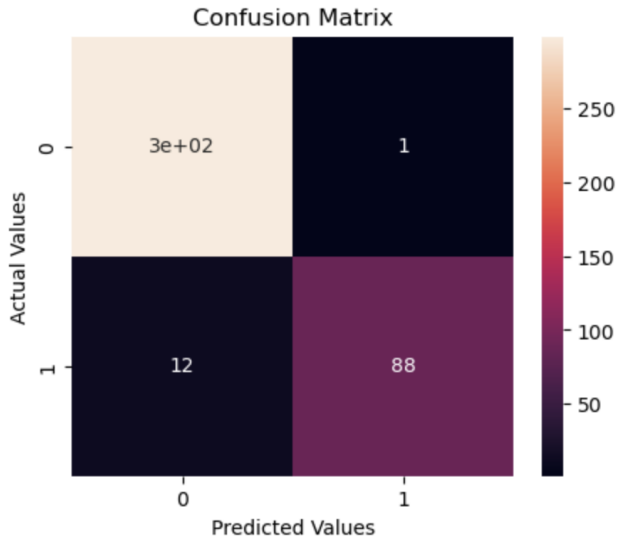
Gradient Boosted Trees

- In this part Grid Search was performed using the `GridSearchCV` implementation of `sklearn` on the Gradient Boost Classifier.
- The best parameters came out to be
  1. n_estimators: 50
  2. max_depth: 6
  3. subsample : 0.6
- The training time and the accuracy, precision and recall values in the training and validation sets is shown in the table below.
- The time taken to train is the time taken to perform Grid Search on the given sets of parameters.

Train Time	Train: Accuracy	Precision	Recall	Val: Accuracy	Precision	Recall
1h 32min	100%	100%	100%	96.75%	98.8764%	88.0%

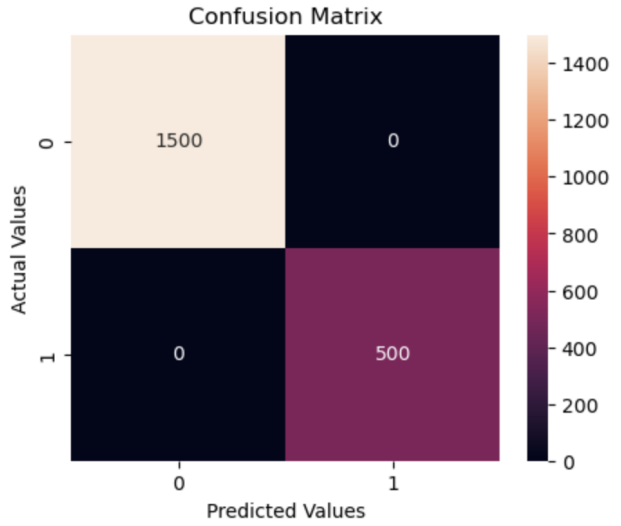
- We observe very high accuracy on the validation set (approximately 97%).
- However the accuracy is very slightly less than that obtained in he case of Random Forest Classifier with its best set of parameters.

Confusion Matrix on the Validation Set



Grid Search best parameters

Confusion Matrix on the Training Set



Grid Search best parameters

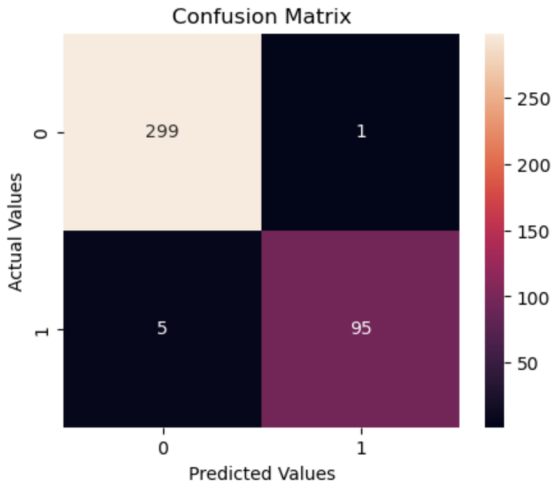
XGBoost

- In this part Grid Search was performed using the `GridSearchCV` implementation of `sklearn` on the Gradient Boost Classifier.
- The best parameters came out to be
  1. n_estimators: 40
  2. max_depth: 6
  3. subsample : 0.6
- The training time and the accuracy, precision and recall values in the training and validation sets is shown in the table below.
- The time taken to train is the time taken to perform Grid Search on the given sets of parameters.

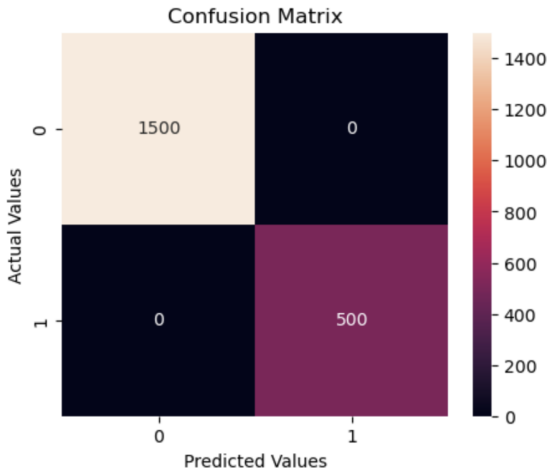
Train: Accuracy	Precision	Recall	Val: Accuracy	Precision	Recall
100%	100%	100%	98.5%	98.958333%	95.0%

- The accuracy obtained in this case is more than that obtained with the Gradient Boost Classifier.
- The accuracy is also slightly higher than the Random Forest classifier with its best parameters.

Confusion Matrix on Validation Set and Training Set



Validation Set



Training Set

3.2. Multi-class Classification

Overview

- In this part, the task was to classify 32x32x3 dimensional images into Persons, Cars, Dogs and Airplanes using **Decision Trees**.
- All the parts have used the `scikit-learn` library.
- The confusion matrices on the train and validation data have been placed with each part and not separately in 3.2f.
- After experimenting with different hyperparameters in each of the subparts, the best parameters obtained have been used in the `main_multi.py` file.

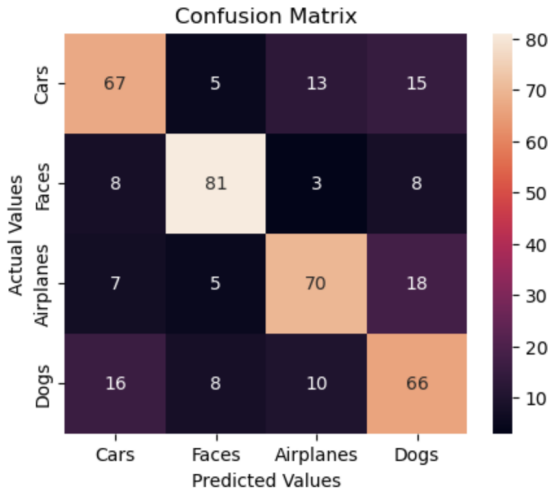
a) Decision Tree sklearn

`max_depth`	`min_samples_split`	Time Taken	Accuracy on Train	Accuracy on Validation
Default	Default	3.048776s	100%	71.0%
10	Default	2.820547s	98.5%	73.5%
Default	7	3.06214s	97.85%	72.5%
10	7	3.0672567s	96.90%	73.5%

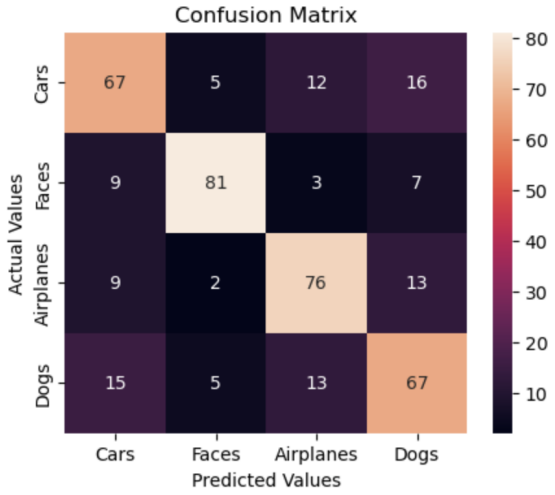
Observations and Comparison

- The best results were obtained in the cases where the `max_depth` was set to 10 among all the 4 combinations tried.
- Setting `min_samples_split` to 7 also resulted in better accuracy on the validation set compared to the default values.
- Though, restricting the growth of the decision tree on either of the 2 parameters leads to a smaller tree grown which consequently takes less time to build.
- The reduction in time is more noticeable when we restrict the `max_depth` to 10 which tells us that the size of the original tree was larger than 10 and hence took more time to train compared to the tree with restricted depth.
- However when we set both `max_depth` to 10 and `min_samples_split` to 7, there is no apparant reduction in the training time.
- There is also a slight reduction in the train accuracy when restricting the free growth of the tree, however this tells us that the original tree had overfitted on the train data and this reduction in train accuracy actually has led to an increase in the validation accuracy.
- The best parameters in this case were observed to be
  1. `max_depth` = 10
  2. `min_samples_split` = 7
- The accuracy on validation set for this case was 73.5% and time taken to train was 3.07 seconds.

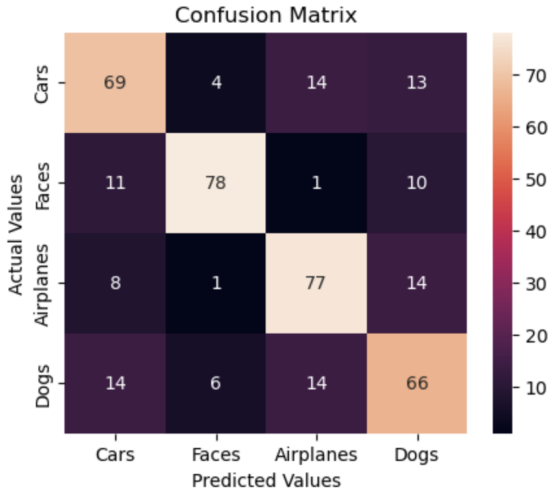
Confusion Matrix on Validation Set



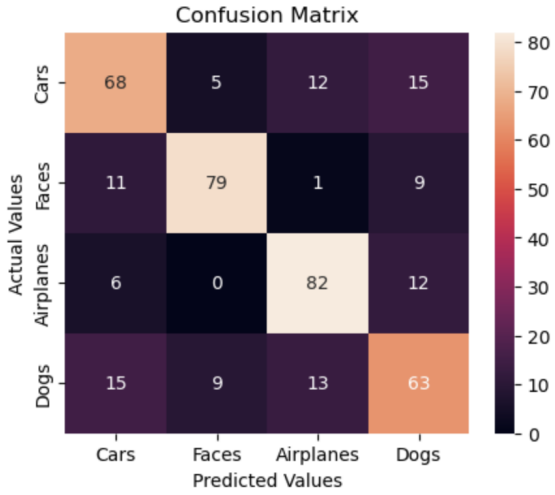
Default Parameters



max_depth = 10

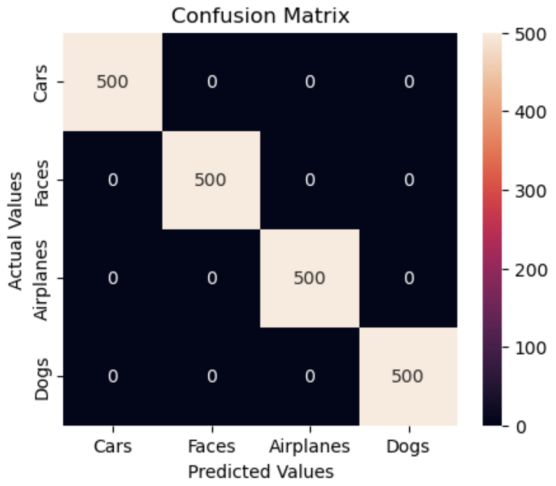


min_samples_split = 7

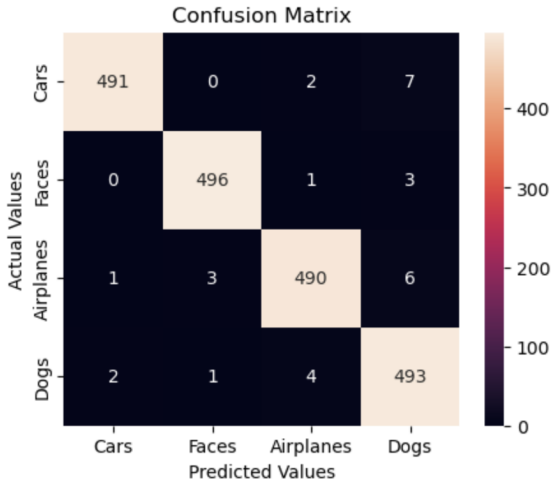


depth = 10, split = 7

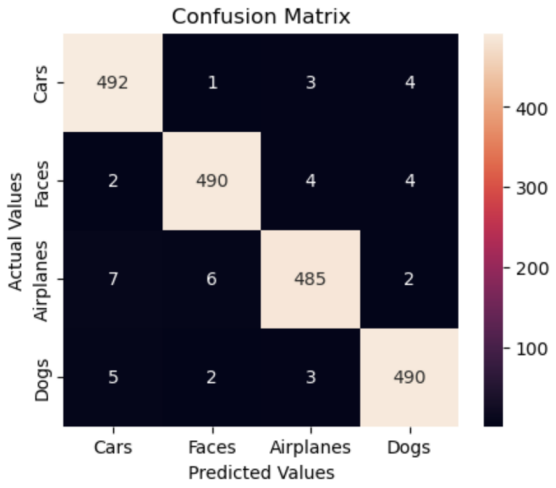
Confusion Matrix on Training Set



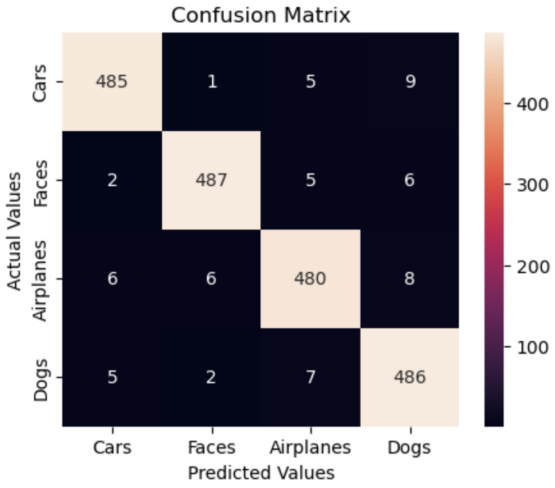
Default Parameters



max_depth = 10



min_samples_split = 7



depth = 10, split = 7

b) Decision Tree Grid Search and Visualisation

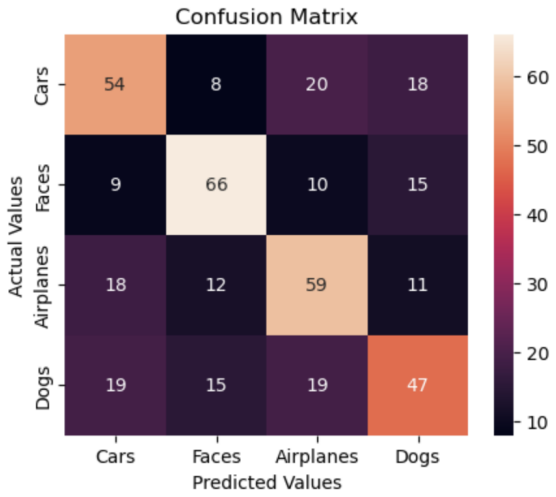
Top 10 features

max_depth	min_samples_split	Accuracy on Train	Accuracy on Validation
Default	Default	100%	56.5%
10	Default	84.25%	61.0%
Default	7	89.9%	57.0%
10	7	81.05%	63.25%

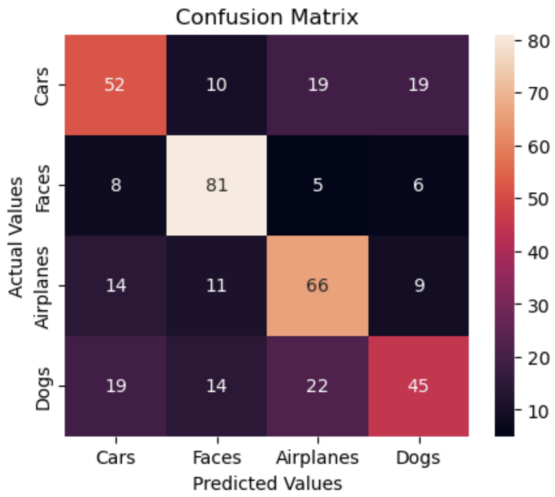
Observations and Comparison

- The `SelectKBest` function (a part of `sklearn`) was used in this case to obtain the top 10 features and the decision tree was trained using these 10 features.
- Setting `min_samples_split` to 7 and `max_depth` to 10 resulted in the best accuracy on the validation set.
- There is also a reduction in the train accuracy when restricting the free growth of the tree by specifying the `max_depth` and `min_samples_split` parameters.
- At the same time there is an increase in the validation accuracy when the growth of the tree is restricted. This tells us that the tree with default parameters had overfitted the train data (hence the 100% accuracy on train).
- The overall accuracy is less compared to the tree built with all the features because the features are now much less (10 instead of 3072).
- Since there are much less features to train on in this case, the training time was also much less (in hundredths of a second compared to a few seconds in 3.2a).
- The best parameters in this case were observed to be the same as 3.2a and are as follows
  1. `max_depth` = 10
  2. `min_samples_split` = 7
- The accuracy on validation set for this case was 63.25% which is not very high.

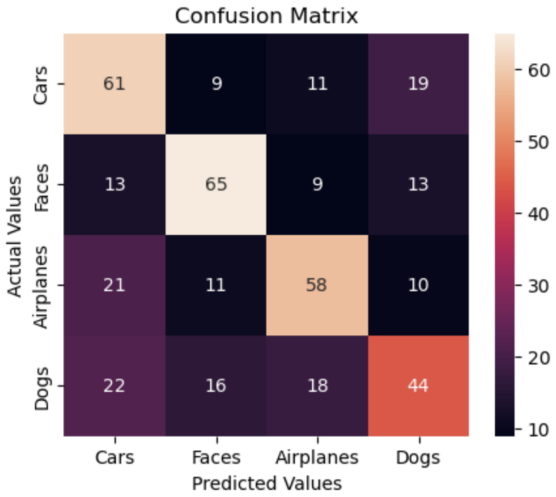
Confusion Matrix on Validation Set



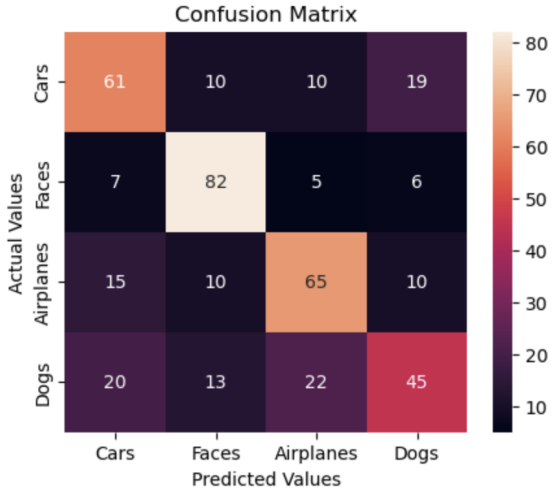
Default Parameters



max_depth = 10

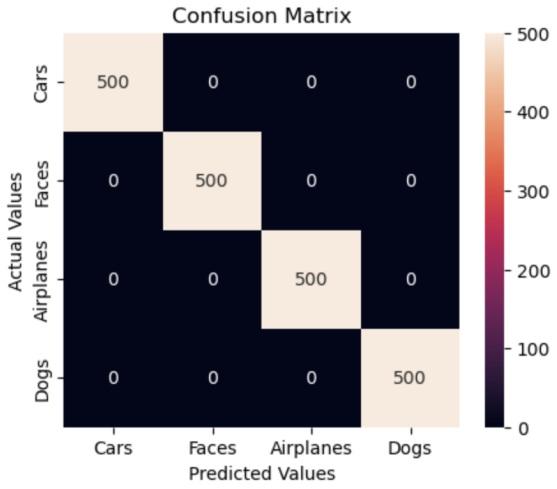


min_samples_split = 7

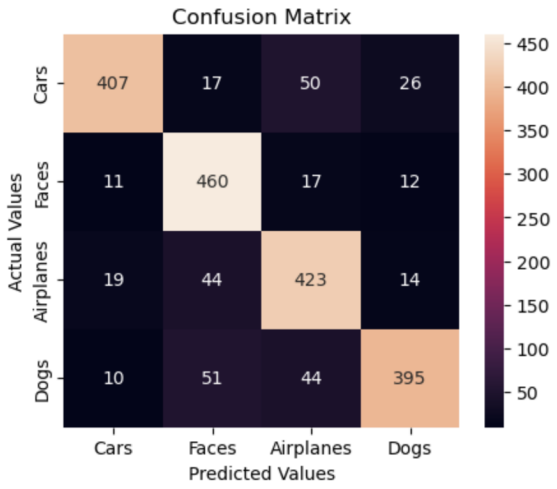


depth = 10, split = 7

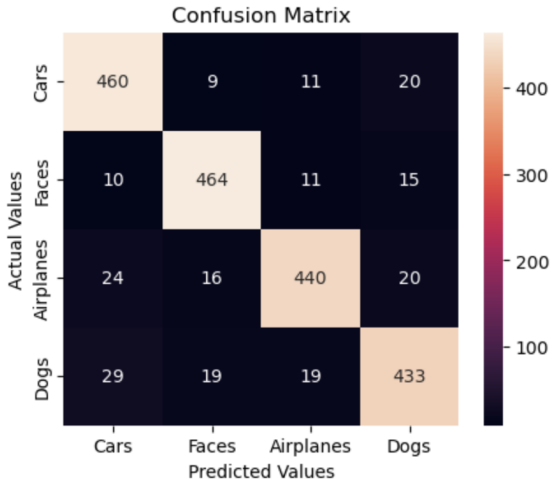
Confusion Matrix on Training Set



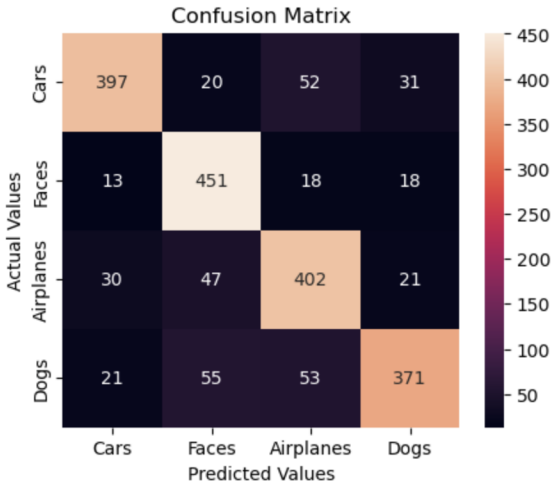
Default Parameters



max_depth = 10



min_samples_split = 7



depth = 10, split = 7

Visualizing the Tree

The trees built are very large and hence cannot be properly viewed here. The links for the visualised trees are provided below:

- Default Parameters
- Best Parameters

Grid Search

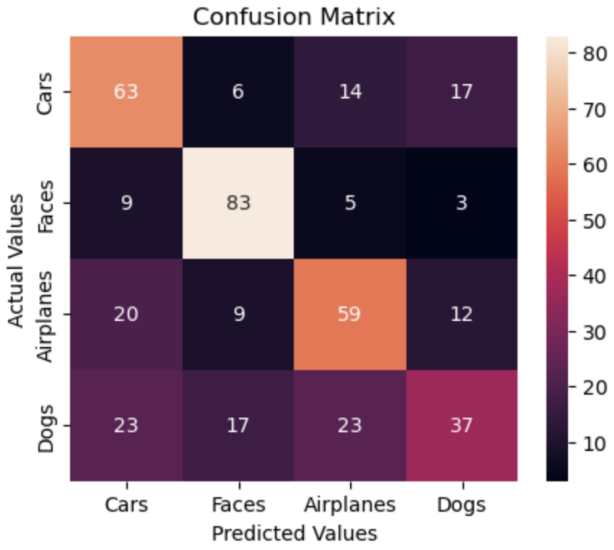
- In this part Grid Search was performed using the `GridSearchCV` implementation of `sklearn` on the decision tree classifier (with default hyper parameters).
- The best parameters came out to be
  1. criterion: entropy
  2. max_depth: 7
  3. min_samples_split = 9
- Time taken to train with best parameters was 0.04131 seconds.

Accuracy on Train	Accuracy on Validation
73.75%	60.75%

Observations

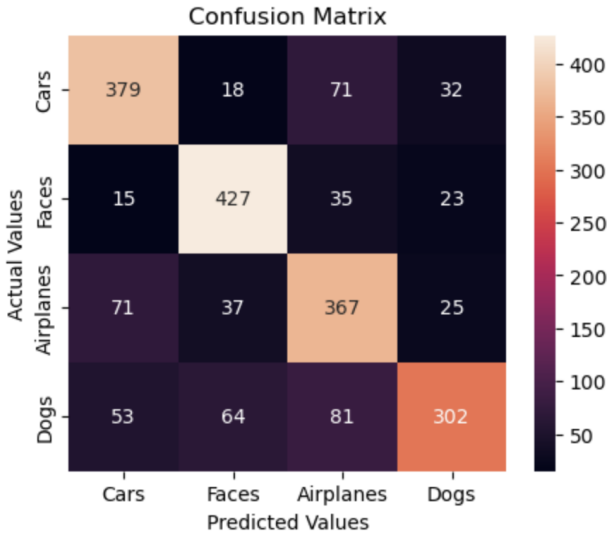
- The observations are very similar to our observations in Grid Search in the binary classifier.
- This is not very different and in fact not much better than any of the set of hyperparameters we had tried previously in b). All of them had yielded equally as good if not better results on the validation set.
- There is also a noticeable reduction in the train accuracy, telling us that overfitting has been prevented however in this scenario the tree might have underfit.
- But since Grid Search finds out the best parameters after cross validation it is possible that these are a better set than the one we got better results with while experimenting manually.
- Compared to a) where we used all the features, the accuracy is less by about 10%.

Confusion Matrix on the Validation Set



Grid Search best parameters

Confusion Matrix on the Training Set

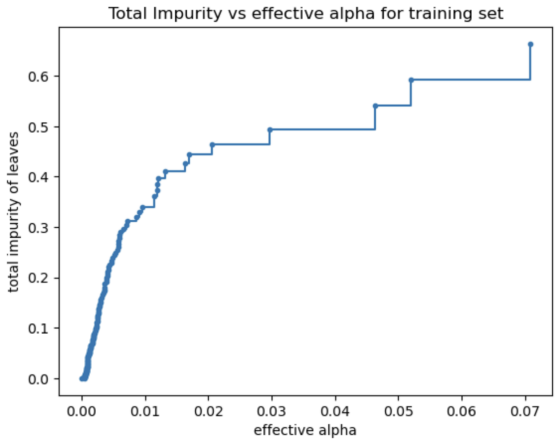


Grid Search best parameters

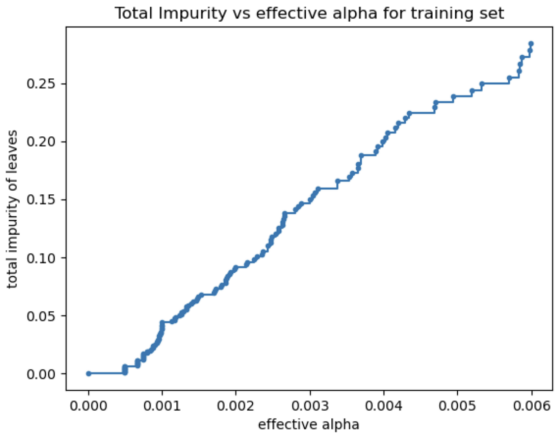
c) Decision Tree Post Pruning with Cost Complexity Pruning

- We used the `DecisionTreeClassifier.cost_complexity_pruning_path` (a part of `sklearn`) to obtain the total leaf impurities and effective alphas.
- Below are the various plots showing the variation of different parameters with the effective alphas which were asked.

Total Impurity v/s Effective Alphas

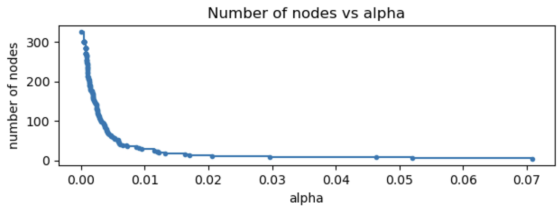


For all effective alphas

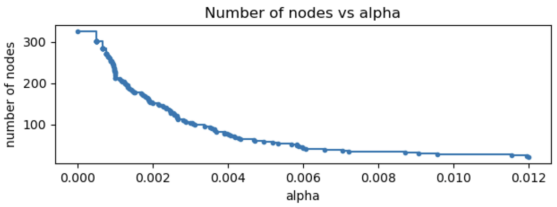


Excluding some alphas from the end

Number of Nodes v/s Effective Alphas

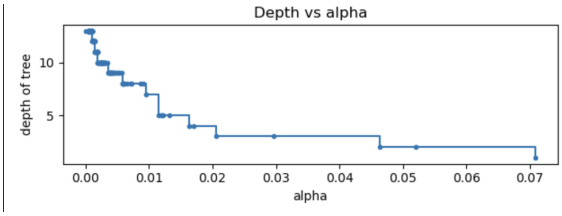


For all effective alphas

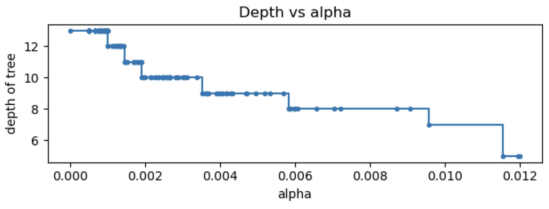


Excluding some alphas from the end

Depth v/s Effective Alphas

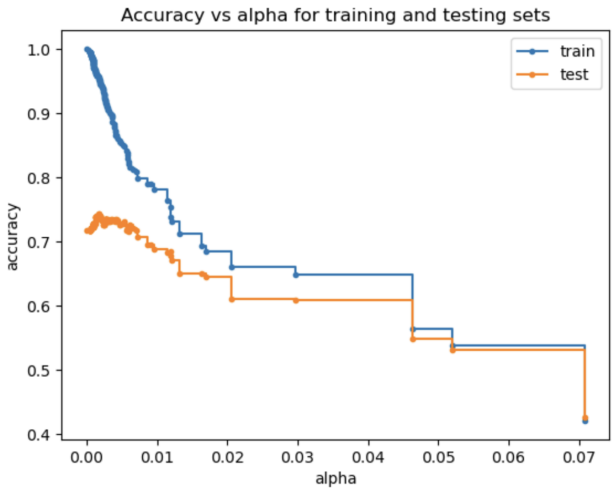


For all effective alphas



Excluding some alphas from the end

Accuracy on Train and Test v/s Alphas



For all effective alphas

Observations

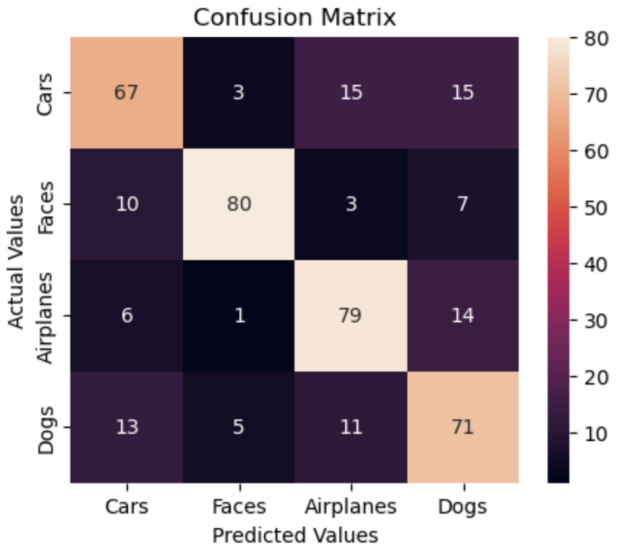
- As alpha increases, more of the tree is pruned, which increases the total impurity of the leaves as more misclassifications will occur.
- As more and more of the tree is pruned, the number of nodes reduces (pruning means removing nodes from the tree).
- As the number of nodes in the tree decreases, the depth of the tree also consequently decreases.
- As alpha increases, the validation set accuracy initially increases but soon reaches a maxima and starts decreasing.
- The maximum validation set accuracy of 74.25% is achieved at alpha = 0.0016969696969696972. The training accuracy at this alpha is 95.70%.
- The training accuracy decreases with increase in alpha. Because as the number of nodes reduces, there is a tradeoff between the training accuracy and the validation accuracy. Initially the tree has overfitted to the train set and produces almost 100% accuracy on the train set, however it performs poorly on the validation set. Upon pruning the train accuracy decreases slightly however there is a significant improvement in the validation accuracy.

Visualizing the best-pruned tree

The tree would not be clearly visible here hence I am attaching a link of the best-pruned tree that was obtained in this part.

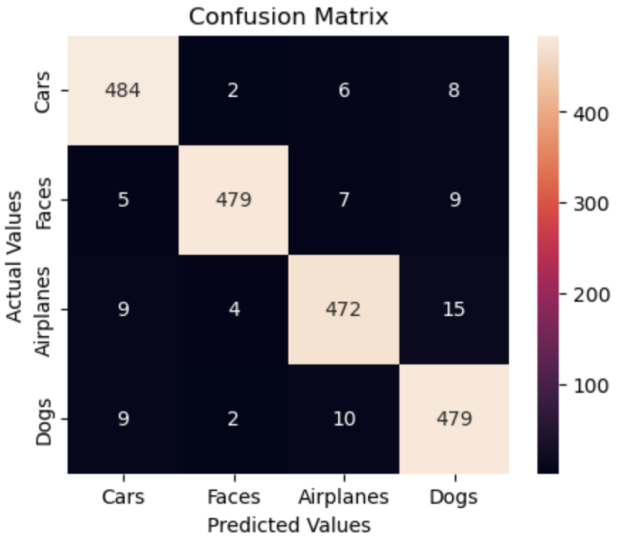
- Best Pruned Tree

Confusion Matrix on the Validation Set



Grid Search best parameters

Confusion Matrix on the Training Set



Grid Search best parameters

d) Random Forest

- We have used the `RandomForestClassifier` from `sklearn` in this part to use the **Random Forest** Classifier.
- The accuracy, precision and recall on the train and validation sets for default parameters is shown below.

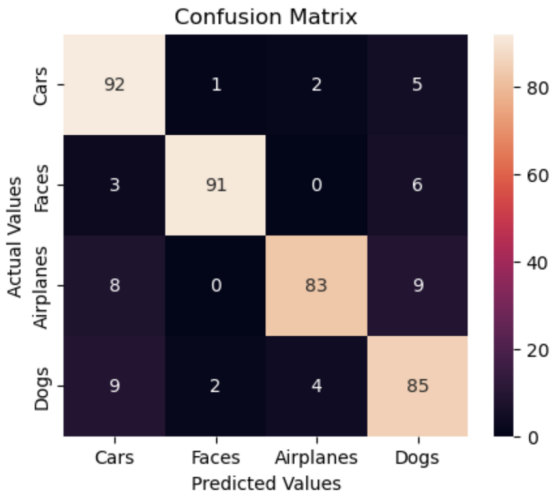
Accuracy on Train	Accuracy on Val
100%	87.75%

Grid Search

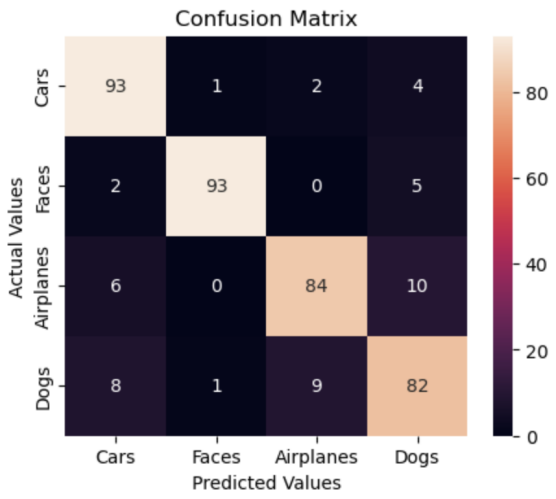
- In this part Grid Search was performed using the `GridSearchCV` implementation of `sklearn` on the Random Forest classifier (with default hyper parameters).
- The best parameters came out to be
  1. criterion: entropy
  2. max_depth: 10
  3. min_samples_split = 7
  4. n_estimators = 150
- The accuracy on the train and validation sets with the best set of hyperparameters is as shown below.

Accuracy on Train	Accuracy on Val
100%	88.0%

Confusion Matrix on Validation Set

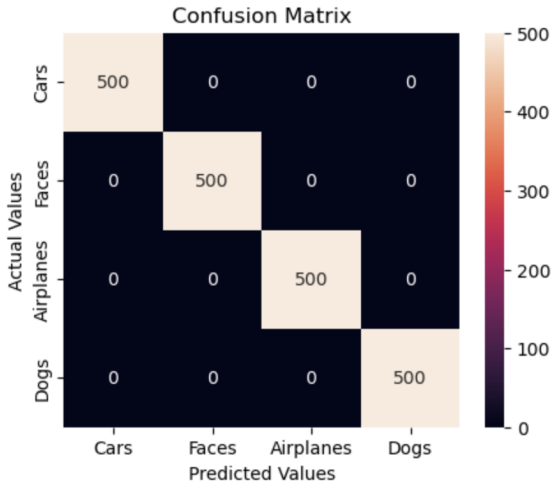


Default Parameters

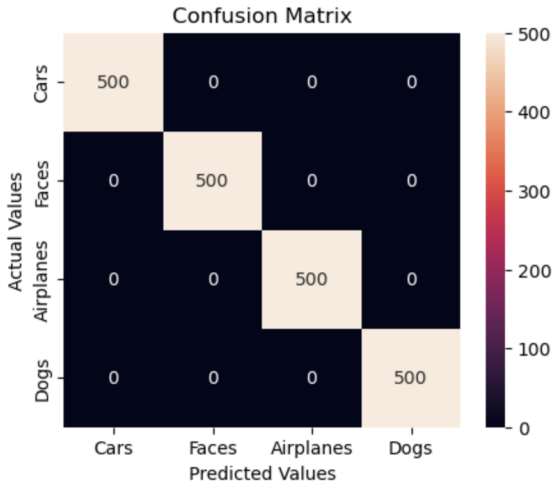


Best Parameters

Confusion Matrix on Training Set



Default Parameters



Best Parameters

e) Gradient Boosted Trees and XGBoost

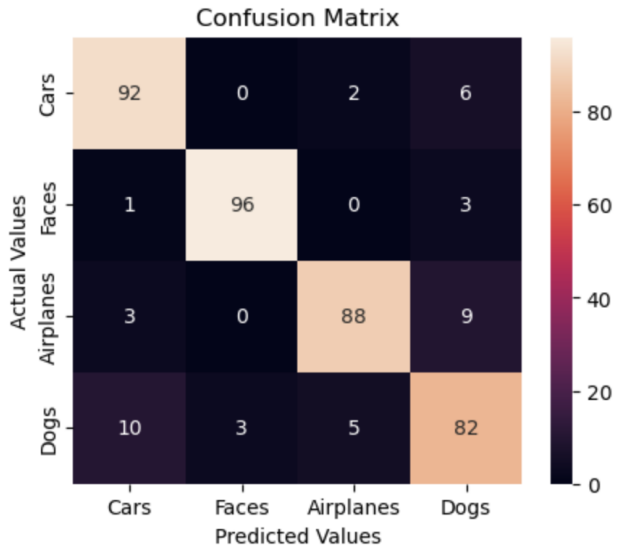
Gradient Boosted Trees

- Grid Search was performed on the given set of parameters.
- The best parameters came out to be
  1. n_estimators: 50
  2. max_depth: 9
  3. subsample : 0.5
- The training time and the accuracy, precision and recall values in the training and validation sets is shown in the table below.
- The time taken to train is the time taken to perform Grid Search on the given sets of parameters.

Train Time	Train: Accuracy	Val: Accuracy
5h 34min	100%	89.5%

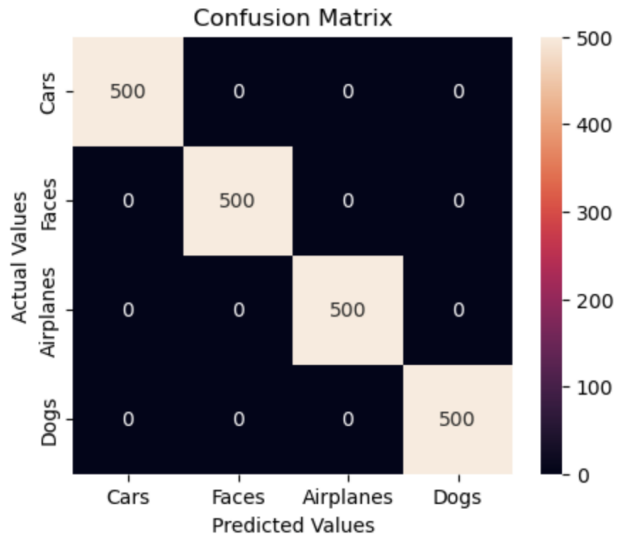
- We observe very high accuracy on the validation set (approximately 89.5%).
- The accuracy is slightly more than that obtained in the case of Random Forest Classifier even with its best set of parameters.

Confusion Matrix on the Validation Set



Grid Search best parameters

Confusion Matrix on the Training Set



Grid Search best parameters

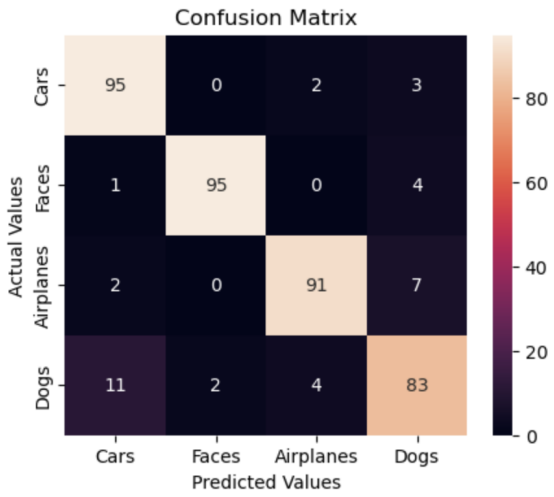
XGBoost

- In this part Grid Search was performed using the `GridSearchCV` implementation of `sklearn` on the XGBoost Classifier.
- The best parameters came out to be
  1. n_estimators: 50
  2. max_depth: 9
  3. subsample : 0.6
- The accuracy values for the training and validation sets is shown in the table below.

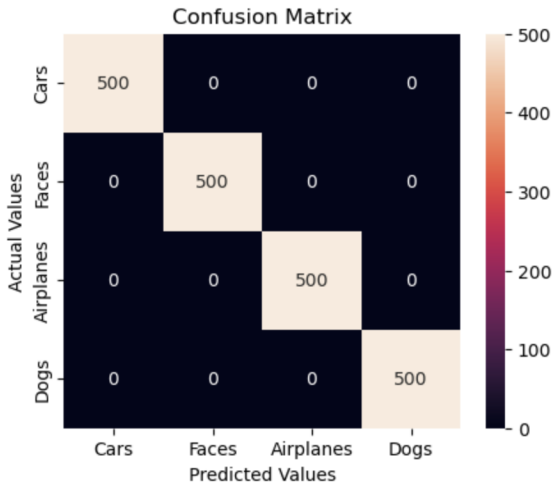
Train Accuracy	Validation Accuracy
100%	91%

- We observe that the accuracy on the validation set is more in this case compared to the Gradient Boost Classifier.
- The accuracy is also better than the Random Forest Classifier with its best parameters.

Confusion Matrix on Validation Set and Training Set



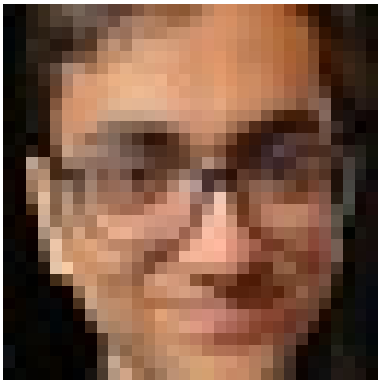
Validation Set



Training Set

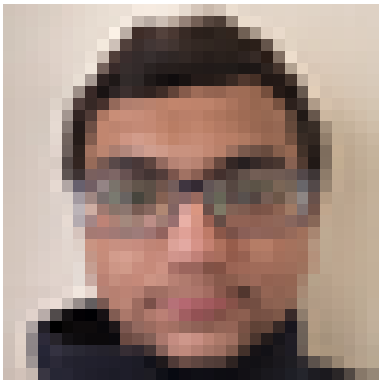
g) Real Time Application

- I tested various models and the `GradientBoostClassifier` produced the best results for me.
- I tested on 10 photos and obtained an accuracy of 50%.
- Upon observing the train data, I realised that most of the training images were of white people and since my complexion is not that fair, the models were having a tough time trying to distinguish me between a dog and a person.
- This is because most of the images of dogs were darker in colour.
- Below are some of the images and the results obtained:



Predicted Person

Predicted Dog

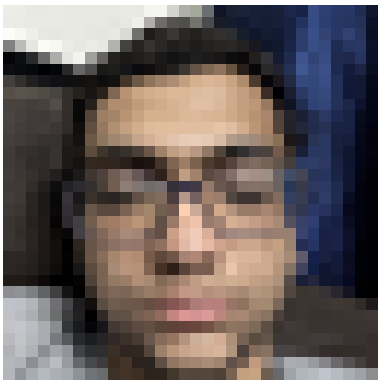


Predicted Car

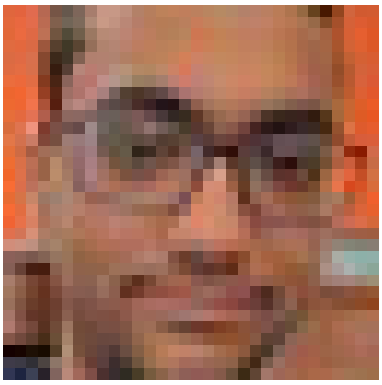
Predicted Dog

Predicted Person

Predicted Dog



Predicted Person



Predicted Dog

Predicted Person

- I am wearing specs in all the images so that could also have led to some misclassifications.
- I tested on images of a few friends and obtained similar results.
- This is not a very good set of images and the model has been trained on very specific looking images with the face of the person covering the entire image.
- The images in which my face was at an angle were misclassified.