Assignment 1

Akshat Chaudhary (2021MT60814) Aniket Pandey (2021MT60266)
Aditya Arya (2021MT60958)

March 16, 2024

1 Preprocessing of Categorical Data and its effects

First, we do not process the categorical variable ourselves and see that Tensorflow handles the categorical data itself and ends up with an accuracy of 86.6%. The OOB evaluation of the model is attached below:

```
Training OOB:
        trees: 1, Out-of-bag evaluation: accuracy:0.796703 logloss:7.32756
        trees: 11, Out-of-bag evaluation: accuracy:0.823293 logloss:2.31055
        trees: 21, Out-of-bag evaluation: accuracy:0.828 logloss:1.13807
        trees: 31, Out-of-bag evaluation: accuracy:0.848 logloss:0.791936
        trees: 41, Out-of-bag evaluation: accuracy:0.852 logloss:0.589944
        trees: 51, Out-of-bag evaluation: accuracy:0.852 logloss:0.589535
        trees: 61, Out-of-bag evaluation: accuracy:0.848 logloss:0.527496
        trees: 71, Out-of-bag evaluation: accuracy:0.852 logloss:0.53131
        trees: 81, Out-of-bag evaluation: accuracy:0.85 logloss:0.397766
        trees: 91, Out-of-bag evaluation: accuracy:0.842 logloss:0.397254
        trees: 101, Out-of-bag evaluation: accuracy:0.844 logloss:0.398846
        trees: 111, Out-of-bag evaluation: accuracy:0.856 logloss:0.397642
        trees: 121, Out-of-bag evaluation: accuracy:0.85 logloss:0.394981
        trees: 131, Out-of-bag evaluation: accuracy:0.85 logloss:0.32862
        trees: 141, Out-of-bag evaluation: accuracy:0.85 logloss:0.32981
        trees: 151, Out-of-bag evaluation: accuracy:0.856 logloss:0.328501
        trees: 161, Out-of-bag evaluation: accuracy:0.86 logloss:0.327844
        trees: 171, Out-of-bag evaluation: accuracy:0.856 logloss:0.326293
        trees: 181, Out-of-bag evaluation: accuracy:0.854 logloss:0.324163
        trees: 191, Out-of-bag evaluation: accuracy:0.856 logloss:0.322485
        trees: 201, Out-of-bag evaluation: accuracy:0.864 logloss:0.321333
        trees: 211, Out-of-bag evaluation: accuracy:0.862 logloss:0.322125
        trees: 221, Out-of-bag evaluation: accuracy:0.864 logloss:0.322272
        trees: 231, Out-of-bag evaluation: accuracy:0.856 logloss:0.322001
        trees: 241, Out-of-bag evaluation: accuracy:0.862 logloss:0.320564
        trees: 251, Out-of-bag evaluation: accuracy:0.862 logloss:0.318527
        trees: 261, Out-of-bag evaluation: accuracy:0.864 logloss:0.318395
        trees: 271, Out-of-bag evaluation: accuracy:0.866 logloss:0.318744
        trees: 281, Out-of-bag evaluation: accuracy:0.866 logloss:0.31836
        trees: 291, Out-of-bag evaluation: accuracy:0.866 logloss:0.31756
        trees: 300, Out-of-bag evaluation: accuracy:0.866 logloss:0.316497
```

Now we compare it with the model trained on encoded data to understand if this would change the accuracy of the final model. With this, we get an accuracy of 86.2%. The OOB evaluation of this new model is attached here:

```
Training OOB:
        trees: 1, Out-of-bag evaluation: accuracy:0.824176 logloss:6.33735
        trees: 11, Out-of-bag evaluation: accuracy:0.829317 logloss:2.30354
        trees: 21, Out-of-bag evaluation: accuracy:0.832 logloss:1.06849
        trees: 31, Out-of-bag evaluation: accuracy:0.846 logloss:0.787187
        trees: 41, Out-of-bag evaluation: accuracy:0.854 logloss:0.586656
        trees: 51, Out-of-bag evaluation: accuracy:0.848 logloss:0.585979
        trees: 61, Out-of-bag evaluation: accuracy:0.846 logloss:0.523646
        trees: 71, Out-of-bag evaluation: accuracy:0.848 logloss:0.528567
        trees: 81, Out-of-bag evaluation: accuracy:0.848 logloss:0.395
        trees: 91, Out-of-bag evaluation: accuracy:0.842 logloss:0.394203
        trees: 101, Out-of-bag evaluation: accuracy:0.846 logloss:0.396517
        trees: 111, Out-of-bag evaluation: accuracy:0.854 logloss:0.395143
        trees: 121, Out-of-bag evaluation: accuracy:0.852 logloss:0.392414
        trees: 131, Out-of-bag evaluation: accuracy:0.856 logloss:0.326282
        trees: 141, Out-of-bag evaluation: accuracy:0.85 logloss:0.327723
        trees: 151, Out-of-bag evaluation: accuracy:0.854 logloss:0.326511
        trees: 161, Out-of-bag evaluation: accuracy:0.86 logloss:0.324383
        trees: 171, Out-of-bag evaluation: accuracy:0.862 logloss:0.324054
        trees: 181, Out-of-bag evaluation: accuracy:0.86 logloss:0.321983
        trees: 191, Out-of-bag evaluation: accuracy:0.856 logloss:0.320452
        trees: 201, Out-of-bag evaluation: accuracy:0.866 logloss:0.319077
        trees: 211, Out-of-bag evaluation: accuracy:0.866 logloss:0.31982
        trees: 221, Out-of-bag evaluation: accuracy:0.858 logloss:0.319676
        trees: 231, Out-of-bag evaluation: accuracy:0.858 logloss:0.318936
        trees: 241, Out-of-bag evaluation: accuracy:0.866 logloss:0.317153
        trees: 251, Out-of-bag evaluation: accuracy:0.864 logloss:0.315538
        trees: 261, Out-of-bag evaluation: accuracy:0.866 logloss:0.316007
        trees: 271, Out-of-bag evaluation: accuracy:0.868 logloss:0.315714
        trees: 281, Out-of-bag evaluation: accuracy:0.866 logloss:0.315192
        trees: 291, Out-of-bag evaluation: accuracy:0.862 logloss:0.314472
        trees: 300, Out-of-bag evaluation: accuracy:0.862 logloss:0.313398
```

Such close values in the accuracy's imply that TensorFlow successfully handles categorical and ordinal data by itself.

2 Splitting of the dataset

The dataset is split into a training set and a test set. The training set consists of 70% of the original dataset while the test set consists of 30%, as shown below:

	Lab-Test1(30)	Lab-Test2(24)	Midsem Test (90)	Gender	Attendance	Grade
5	8.00	24	12.0	Male	High	D
116	16.00	24	51.0	Male	High	B
45	4.25	0	26.0	Female	High	D
16	0.00	24	40.0	Female	High	C
462	3.75	24	23.0	Female	High	C-
..	...	...	...	...	...	...
106	2.25	24	42.0	Male	High	C
270	16.00	24	38.0	Female	Moderate	B-
348	7.25	24	39.0	Female	Low	C
435	5.50	24	37.0	Female	High	C
102	3.25	22	28.0	Male	High	C-

[350 rows x 6 columns]

	Lab-Test1(30)	Lab-Test2(24)	Midsem Test (90)	Gender	Attendance	Grade
361	6.25	24	55.0	Male	High	B
73	12.00	24	53.0	Female	High	B
374	13.25	24	68.0	Female	Moderate	A
155	12.50	20	0.0	Male	High	D
104	19.00	24	60.0	Male	High	A
..	...	...	...	...	...	...
266	1.25	24	42.0	Female	High	C
23	5.25	24	46.0	Female	Moderate	B-
222	9.75	24	38.0	Female	High	B-
261	14.50	24	59.0	Female	High	A-
426	5.25	23	37.0	Female	High	C

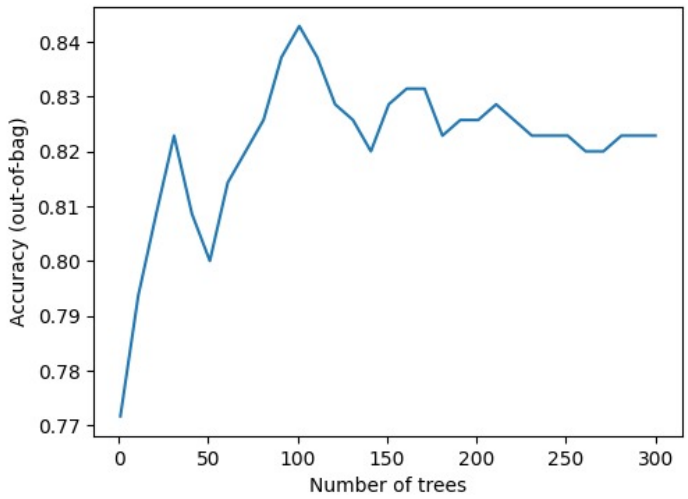
[150 rows x 6 columns]

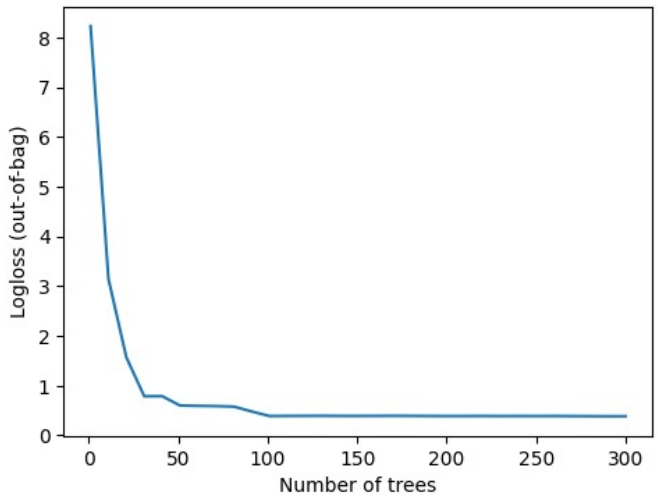
After fitting the model with this training set, we test the accuracy of this model on the test set. We got an accuracy of 86.67%.

```
Compiling model...
Model compiled.
1/1 [==============================] - 0s 481ms/step - loss: 0.0000e+00 - accuracy: 0.8667

loss: 0.0000
accuracy: 0.8667
```

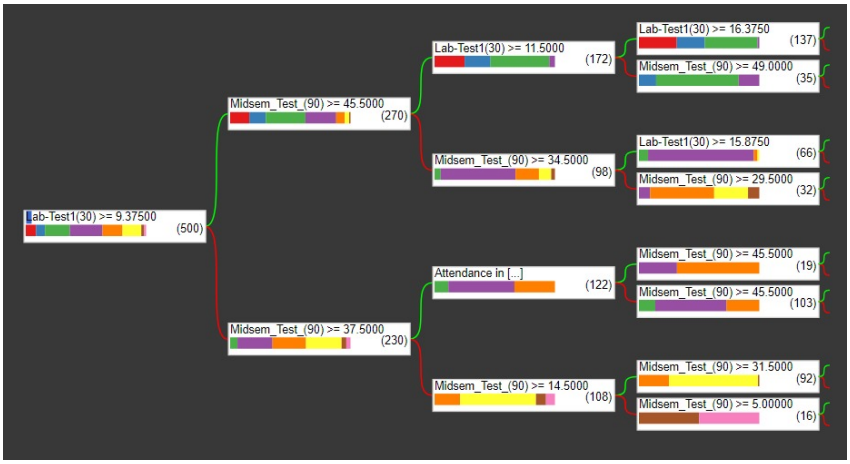
Visualizing the accuracy and log loss vs the number of trees, we get these graphs:





3 Visualisation of the Forest

After using the `plotter` function of the `tensorflow` library we get the following visualisation:



4 Gradient Boosted Trees

We train a gradient boosted tree on the particular training set. We get an accuracy of 85.33% on the test set.

```
Use /tmp/tmprqqlij54 as temporary training directory
Reading training dataset...
Training dataset read in 0:00:00.562699. Found 350 examples.
Training model...
Model trained in 0:00:00.709104
Compiling model...
Model compiled.
1/1 [==============================] - 0s 190ms/step - loss: 0.0000e+00 - accuracy: 0.8533
```

We compare this accuracy with that of a decision forest of 30 trees. In the latter case, we get an accuracy of 86%.

```
Use /tmp/tmpg43_aep3 as temporary training directory
Reading training dataset...
WARNING:tensorflow:5 out of the last 5 calls to <function CoreModel._consumes_training_examples_until_eof at 0x7a6d
Training dataset read in 0:00:00.234159. Found 350 examples.
Training model...
Model trained in 0:00:00.057799
Compiling model...
WARNING:tensorflow:5 out of the last 5 calls to <function InferenceCoreModel.make_predict_function.<locals>.predict
Model compiled.
WARNING:tensorflow:5 out of the last 7 calls to <function InferenceCoreModel.yggdrasil_model_path_tensor at 0x7a6dd
1/1 [==============================] - 0s 192ms/step - loss: 0.0000e+00 - accuracy: 0.8600
```

The accuracy's and log loss for Random Forest have already been plotted above. We see that both the models give us approximately the same accuracies.

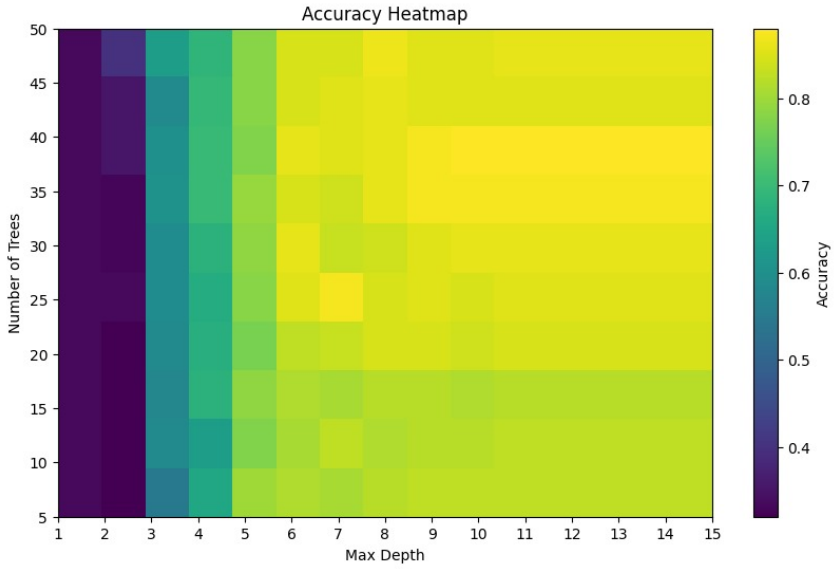
5 Analysis of Random Forests Hyperparameters and Accuracies

First, we compare testing and training accuracy for Decision trees. We got the accuracies below.

```
Use /tmp/tmp3tol82v4 as temporary training directory
Reading training dataset...
WARNING:tensorflow:6 out of the last 6 calls to <function CoreModel._consumes_training_example
Training dataset read in 0:00:00.409677. Found 350 examples.
Training model...
Model trained in 0:00:00.506867
Compiling model...
WARNING:tensorflow:6 out of the last 6 calls to <function InferenceCoreModel.make_predict_func
Model compiled.
WARNING:tensorflow:6 out of the last 8 calls to <function InferenceCoreModel.yggdrasil_model_p
1/1 [==============================] - 0s 251ms/step - loss: 0.0000e+00 - accuracy: 0.8667

WARNING:tensorflow:5 out of the last 5 calls to <function InferenceCoreModel.make_test_functio
1/1 [==============================] - 0s 192ms/step - loss: 0.0000e+00 - accuracy: 0.9400
```

As expected, we get 94% accuracy on the training set and slightly less accuracy of 86.67% on the test set. For the hyper parameters, we make a heatmap of test accuracy on the basis of the number of trees and max depth of trees. We obtain the following heatmap:



From this heatmap, it is easy to see that the optimal parameters for the accuracy above 85% are the number of trees = 25 and the maximum depth = 7. For a maximum depth of 7, we see that increasing or decreasing the number of trees from 25 decreases the accuracy on the test set. In the case of the number of trees = 25, we see that increasing the max depth makes the accuracy very slightly. Hence 25 trees and their max depth being 7 is optimal for our model. In this case, we get an accuracy of 87.33%.

```
Use /tmp/tmpnefp0q24 as temporary training directory
Reading training dataset...
Training dataset read in 0:00:00.934388. Found 350 examples.
Training model...
Model trained in 0:00:00.096906
Compiling model...
Model compiled.
1/1 [==============================] - 0s 289ms/step - loss: 0.0000e+00 - accuracy: 0.8733
```