HOMEWORK 1

DESCRIPTION OF TASKS

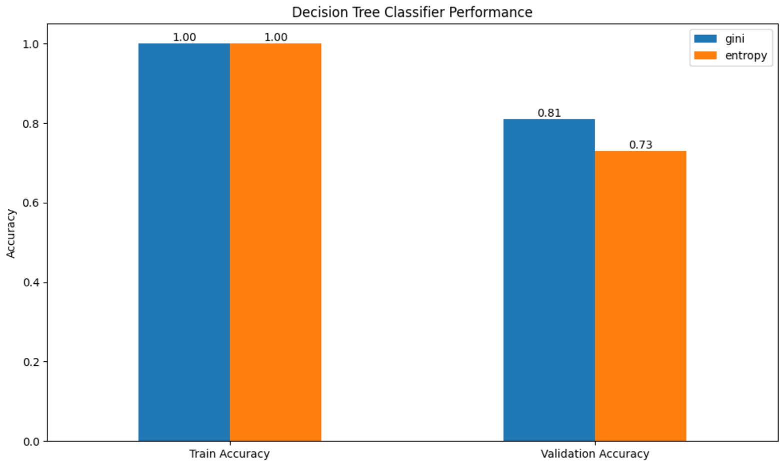
1. **Pre-processing –** For pre-processing, I have used the code from HW0 solutions. This includes converting the characters into lowercase, removing punctuation, tokenization, filtering out stop words and stemming.

   Later, the processed text are vectorized using the TfidVectorizer and ngrams(1,2) which includes unigrams and bigrams.

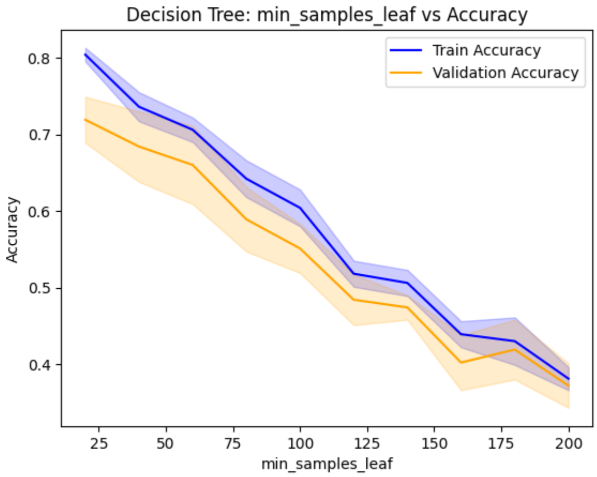
2.1 For this question, the data is split into 80% and 20% as Test data and validation data. 2 criteria namely, Entropy and Gini are considered and both of them used for Decision Tree Classifier. The training accuracy and the validation accuracy are as follows:

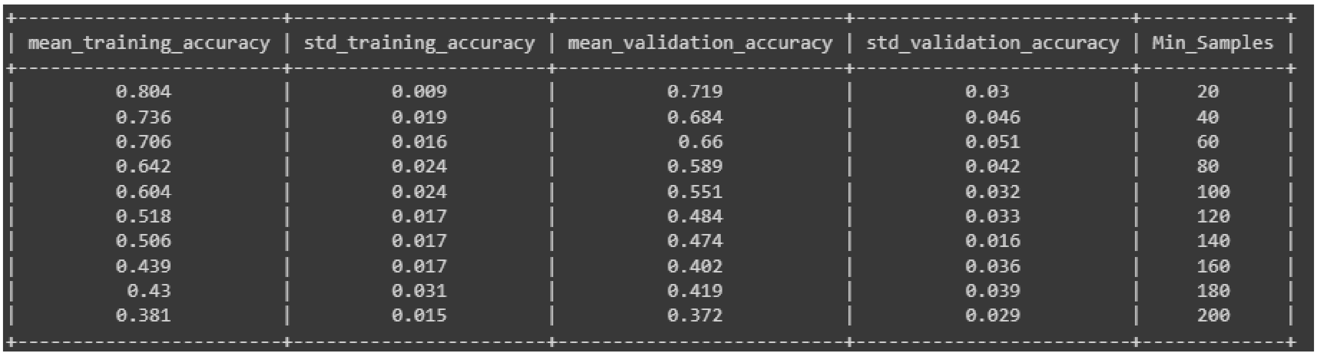
{'gini': {'Training': 1.0, 'Validation': 0.81}, 'entropy': {'Training': 1.0, 'Validation': 0.73}}

The bar chart that we get as the output is pasted below:



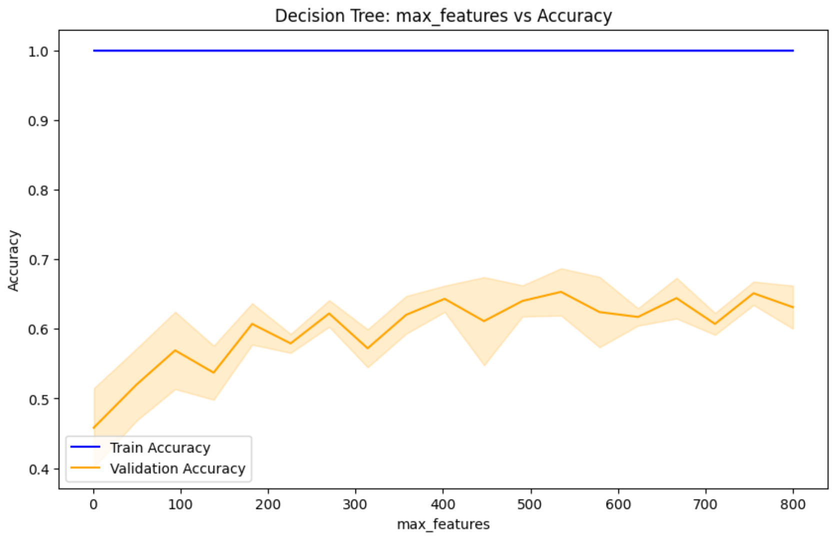
2.2.2 Min Samples vs Accuracy

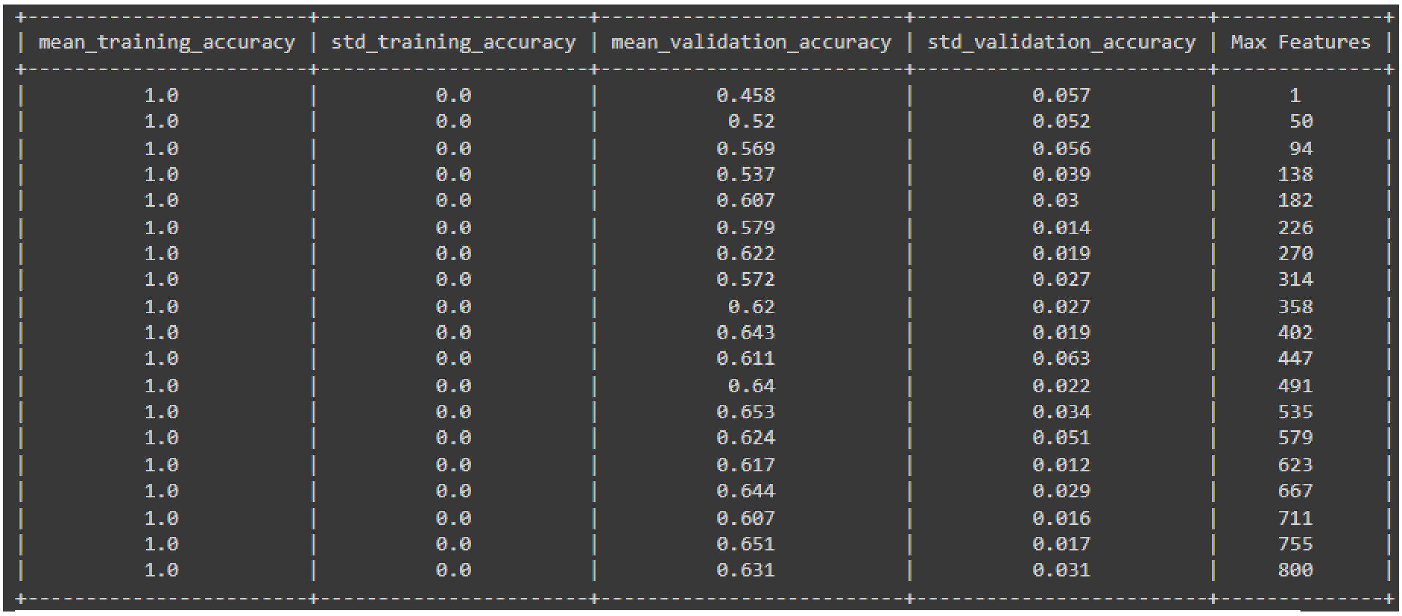




```
+------------------------+-----------------------+--------------------------+-------------------------+-------------+
| mean_training_accuracy | std_training_accuracy | mean_validation_accuracy | std_validation_accuracy | Min_Samples |
+------------------------+-----------------------+--------------------------+-------------------------+-------------+
|         0.804          |         0.009         |          0.719           |           0.03          |      20     |
|         0.736          |         0.019         |          0.684           |          0.046          |      40     |
|         0.706          |         0.016         |           0.66           |          0.051          |      60     |
|         0.642          |         0.024         |          0.589           |          0.042          |      80     |
|         0.604          |         0.024         |          0.551           |          0.032          |     100     |
|         0.518          |         0.017         |          0.484           |          0.033          |     120     |
|         0.506          |         0.017         |          0.474           |          0.016          |     140     |
|         0.439          |         0.017         |          0.402           |          0.036          |     160     |
|          0.43          |         0.031         |          0.419           |          0.039          |     180     |
|         0.381          |         0.015         |          0.372           |          0.029          |     200     |
+------------------------+-----------------------+--------------------------+-------------------------+-------------+
```

2.2.3 Max features Vs Accuracy





```
+------------------------+-----------------------+--------------------------+-------------------------+--------------+
| mean_training_accuracy | std_training_accuracy | mean_validation_accuracy | std_validation_accuracy | Max Features |
+------------------------+-----------------------+--------------------------+-------------------------+--------------+
|          1.0           |          0.0          |          0.458           |          0.057          |      1       |
|          1.0           |          0.0          |           0.52           |          0.052          |      50      |
|          1.0           |          0.0          |          0.569           |          0.056          |      94      |
|          1.0           |          0.0          |          0.537           |          0.039          |     138      |
|          1.0           |          0.0          |          0.607           |          0.03           |     182      |
|          1.0           |          0.0          |          0.579           |          0.014          |     226      |
|          1.0           |          0.0          |          0.622           |          0.019          |     270      |
|          1.0           |          0.0          |          0.572           |          0.027          |     314      |
|          1.0           |          0.0          |           0.62           |          0.027          |     358      |
|          1.0           |          0.0          |          0.643           |          0.019          |     402      |
|          1.0           |          0.0          |          0.611           |          0.063          |     447      |
|          1.0           |          0.0          |           0.64           |          0.022          |     491      |
|          1.0           |          0.0          |          0.653           |          0.034          |     535      |
|          1.0           |          0.0          |          0.624           |          0.051          |     579      |
|          1.0           |          0.0          |          0.617           |          0.012          |     623      |
|          1.0           |          0.0          |          0.644           |          0.029          |     667      |
|          1.0           |          0.0          |          0.607           |          0.016          |     711      |
|          1.0           |          0.0          |          0.651           |          0.017          |     755      |
|          1.0           |          0.0          |          0.631           |          0.031          |     800      |
+------------------------+-----------------------+--------------------------+-------------------------+--------------+
```

3.1 Parameters for Random Forest –

The parameters I used for Random Forest are as follows -

`criterion`: Gini

random_state= 42

`min_samples_split`: 2

n_splits = 5 (5 cross validation)

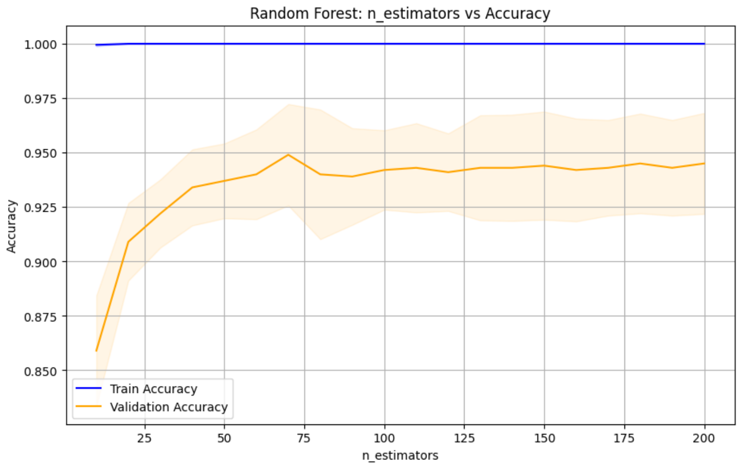
`n_estimators`: Values from 10 to 200, stepping by 10

`min_samples_leaf`: 1

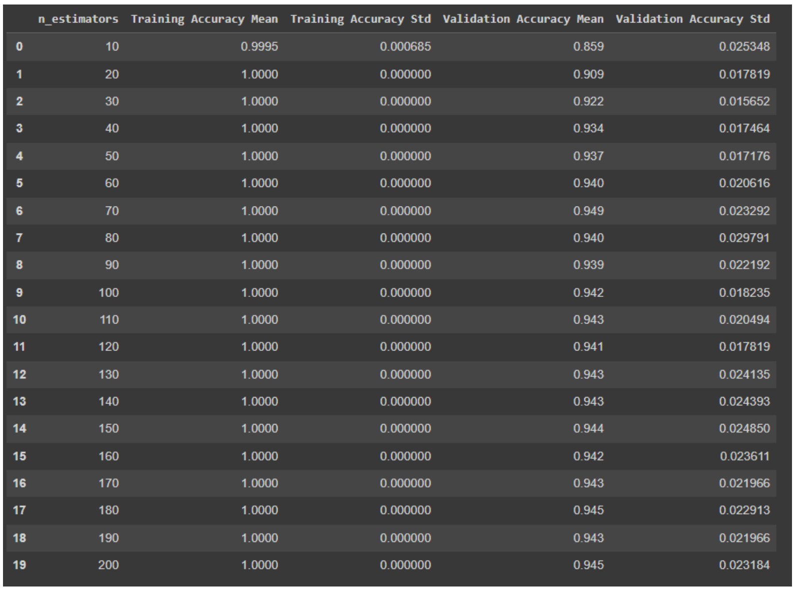
shuffle = True

3.2 plotting the graph and the table for accuracy:

The graph for Random forest with n_estimators:

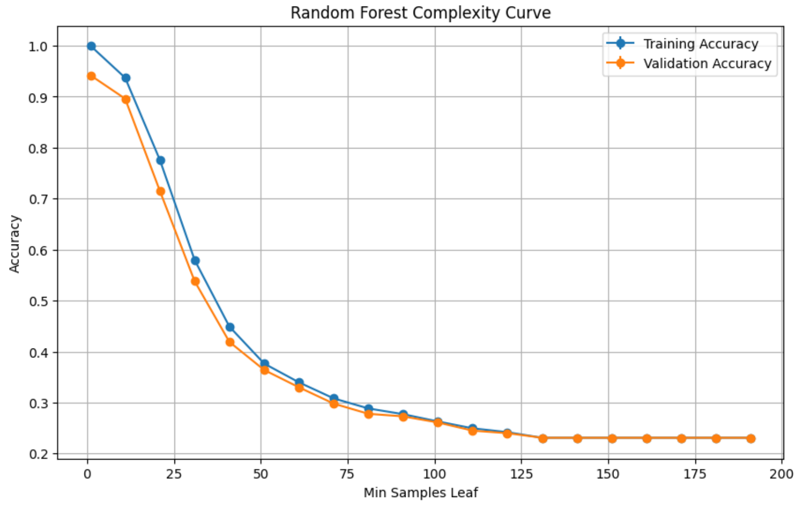


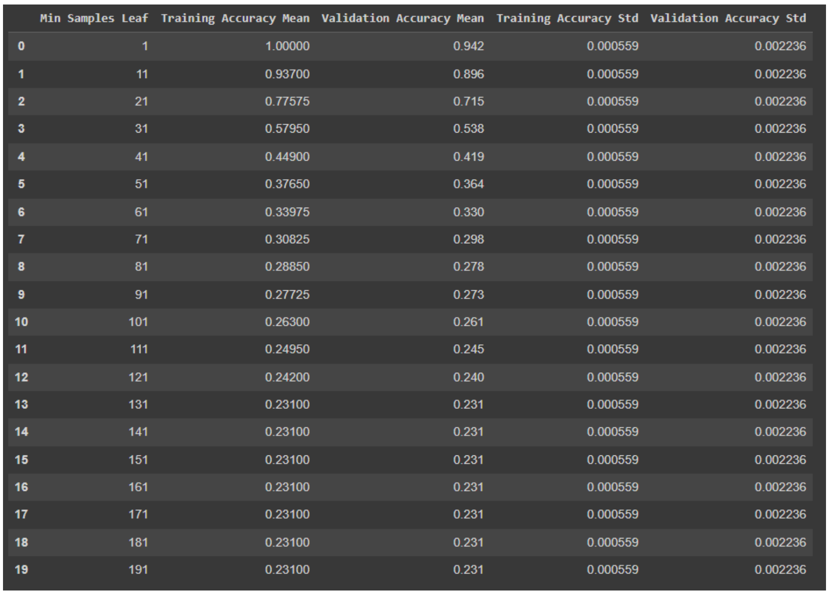
The accuracy for different n_estimators are as follow:



	n_estimators	Training Accuracy Mean	Training Accuracy Std	Validation Accuracy Mean	Validation Accuracy Std
0	10	0.9995	0.000685	0.859	0.025348
1	20	1.0000	0.000000	0.909	0.017819
2	30	1.0000	0.000000	0.922	0.015652
3	40	1.0000	0.000000	0.934	0.017464
4	50	1.0000	0.000000	0.937	0.017176
5	60	1.0000	0.000000	0.940	0.020616
6	70	1.0000	0.000000	0.949	0.023292
7	80	1.0000	0.000000	0.940	0.029791
8	90	1.0000	0.000000	0.939	0.022192
9	100	1.0000	0.000000	0.942	0.018235
10	110	1.0000	0.000000	0.943	0.020494
11	120	1.0000	0.000000	0.941	0.017819
12	130	1.0000	0.000000	0.943	0.024135
13	140	1.0000	0.000000	0.943	0.024393
14	150	1.0000	0.000000	0.944	0.024850
15	160	1.0000	0.000000	0.942	0.023611
16	170	1.0000	0.000000	0.943	0.021966
17	180	1.0000	0.000000	0.945	0.022913
18	190	1.0000	0.000000	0.943	0.021966
19	200	1.0000	0.000000	0.945	0.023184

3.3





	Min Samples Leaf	Training Accuracy Mean	Validation Accuracy Mean	Training Accuracy Std	Validation Accuracy Std
0	1	1.00000	0.942	0.000559	0.002236
1	11	0.93700	0.896	0.000559	0.002236
2	21	0.77575	0.715	0.000559	0.002236
3	31	0.57950	0.538	0.000559	0.002236
4	41	0.44900	0.419	0.000559	0.002236
5	51	0.37650	0.364	0.000559	0.002236
6	61	0.33975	0.330	0.000559	0.002236
7	71	0.30825	0.298	0.000559	0.002236
8	81	0.28850	0.278	0.000559	0.002236
9	91	0.27725	0.273	0.000559	0.002236
10	101	0.26300	0.261	0.000559	0.002236
11	111	0.24950	0.245	0.000559	0.002236
12	121	0.24200	0.240	0.000559	0.002236
13	131	0.23100	0.231	0.000559	0.002236
14	141	0.23100	0.231	0.000559	0.002236
15	151	0.23100	0.231	0.000559	0.002236
16	161	0.23100	0.231	0.000559	0.002236
17	171	0.23100	0.231	0.000559	0.002236
18	181	0.23100	0.231	0.000559	0.002236
19	191	0.23100	0.231	0.000559	0.002236

4.1 Pre-processing the Data to Generate Features

For pre-processing the data, the code follows these steps:

1. Lowercasing: Convert all the text in the dataset to lowercase to ensure consistency, as the case of letters can affect text analysis.
2. Removing Punctuation: Punctuation marks are removed from the text because they're often not useful for text classification tasks.
3. Tokenization: The text is split into individual words or tokens. This process helps in analyzing the text on a word-by-word basis.
4. Removing Stop Words: Stop words (commonly used words that are unlikely to be useful for predictions, such as "the", "is", "in", etc.) are removed to focus on more meaningful words.
5. Stemming: Words are reduced to their base or root form. This helps in treating different forms of a word as the same item (e.g., "running" becomes "run").
   Furthermore, the processed text are vectorized using the TfidVectorizer and ngrams(1,2) which includes unigrams and bigrams.

4.2 The choice of model and parameters is guided by the performance of models with different configurations during validation. As mentioned in the documentation, I have run the code on these models - decision trees (Mean Accuracy= 0.8090), random forests(Mean Accuracy = 0.9350). Based on the accuracy score for each of these model, The best model is picked and its Mean Accuracy score is – 0.935 with n_estimators = 100, random_state = 18.

Based on the accuracy of each of these models, The best model is chosen I have used it to predict the labels for the test data.

4.3 The chosen model is Random forest Classifier as this model gets the highest accuracy score of approximately 94%. The parameters are –

criterion='gini',

random_state=42,

n_estimators = 100