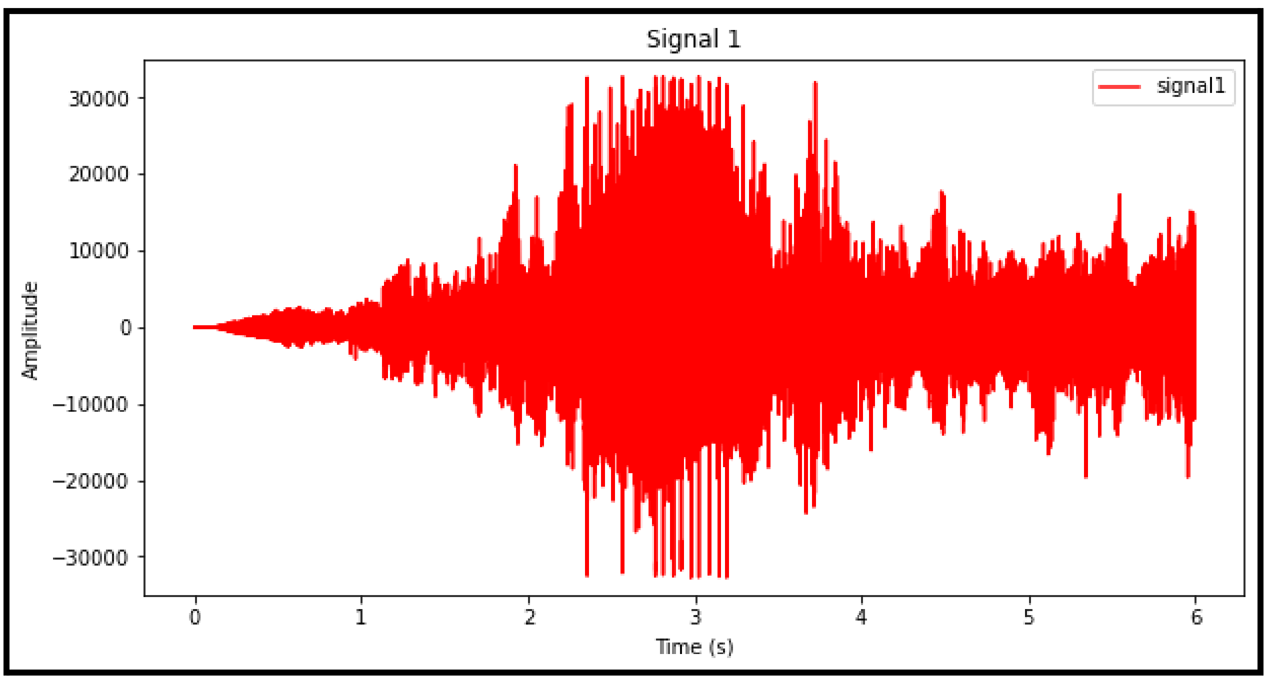
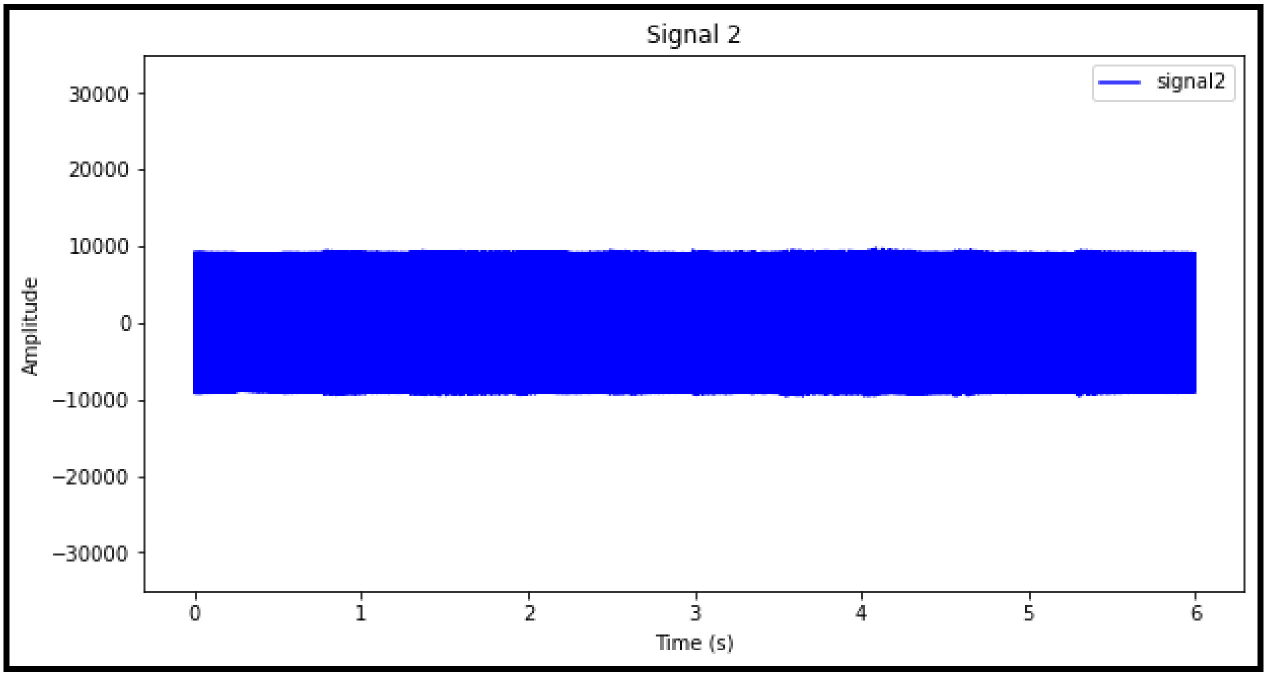
Pattern Recognition and Machine Learning 2022 Winter Semester

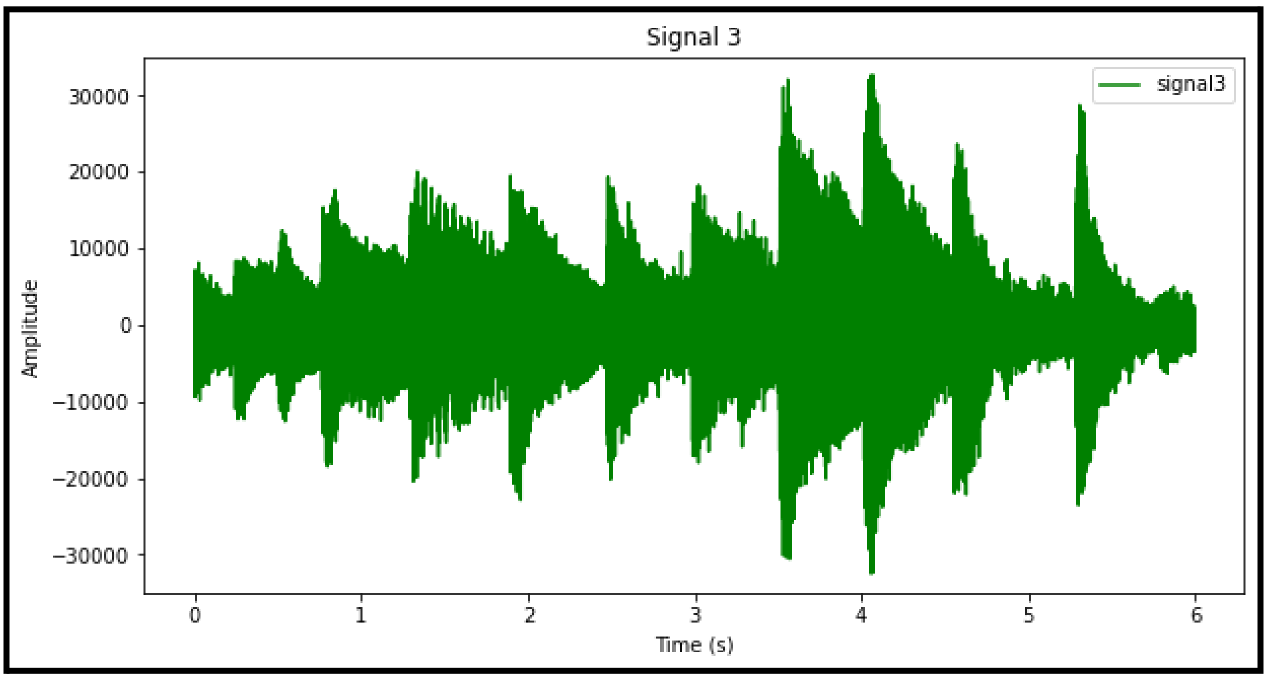
Report - Lab Assignment - 8

Question - 1

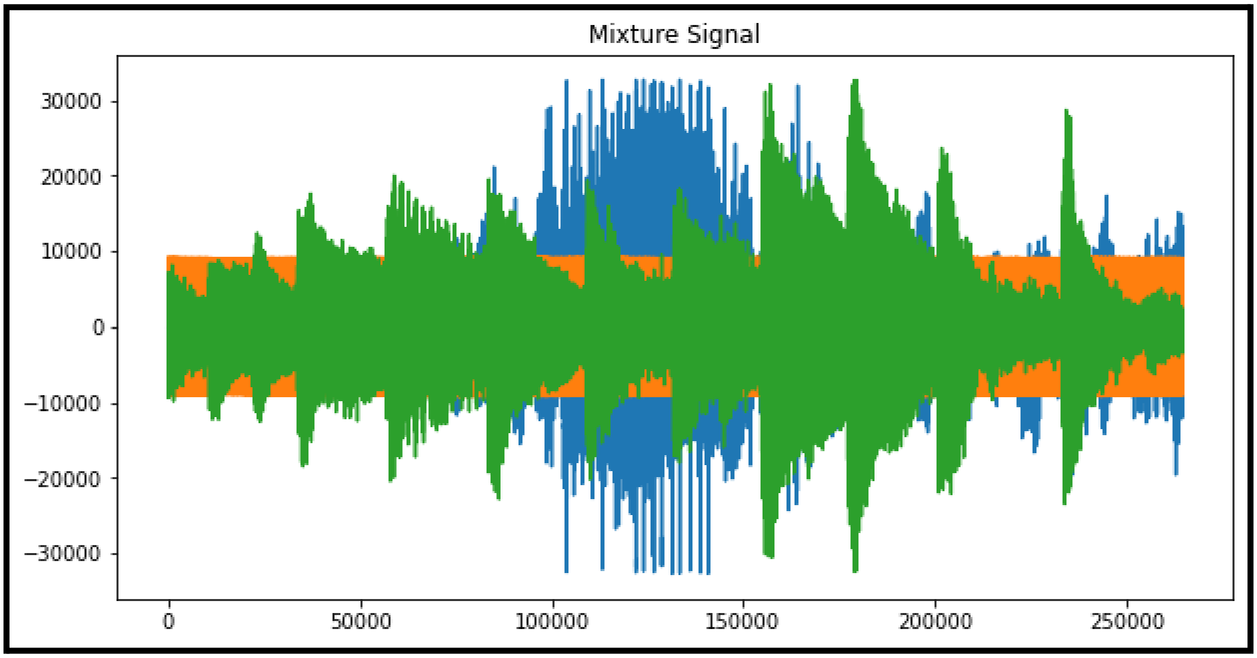
- Imported all the necessary libraries such as wave, pandas, numpy, matplotlib, warnings and IPython.
- Read the 3 audio files and opened them in reading mode.
- Then, I converted the .wav signal into integer array format by reading the audioframes.
- Then visualization of the plots was as follows:




Signal 2
signal2
Amplitude
Time (s)
30000
20000
10000
0
−10000
−20000
−30000
0
1
2
3
4
5
6

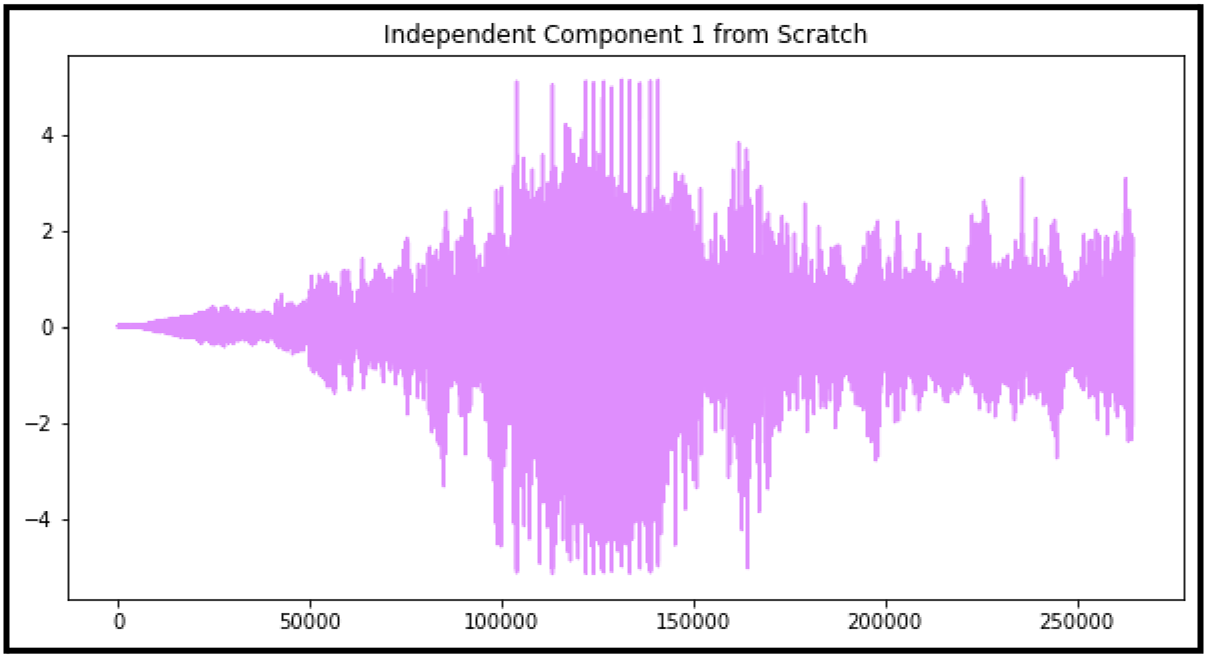

Signal 3
signal3
Amplitude
Time (s)
30000
20000
10000
0
−10000
−20000
−30000
0
1
2
3
4
5
6

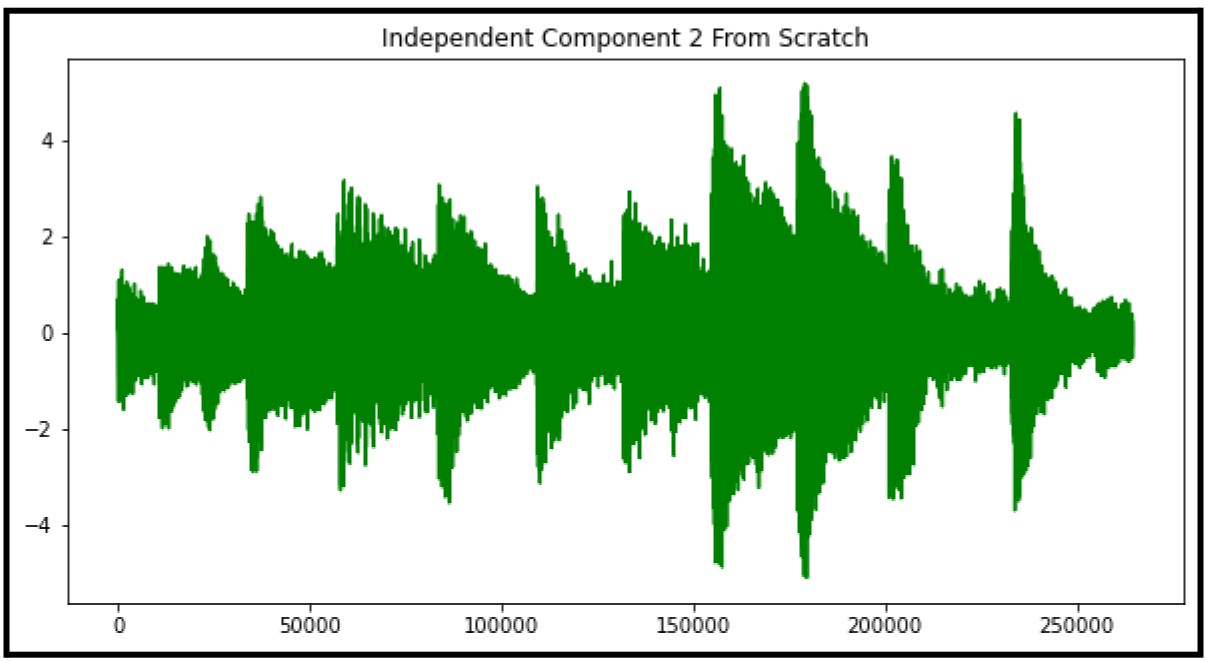
- Then, I listened to the 3 given audio files using IPython.display.Audio(<signal>)
- Then, I merged them to create dataset X.
- And, got the mixture signal as:

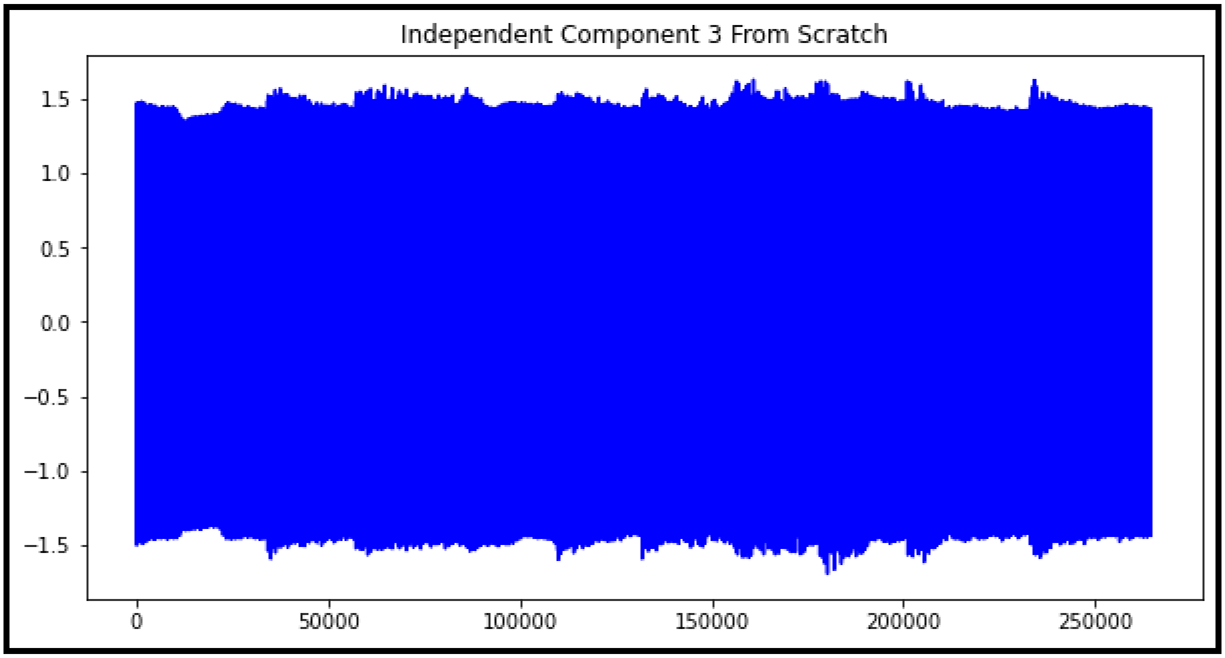


- Then, I implemented the ICA Algorithm from scratch where I passed n_components as 3 for the three signals and used 1000 iterations.
- I firstly centered our merged signal X and then whitened the signal to transform it in such a way that potential correlations between its components are removed (covariance equal to 0) and the variance of each component is equal to 1.
- I was facing an issue with having to create the W matrix with dimensions 264515, 264515 when I was passing the merged matrix X as the argument.
- So, instead what I did was pass the transpose of matrix X and the dimension of W was changed to 3, 3 which was much easier and more efficient to process.

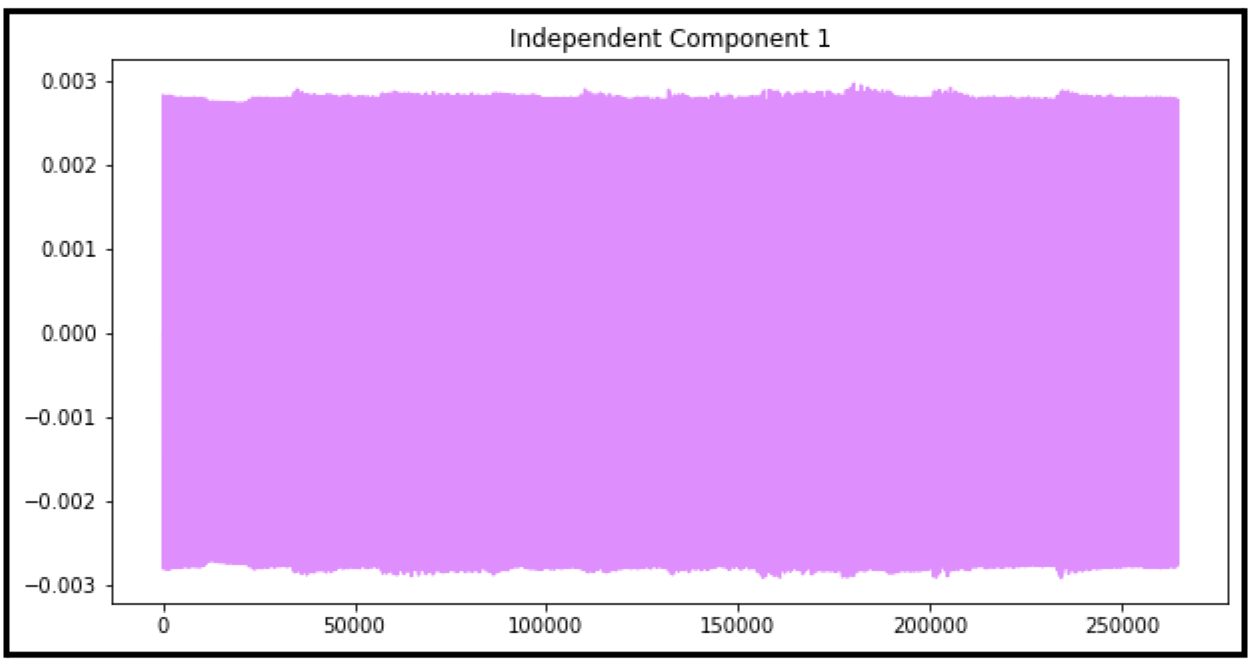
- Then, I visualized the Independent Components that I got from the Scratch implementation of ICA.

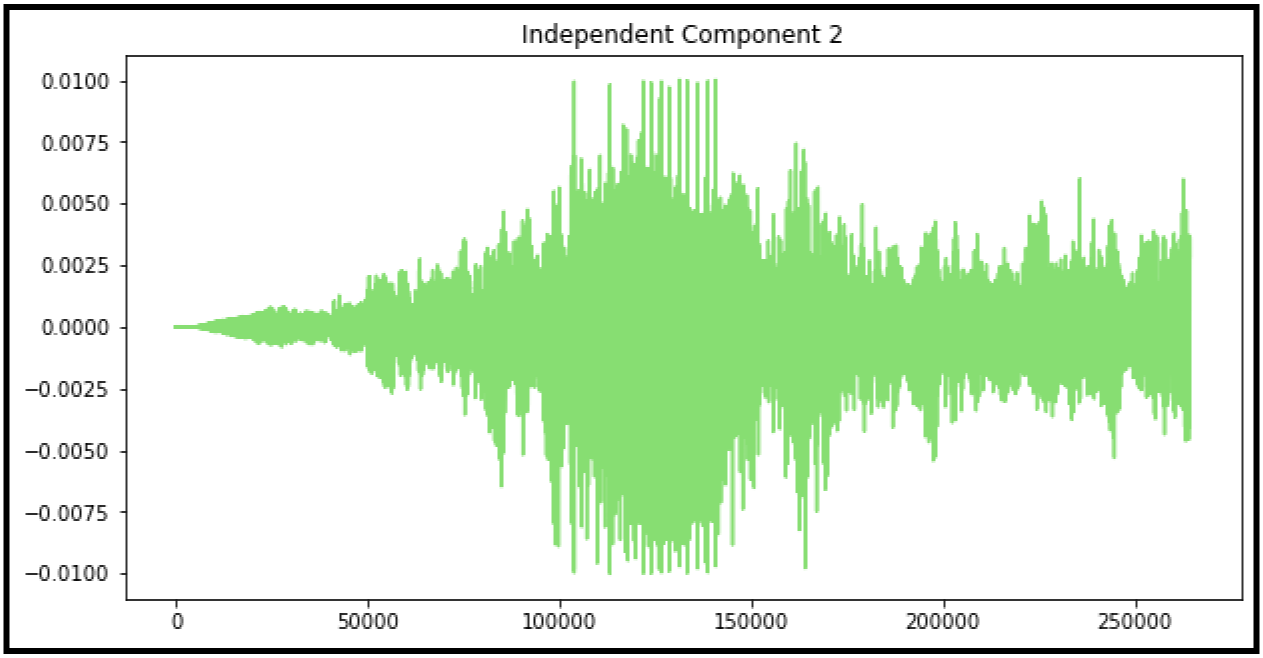


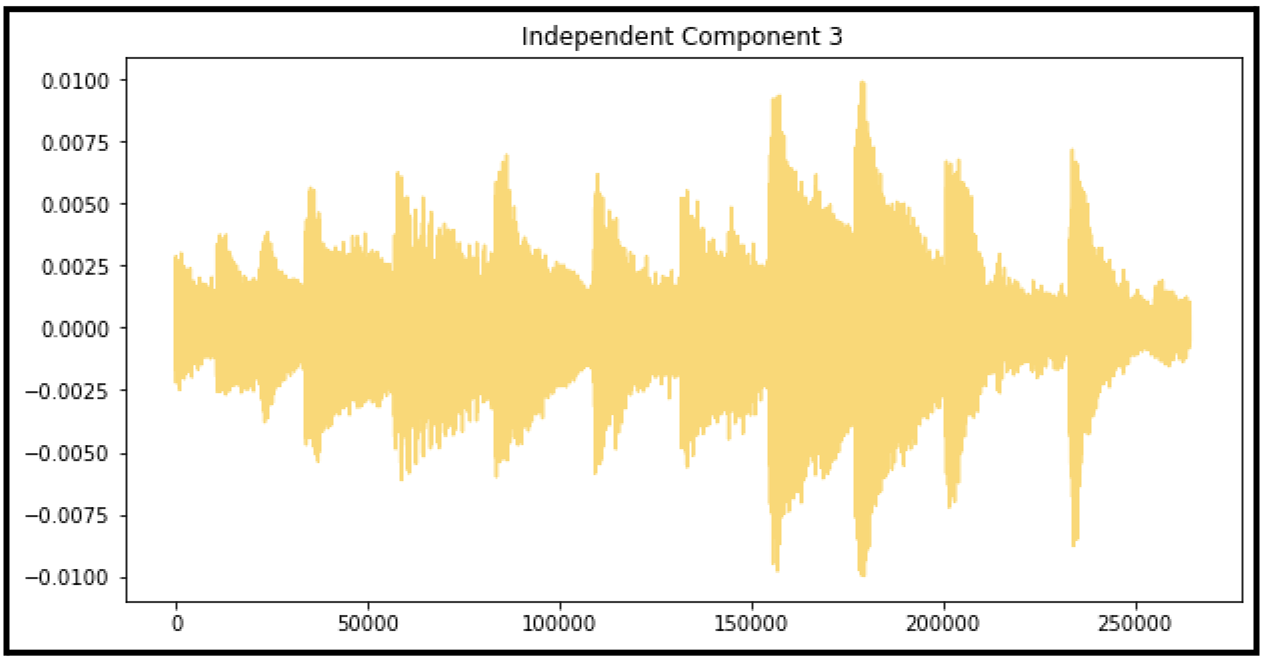




- Then, I saved the final audio files in the same directory and also played them in my jupyter notebook.
- After that I Implemented Fast ICA (import from sklearn.decomposition) selecting num_components = 3
- Plots were as follows:

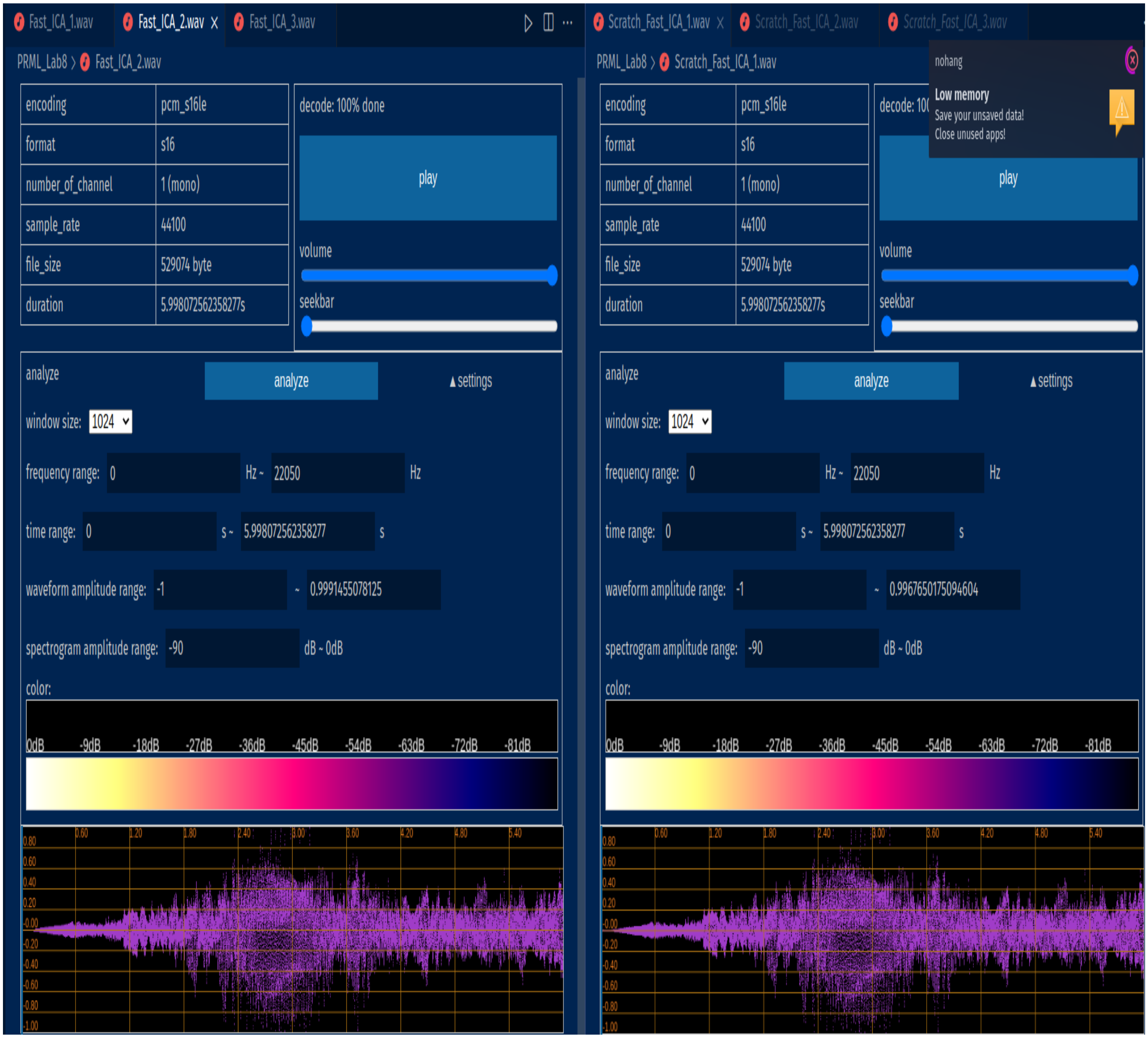






- Then, I saved the final audio files from the Fast ICA in the same directory and also played them in my jupyter notebook.

- Comparison between Scratch and FastICA for IC - 1.



Fast_ICA_1.wav
Fast_ICA_3.wav
PRML_Lab8 › Fast_ICA_1.wav
encoding
pcm_s16le
format
s16
number_of_channel
1 (mono)
sample_rate
44100
file_size
529074 byte
duration
5.998072562358277s
decode: 100% done
play
volume
seekbar
analyze
analyze
▾settings
Scratch_Fast_ICA_3.wav
PRML_Lab8 › Scratch_Fast_ICA_3.wav
encoding
pcm_s16le
format
s16
number_of_channel
1 (mono)
sample_rate
44100
file_size
529074 byte
duration
5.998072562358277s
decode: 100% done
play
volume
seekbar
analyze
analyze
▾settings

<u>Question - 2</u>

- I downloaded the train.csv file from the given dataset and preprocessed, cleaned and prepared the dataset.
- Handled the missing values and encoded the data accordingly.
- Separated features and labels as X and Y respectively.
- Then I created an object of SFS by embedding a Decision Tree classifier object, providing 10 features, forward as True, floating as False and scoring = accuracy.
- Then I trained the SFS model I got and printed out the subsets.

```
{1: {'feature_idx': (11,),
  'cv_scores': array([0.78927867, 0.79308022, 0.79038545, 0.79240653, 0.78676612]),
  'avg_score': 0.7903833960456308,
  'feature_names': ('Online boarding',)},
 2: {'feature_idx': (3, 11),
  'cv_scores': array([0.84798614, 0.85217266, 0.84798614, 0.85169145, 0.84860443]),
  'avg_score': 0.8496881632686332,
  'feature_names': ('Type of Travel', 'Online boarding')},
 3: {'feature_idx': (3, 6, 11),
  'cv_scores': array([0.89196863, 0.89307541, 0.8893701 , 0.89293104, 0.88897979]),
  'avg_score': 0.8912649918655335,
  'feature_names': ('Type of Travel',
   'Inflight wifi service',
   'Online boarding')},
 4: {'feature_idx': (3, 6, 9, 11),
  'cv_scores': array([0.91934941, 0.92319908, 0.91906068, 0.92445022, 0.92252166]),
  'avg_score': 0.9217162073206818,
  'feature_names': ('Type of Travel',
   'Inflight wifi service',
   'Gate location',
   'Online boarding')},
 5: {'feature_idx': (3, 6, 9, 11, 16),
  'cv_scores': array([0.92714499, 0.93046533, 0.92748183, 0.93171647, 0.92906641]),
  'avg_score': 0.9291750066542516,
  'feature_names': ('Type of Travel',
   'Inflight wifi service',
   'Gate location',
   'Online boarding',
   'Baggage handling')},
 6: {'feature_idx': (1, 3, 6, 9, 11, 16),
  'cv_scores': array([0.93888648, 0.94417978, 0.93917521, 0.94263991, 0.94225217]),
  'avg_score': 0.941426709515091,
  'feature_names': ('Customer Type',
   'Type of Travel',
   'Inflight wifi service',
   'Gate location',
   'Online boarding',
   'Baggage handling')},
 7: {'feature_idx': (1, 3, 4, 6, 9, 11, 16),
  'cv_scores': array([0.9472114 , 0.94865502, 0.94624898, 0.9515904 , 0.94947064]),
  'avg_score': 0.948635286758528,
  'feature_names': ('Customer Type',
   'Type of Travel',
   'Class',
   'Inflight wifi service',
   'Gate location',
   'Online boarding',
   'Baggage handling')},
 8: {'feature_idx': (1, 3, 4, 6, 9, 11, 16, 18),
  'cv_scores': array([0.95029113, 0.95178288, 0.94889563, 0.95337087, 0.95264678]),
  'avg_score': 0.9513974558180622,
  'feature_names': ('Customer Type',
   'Type of Travel',
   'Class',
   'Inflight wifi service',
   'Gate location',
   'Online boarding',
   'Baggage handling',
   'Inflight service')},
 9: {'feature_idx': (1, 3, 4, 6, 9, 11, 12, 16, 18),
  'cv_scores': array([0.95125355, 0.95211972, 0.95019489, 0.95327463, 0.95279115]),
  'avg_score': 0.9519267868836468,
  'feature_names': ('Customer Type',
   'Type of Travel',
   'Class',
   'Inflight wifi service',
   'Gate location',
   'Online boarding',
   'Seat comfort',
   'Baggage handling',
   'Inflight service')},
 10: {'feature_idx': (1, 3, 4, 6, 9, 11, 12, 13, 16, 18),
  'cv_scores': array([0.95000241, 0.95057986, 0.94894375, 0.95221597, 0.95168431]),
  'avg_score': 0.9506852575363249,
  'feature_names': ('Customer Type',
   'Type of Travel',
   'Class',
   'Inflight wifi service',
   'Gate location',
   'Online boarding',
   'Seat comfort',
   'Inflight entertainment',
   'Baggage handling',
   'Inflight service')}}
```

- 10 Best Features Selected by SFS: ('Customer Type', 'Type of Travel', 'Class', 'Inflight wifi service', 'Gate location', 'Online boarding', 'Seat comfort', 'Inflight entertainment', 'Baggage handling', 'Inflight service')
- Then, I created four different models
    1. SFS Model - forward = True, floating = False
    2. SBS Model - forward = False, floating = False
    3. SFFS Model - forward = True, floating = True
    4. SBFS Model - forward = False, floating = True

- Cross Validation Scores for SFS are: [0.94933785 0.95006929 0.94926086 0.9513397]
- Cross Validation Scores for SBS are: [0.94687404 0.94702803 0.94818294 0.95049276]
- Cross Validation Scores for SFFS are: [0.9512627 0.95095473 0.95110872 0.9525716]
- Cross Validation Scores for SBFS are: [0.94687404 0.94702803 0.94818294 0.95049276]

- Then I visualized the output from the feature selection in a pandas DataFrame format using the get_metric_dict for all four configurations.

For SFS

	feature_idx	cv_scores	avg_score	feature_names	ci_bound	std_dev	std_err
1	(11,)	[0.7895364952263628, 0.792231290421928, 0.7930...	0.790383	(Online boarding,)	0.003933	0.002454	0.001417
2	(3, 11)	[0.8483215275639051, 0.8512088081305821, 0.850...	0.849688	(Type of Travel, Online boarding)	0.001959	0.001222	0.000705
3	(3, 6, 11)	[0.8915152448413921, 0.8920157068062827, 0.892...	0.891265	(Type of Travel, Inflight wifi service, Online...	0.001961	0.001224	0.000706
4	(3, 6, 9, 11)	[0.919271635355713, 0.9229673544810595, 0.9223...	0.921735	(Type of Travel, Inflight wifi service, Gate l...	0.002316	0.001445	0.000834
5	(1, 3, 6, 9, 11)	[0.9277024946104097, 0.9285494302433015, 0.929...	0.9288	(Customer Type, Type of Travel, Inflight wifi ...	0.001189	0.000742	0.000428
6	(1, 3, 6, 9, 11, 16)	[0.9393671080997844, 0.9424468740375731, 0.939...	0.941282	(Customer Type, Type of Travel, Inflight wifi ...	0.002875	0.001793	0.001035
7	(1, 3, 4, 6, 9, 11, 16)	[0.9464505697566985, 0.9486449029873729, 0.948...	0.94826	(Customer Type, Type of Travel, Class, Infligh...	0.001774	0.001107	0.000639
8	(1, 3, 4, 6, 9, 11, 16, 18)	[0.9498383122882661, 0.9504927625500462, 0.951...	0.950791	(Customer Type, Type of Travel, Class, Infligh...	0.001155	0.00072	0.000416
9	(1, 3, 4, 6, 9, 11, 12, 16, 18)	[0.9486449029873729, 0.9508777332922698, 0.950...	0.950579	(Customer Type, Type of Travel, Class, Infligh...	0.00207	0.001291	0.000746
10	(1, 3, 4, 6, 9, 11, 12, 13, 16, 18)	[0.9493378503233755, 0.9500692947336002, 0.949...	0.950002	(Customer Type, Type of Travel, Class, Infligh...	0.001337	0.000834	0.000482

For SBS

	feature_idx	cv_scores	avg_score	feature_names	ci_bound	std_dev	std_err
22	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,...	[0.9448336926393595, 0.9468740375731445, 0.945...	0.945671	(Gender, Customer Type, Age, Type of Travel, C...	0.001303	0.000813	0.000469
21	(0, 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14...	[0.9449491838620265, 0.9470665229442562, 0.946...	0.946335	(Gender, Customer Type, Age, Type of Travel, C...	0.001324	0.000826	0.000477
20	(0, 1, 2, 3, 4, 6, 7, 9, 10, 11, 12, 13, 14, 1...	[0.9448336926393595, 0.9459886048660302, 0.945...	0.946306	(Gender, Customer Type, Age, Type of Travel, C...	0.00215	0.001341	0.000774
19	(1, 2, 3, 4, 6, 7, 9, 10, 11, 12, 13, 14, 15, ...	[0.944795195565137, 0.9463350785340314, 0.9463...	0.946643	(Customer Type, Age, Type of Travel, Class, In...	0.002493	0.001555	0.000898
18	(1, 2, 3, 4, 6, 9, 10, 11, 12, 13, 14, 15, 16,...	[0.9457191253464736, 0.9464890668309208, 0.945...	0.946759	(Customer Type, Age, Type of Travel, Class, In...	0.002774	0.001731	0.000999
17	(1, 2, 3, 4, 6, 9, 10, 11, 12, 13, 14, 15, 16,...	[0.946181090237142, 0.947143517092701, 0.94548...	0.946922	(Customer Type, Age, Type of Travel, Class, In...	0.002039	0.001272	0.000734
16	(1, 2, 3, 4, 6, 9, 11, 12, 13, 14, 15, 16, 17,...	[0.947143517092701, 0.9479904527255929, 0.9444...	0.946961	(Customer Type, Age, Type of Travel, Class, In...	0.002378	0.001483	0.000856
15	(1, 2, 3, 4, 6, 9, 11, 12, 13, 15, 16, 17, 18,...	[0.9460271019402525, 0.9462580843855867, 0.946...	0.946662	(Customer Type, Age, Type of Travel, Class, In...	0.000871	0.000543	0.000314
14	(1, 2, 3, 4, 6, 9, 11, 12, 13, 16, 17, 18, 19,...	[0.9444487218971358, 0.9460271019402525, 0.943...	0.945382	(Customer Type, Age, Type of Travel, Class, In...	0.002131	0.001329	0.000767
13	(1, 2, 3, 4, 6, 9, 11, 12, 16, 17, 18, 19, 21)	[0.9432553125962427, 0.9472205112411457, 0.944...	0.944603	(Customer Type, Age, Type of Travel, Class, In...	0.00264	0.001647	0.000951
12	(1, 2, 3, 4, 6, 9, 11, 12, 16, 17, 18, 21)	[0.9432168155220203, 0.9453341546042501, 0.945...	0.944728	(Customer Type, Age, Type of Travel, Class, In...	0.001529	0.000954	0.000551
11	(1, 2, 3, 4, 6, 9, 11, 12, 16, 17, 18)	[0.9438327687095781, 0.9458731136433631, 0.944...	0.94494	(Customer Type, Age, Type of Travel, Class, In...	0.001596	0.000995	0.000575
10	(1, 3, 4, 6, 9, 11, 12, 16, 17, 18)	[0.9468740375731445, 0.9470280258700339, 0.948...	0.948144	(Customer Type, Type of Travel, Class, Infligh...	0.00232	0.001447	0.000835

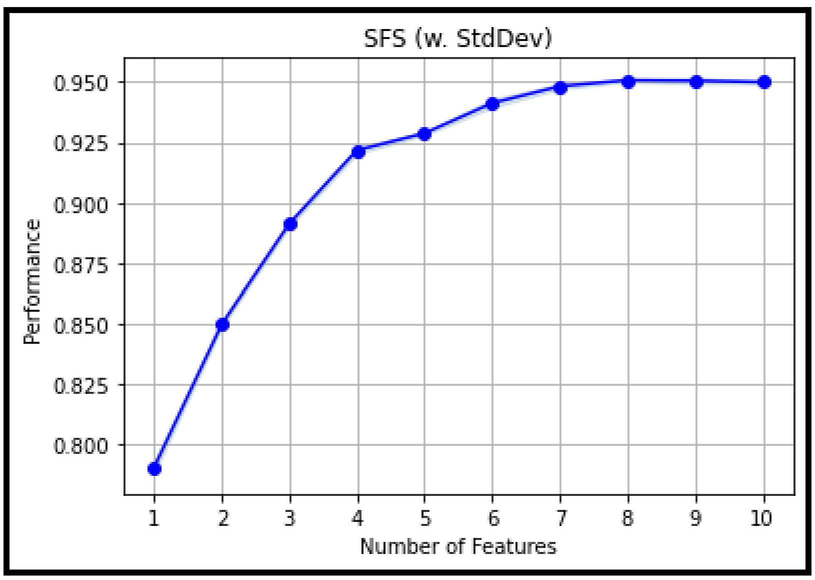
For SFFS

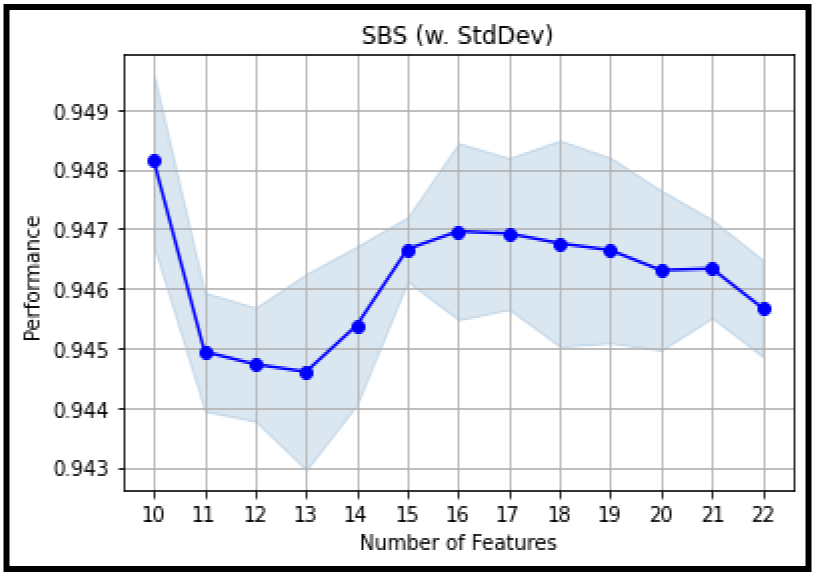
	feature_idx	cv_scores	avg_score	feature_names	ci_bound	std_dev	std_err
1	(11,)	[0.7895364952263628, 0.792231290421928, 0.7930...	0.790383	(Online boarding,)	0.003933	0.002454	0.001417
2	(3, 11)	[0.8483215275639051, 0.8512088081305821, 0.850...	0.849688	(Type of Travel, Online boarding)	0.001959	0.001222	0.000705
3	(3, 6, 11)	[0.8915152448413921, 0.8920157068062827, 0.892...	0.891265	(Type of Travel, Inflight wifi service, Online...	0.001961	0.001224	0.000706
4	(3, 6, 9, 11)	[0.919271635355713, 0.9229673544810595, 0.9223...	0.921735	(Type of Travel, Inflight wifi service, Gate l...	0.002316	0.001445	0.000834
5	(1, 3, 6, 9, 11)	[0.9277024946104097, 0.9285494302433015, 0.929...	0.9288	(Customer Type, Type of Travel, Inflight wifi ...	0.001189	0.000742	0.000428
6	(1, 3, 6, 9, 11, 16)	[0.9393671080997844, 0.9424468740375731, 0.939...	0.941282	(Customer Type, Type of Travel, Inflight wifi ...	0.002875	0.001793	0.001035
7	(1, 3, 4, 6, 9, 11, 16)	[0.9464505697566985, 0.9486449029873729, 0.948...	0.94826	(Customer Type, Type of Travel, Class, Infligh...	0.001774	0.001107	0.000639
8	(1, 3, 4, 6, 9, 11, 16, 18)	[0.9498383122882661, 0.9504927625500462, 0.951...	0.950791	(Customer Type, Type of Travel, Class, Infligh...	0.001155	0.00072	0.000416
9	(1, 3, 4, 6, 11, 12, 13, 16, 18)	[0.9506467508469356, 0.9507237449953804, 0.950...	0.950993	(Customer Type, Type of Travel, Class, Infligh...	0.001523	0.00095	0.000548
10	(1, 3, 4, 6, 11, 12, 13, 16, 18, 19)	[0.9512627040344934, 0.9509547274407145, 0.951...	0.951474	(Customer Type, Type of Travel, Class, Infligh...	0.00103	0.000643	0.000371

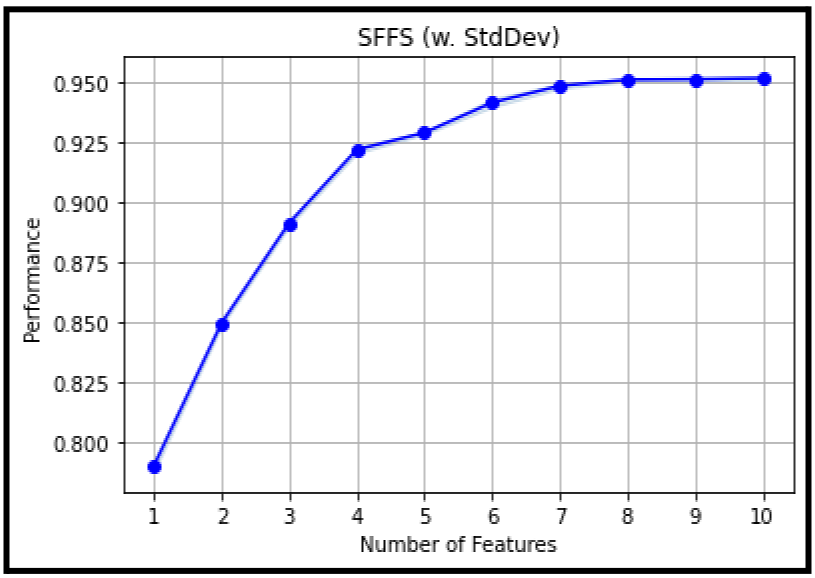
For SBFS

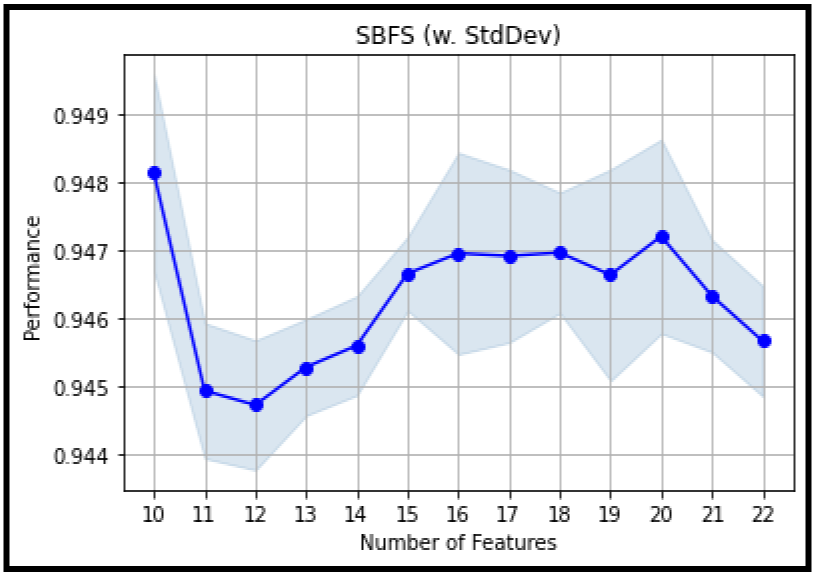
	feature_idx	cv_scores	avg_score	feature_names	ci_bound	std_dev	std_err
22	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,...	[0.9448336926393595, 0.9468740375731445, 0.945...	0.945671	(Gender, Customer Type, Age, Type of Travel, C...	0.001303	0.000813	0.000469
21	(0, 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14...	[0.9449491838620265, 0.9470665229442562, 0.946...	0.946335	(Gender, Customer Type, Age, Type of Travel, C...	0.001324	0.000826	0.000477
20	(1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 1...	[0.9451801663073607, 0.94810594394826, 0.94664...	0.947211	(Customer Type, Age, Type of Travel, Class, Fl...	0.002288	0.001427	0.000824
19	(1, 2, 3, 4, 6, 7, 9, 10, 11, 12, 13, 14, 15, ...	[0.944795195565137, 0.9463350785340314, 0.9463...	0.946643	(Customer Type, Age, Type of Travel, Class, In...	0.002493	0.001555	0.000898
18	(0, 1, 2, 3, 4, 6, 9, 10, 11, 12, 13, 14, 15, ...	[0.9466430551278103, 0.9476054819833692, 0.945...	0.94697	(Gender, Customer Type, Age, Type of Travel, C...	0.00142	0.000886	0.000511
17	(1, 2, 3, 4, 6, 9, 10, 11, 12, 13, 14, 15, 16,...	[0.946181090237142, 0.947143517092701, 0.94548...	0.946922	(Customer Type, Age, Type of Travel, Class, In...	0.002039	0.001272	0.000734
16	(1, 2, 3, 4, 6, 9, 11, 12, 13, 14, 15, 16, 17,...	[0.947143517092701, 0.9479904527255929, 0.9444...	0.946961	(Customer Type, Age, Type of Travel, Class, In...	0.002378	0.001483	0.000856
15	(1, 2, 3, 4, 6, 9, 11, 12, 13, 15, 16, 17, 18,...	[0.9460271019402525, 0.9462580843855867, 0.946...	0.946662	(Customer Type, Age, Type of Travel, Class, In...	0.000871	0.000543	0.000314
14	(1, 2, 3, 4, 6, 9, 11, 12, 14, 16, 17, 18, 20,...	[0.94467970434247, 0.9463735756082537, 0.94510...	0.945604	(Customer Type, Age, Type of Travel, Class, In...	0.001168	0.000729	0.000421
13	(1, 2, 3, 4, 6, 9, 11, 12, 14, 16, 17, 18, 21)	[0.945218663381583, 0.9452571604558053, 0.9443...	0.945286	(Customer Type, Age, Type of Travel, Class, In...	0.001137	0.00071	0.00041
12	(1, 2, 3, 4, 6, 9, 11, 12, 16, 17, 18, 21)	[0.9432168155220203, 0.9453341546042501, 0.945[illegible]	[illegible]	[illegible]stomer Type, Age, Type of Travel, Class, In...	0.001529	0.000954	0.000551
11	(1, 2, 3, 4, 6, 9, 11, 12, 16, 17, 18)	[0.9438327687095781, 0.9458731136433631, 0.944...	[illegible]	[illegible]stomer Type, Age, Type of Travel, Class, In...	0.001596	0.000995	0.000575
10	(1, 3, 4, 6, 9, 11, 12, 16, 17, 18)	[0.9468740375731445, 0.9470280258700339, 0.948...	0.948144	(Customer Type, Type of Travel, Class, Infligh...	0.00232	0.001447	0.000835

- Then I plotted the results for each configuration (from mlxtend.plotting import plot_sequential_feature_selection as plot_sfs).

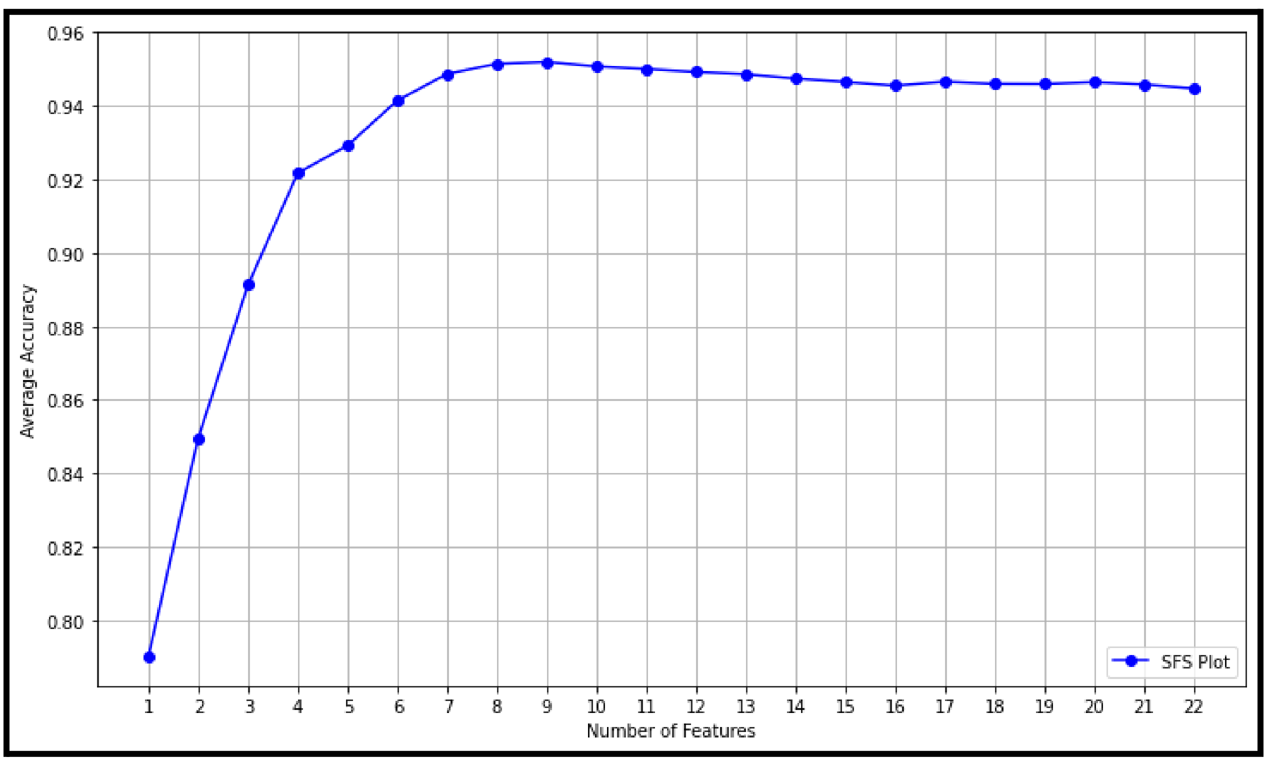



SBS (w. StdDev)
Performance
0.949
0.948
0.947
0.946
0.945
0.944
0.943
10
11
12
13
14
15
16
17
18
19
20
21
22
Number of Features


SFFS (w. StdDev)
Performance
0.950
0.925
0.900
0.875
0.850
0.825
0.800
1
2
3
4
5
6
7
8
9
10
Number of Features



- Then I created objects of SFS by embedding Decision Tree classifier objects, varying features from 1 to complete 22, forward as True, floating as False and scoring = accuracy and observed the following results.



We see that the best number of features is 9 with an average accuracy of 95.05