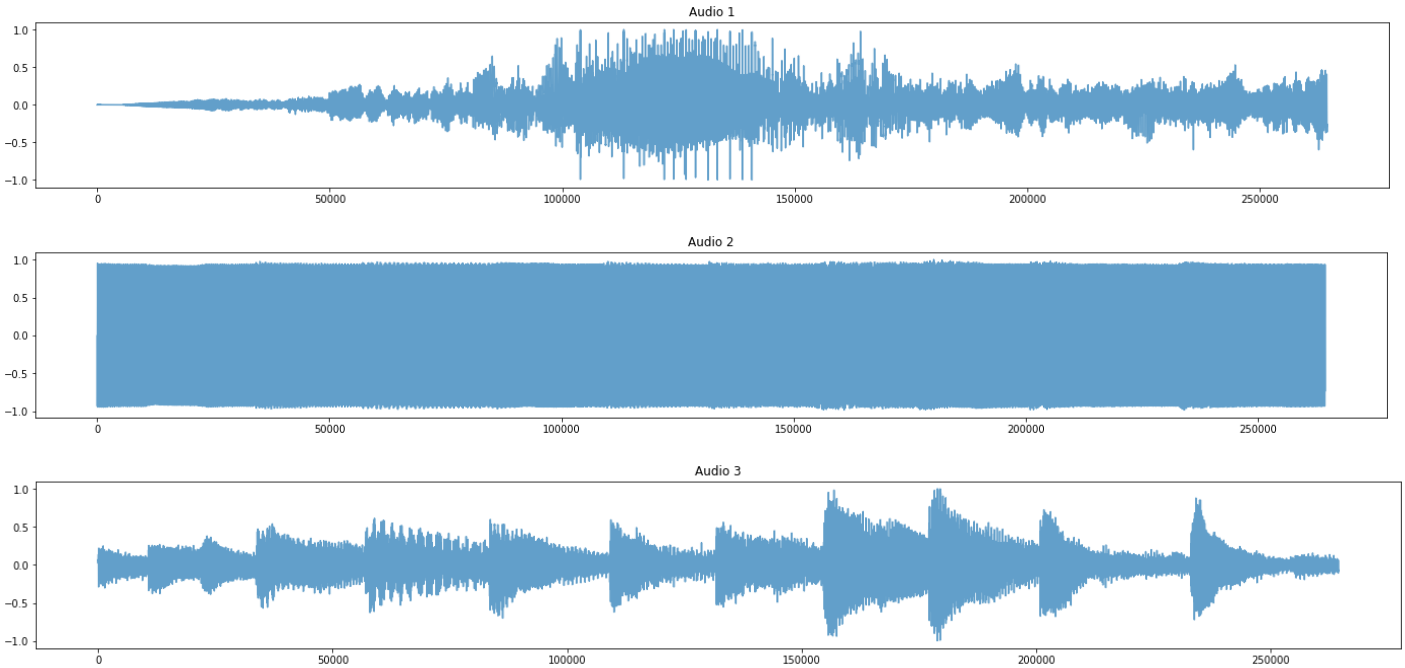
Maniya Yash Rajeshbhai
B20CS033

ML Lab 8 Report

Question 1 :

1. Read, Visualise and Listen the audio files :

- Used wave library to read the .wav audio files and converted them into arrays containing amplitudes. Plotted all the three audio signals, also displayed audio player for listening to the audios.



2. Creating Dataset :

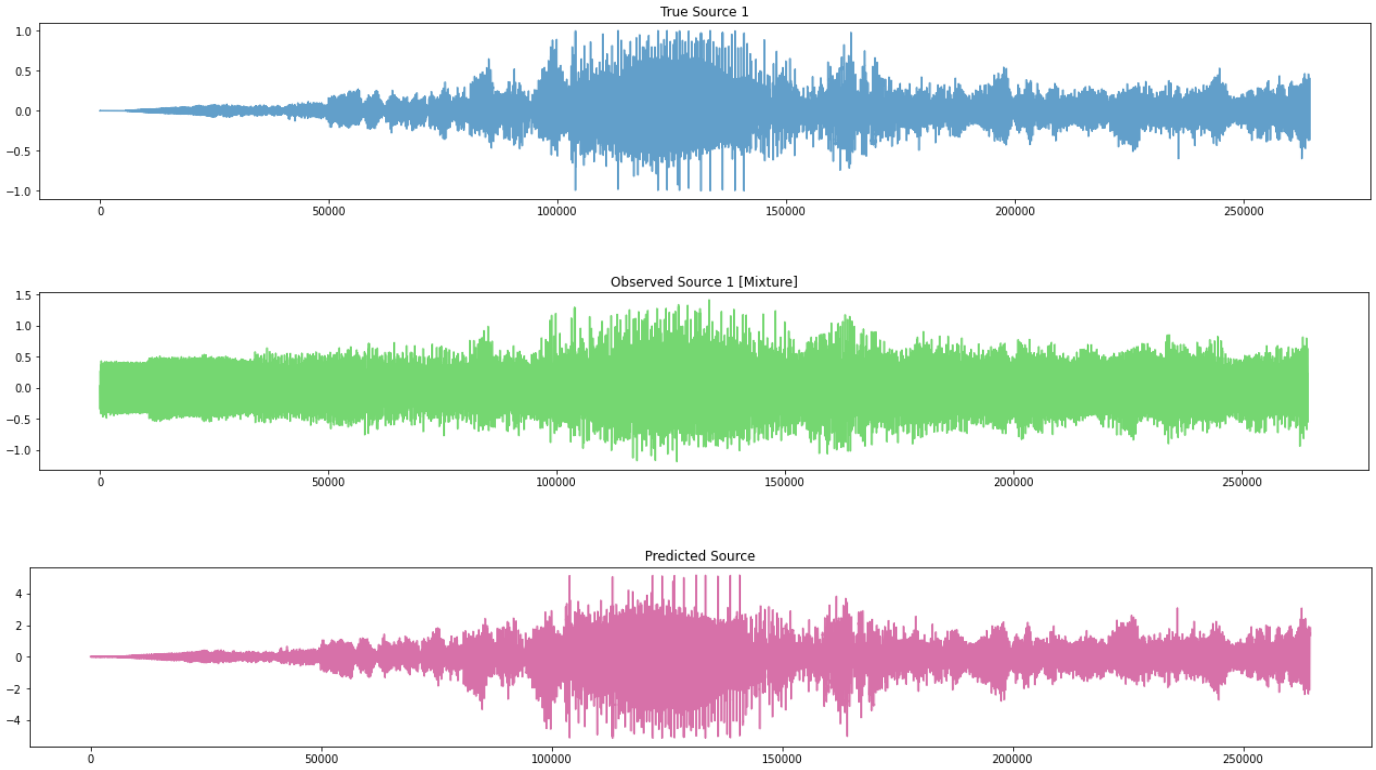
- From the extracted audio amplitude values, created a dataframe of required shape of (3x frames).

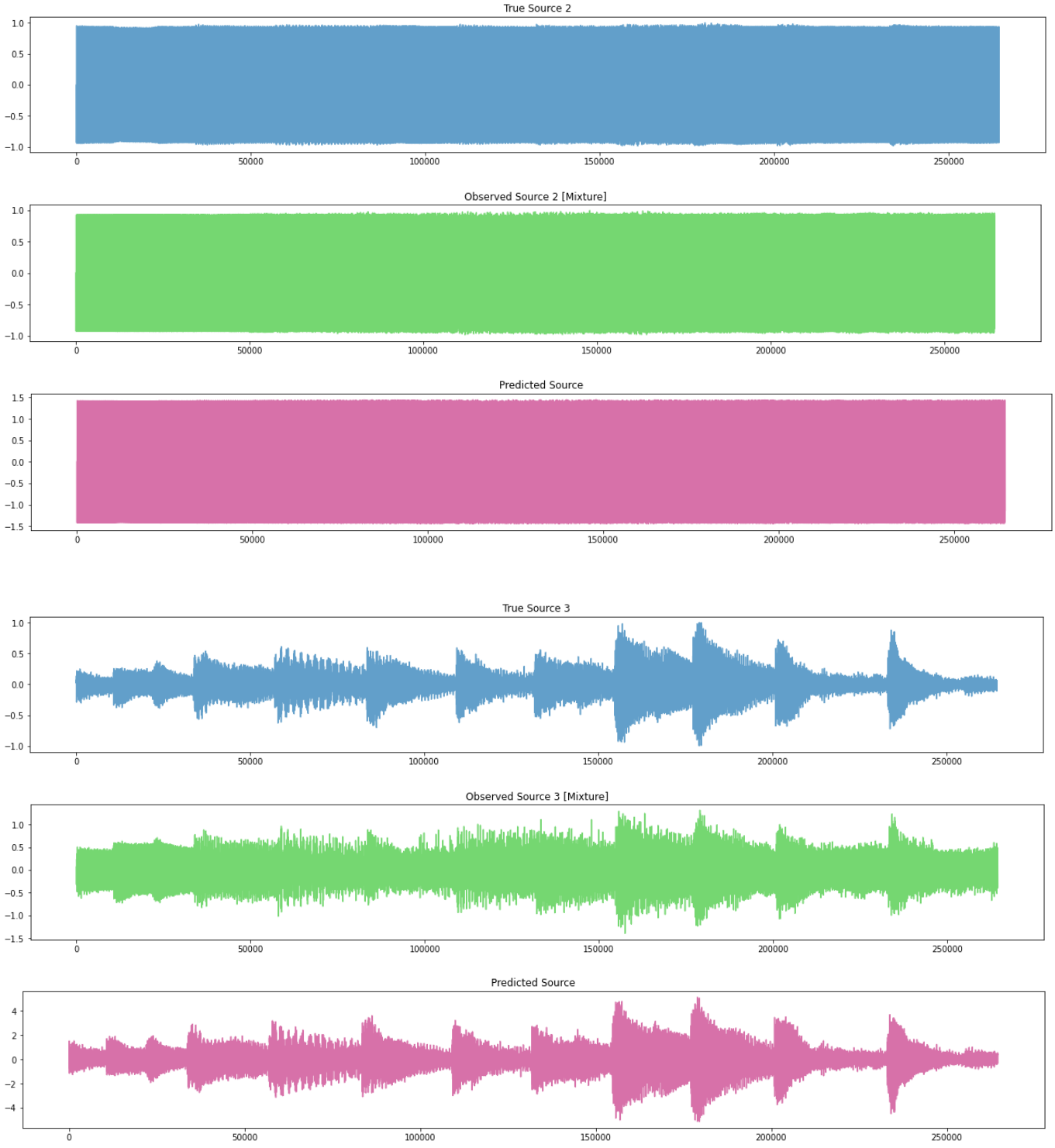
3. ICA from scratch :

- Defined Mixing matrix to get mixed signals (observed sources).
- Centred and whitened the data by subtracting the mean and modifying data using dot products with covariance matrix.
- Implemented ICA from scratch applying the required operations : tanh, matrix multiplications with W matrix (initialised with random values).

4. Creating Plots :

- Plotted the signals of **True Signals, Observed Mixed Sources, Predicted Sources** for comparing the True Signals and Predicted Sources.
- We see that Mixed signals possess the characteristics of all three Signals in different proportions (depends on the Mixing Matrix).
- The Predicted Sources are almost similar to the Original Sources so we say that ICA we implemented works charm.



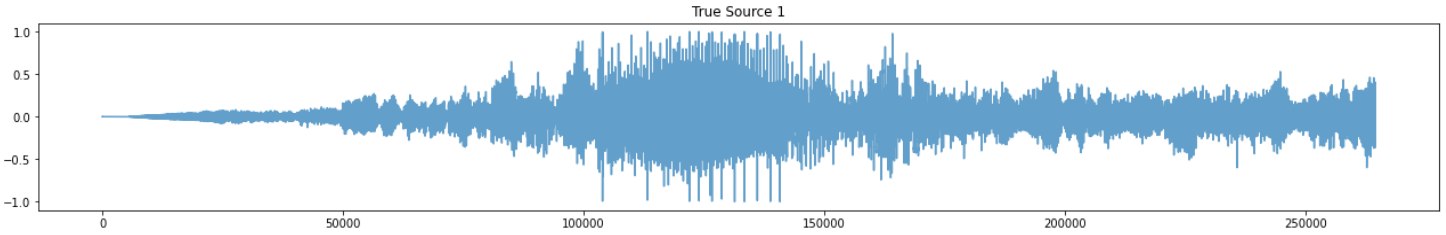


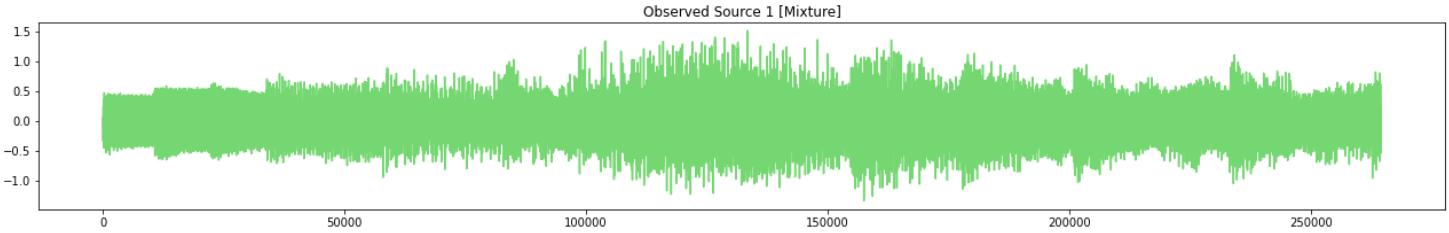
5. FastICA Implementation :

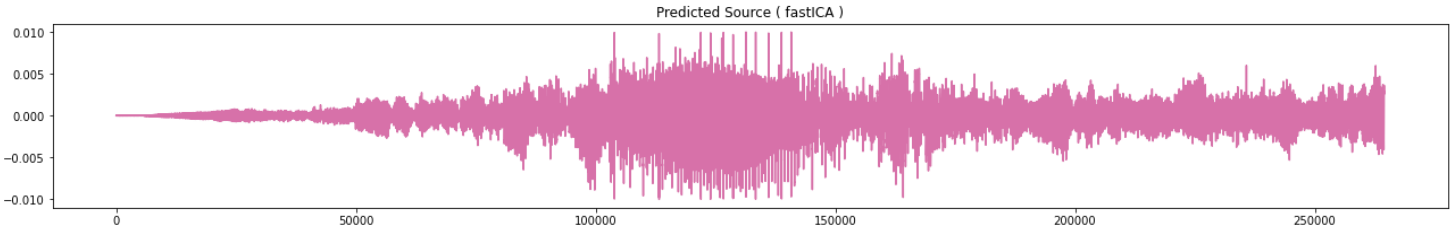
- Used sklearn to implement the FastICA with num_components = 3 and random_state = 0.

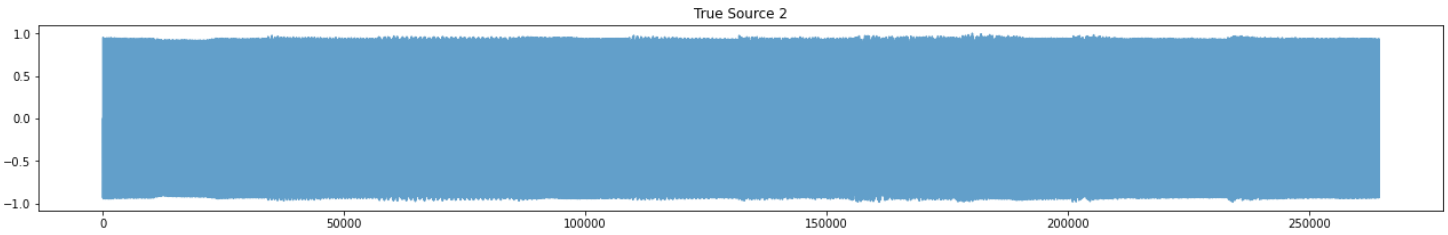
6. Visualising and listening the predicted Sources :

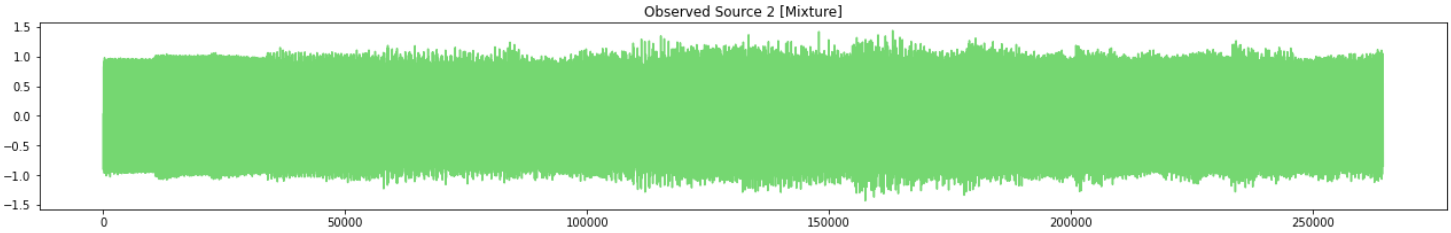
- Plotted the signals of **True Signals, Observed Mixed Sources, Predicted Sources** we got from fastICA.
- The observations are almost similar to that we got in scratch implementation.
- Since the mixing matrix was changed, the Mixed sources seem to be different but however this does not change the predicted sources.
- Predicted Sources from fastICA are almost similar to the Original sources.

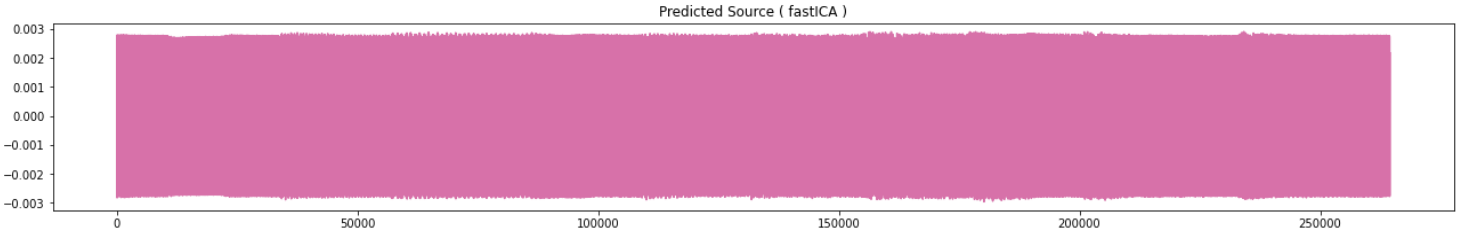


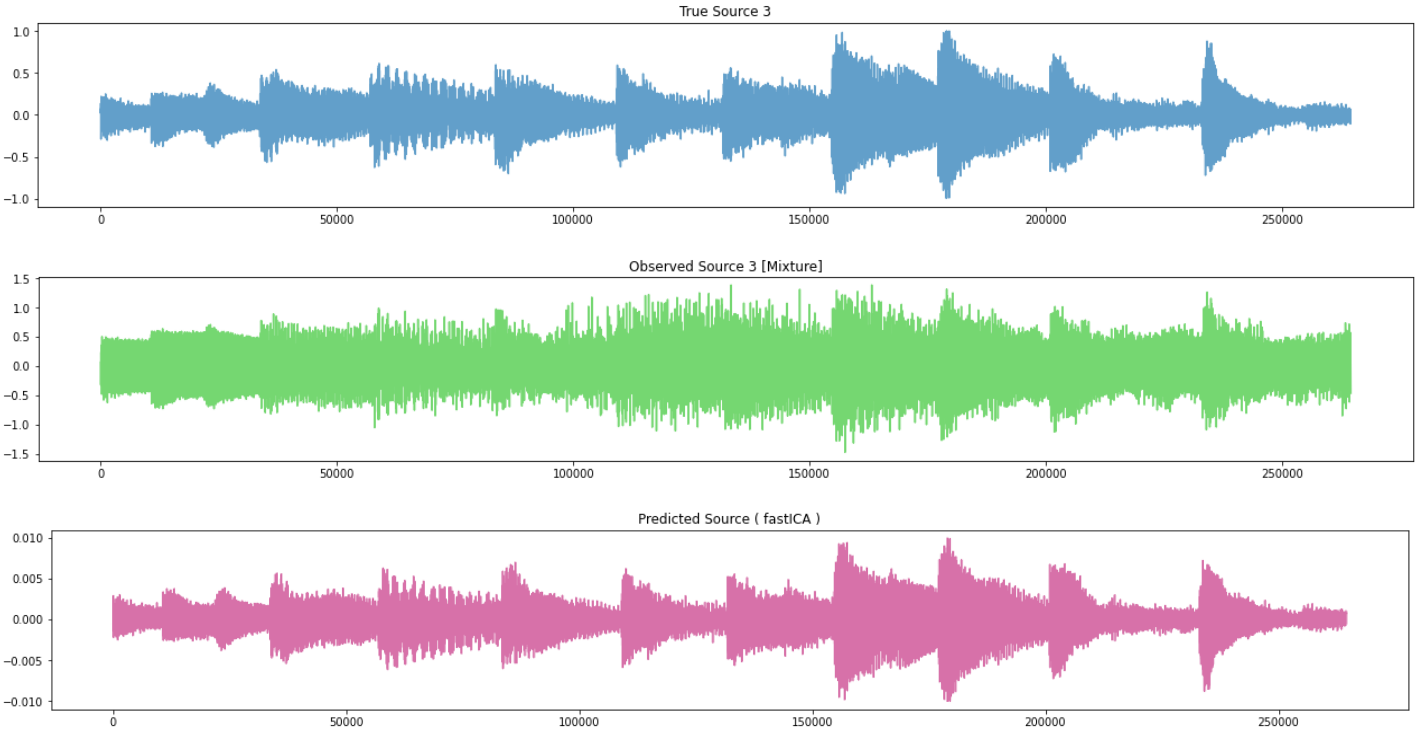












7. Observations and Conclusions :

- We see that both ICA and FastICA predicts almost the same results [From Plots].
- Both the algorithms are almost similar but the former is slower and takes up more memory than FastICA [Googled].

Question 2 :

A. Preprocessing :

- Preprocessed the data. Did scaling for continuous data and label encoding for categorical classes.

B. SFS Implementation from scratch :

- Implemented Sequential Feature Selection [scratch_SFS() class] from scratch by selecting features which gave maximum accuracy on validation data and kept adding them to a list to keep track of best k_features.
- Also used cross validation while looping over all the features.

C. Training SFS model:

- Trained the SFS model to extract the best 10 features and got the following results.

```
Selected 10 Features are :
['Online boarding', 'Type of Travel', 'Inflight wifi service', 'Gate location', 'Inflight service', 'Customer Type',

Accuracies for the Selected Features :
[0.784, 0.8503, 0.8919, 0.9245, 0.9284, 0.9409, 0.951, 0.9511, 0.9528, 0.9535]
```

Top 6 features are visible in the image. Also we see with increasing number of features, validation accuracy increases.

D. Train using different configurations :

- Trained following configurations from SFS model by changing the parameters:
  - SFS (forward True, floating False)
  - SBS (forward False, floating False)
  - SFFS (forward True, floating True)
  - SBFS (forward False, floating True)

Cross Validation accuracies on different configurations :

```
CV Scores for SFS:
[0.94903278 0.95007529 0.9501892  0.95177234] , Mean Accuracy = 0.9503

CV Scores for SBS:
[0.94883972 0.95022974 0.94953278 0.95188818] , Mean Accuracy = 0.9501

CV Scores for SFFS:
[0.95065447 0.95146531 0.95057533 0.95281489] , Mean Accuracy = 0.9514

CV Scores for SBFS:
[0.94737249 0.94694776 0.94764074 0.9498803 ] , Mean Accuracy = 0.9480
```

We see that the SFFS model proves best, giving the best 10 features which results in 0.9514 mean accuracy.
Whereas the SBFS model performs poor compared to others.

E. Visualise the outputs [using get_metric_dict]:

Made dataframes for all the configurations of SFS from the recorded metrics obtained from sfs.get_metric_dict .

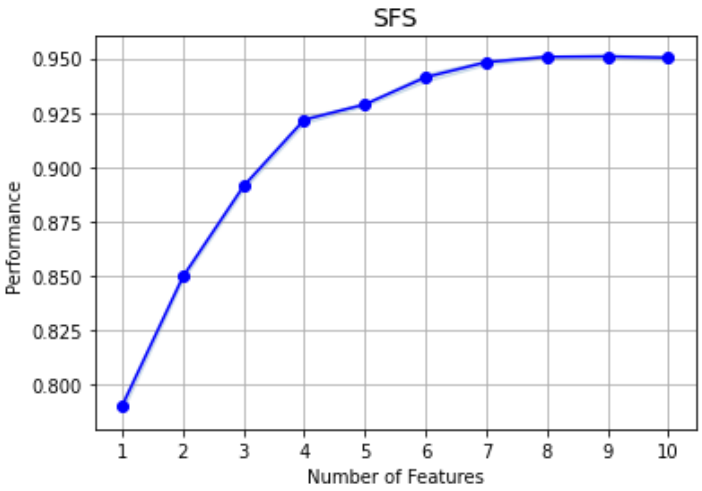
For SFS :

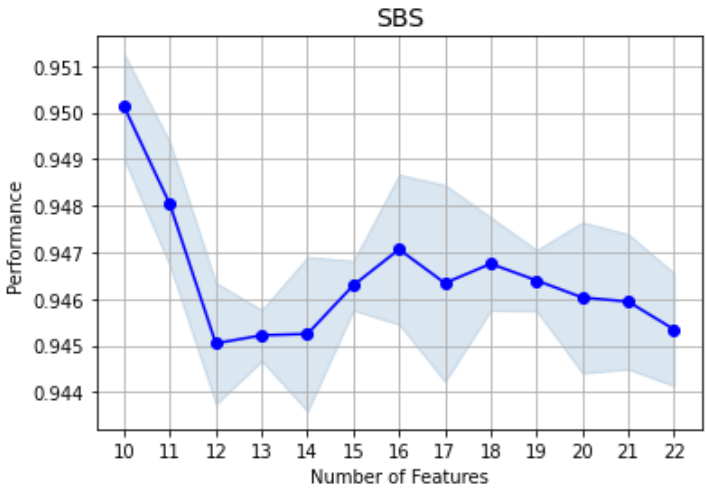
	feature_idx	cv_scores	avg_score	feature_names	ci_bound	std_dev	std_err
1	(11,)	[0.7897602224024094, 0.7918066334607514, 0.793...	0.790335	(Online boarding,)	0.004065	0.002536	0.001464
2	(3, 11)	[0.8483339125062743, 0.8511911656820726, 0.850...	0.849615	(Type of Travel, Online boarding)	0.002085	0.0013	0.000751
3	(3, 6, 11)	[0.8914629908490675, 0.8920421637901077, 0.892...	0.891249	(Type of Travel, Inflight wifi service, Online...	0.002214	0.001381	0.000797
4	(3, 6, 9, 11)	[0.9192246804895942, 0.9228927757828488, 0.922...	0.921733	(Type of Travel, Inflight wifi service, Gate l...	0.002346	0.001463	0.000845
5	(1, 3, 6, 9, 11)	[0.9277578284875864, 0.9284914475462374, 0.929...	0.928828	(Customer Type, Type of Travel, Inflight wifi ...	0.001202	0.00075	0.000433
6	(1, 3, 6, 9, 11, 16)	[0.9393412873083903, 0.9425846557782154, 0.939...	0.9413	(Customer Type, Type of Travel, Inflight wifi ...	0.002925	0.001825	0.001053
7	(1, 3, 4, 6, 9, 11, 16)	[0.946368585659678, 0.9487625004826441, 0.9485...	0.94824	(Customer Type, Type of Travel, Class, Infligh...	0.001789	0.001116	0.000645
8	(1, 3, 4, 6, 9, 11, 16, 18)	[0.9497664002471138, 0.9503841847175567, 0.950...	0.950673	(Customer Type, Type of Travel, Class, Infligh...	0.001128	0.000704	0.000406
9	(1, 3, 4, 6, 9, 11, 12, 16, 18)	[0.9491486157766709, 0.9509247461291942, 0.951...	0.950866	(Customer Type, Type of Travel, Class, Infligh...	0.001748	0.001091	0.00063
10	(1, 3, 4, 6, 9, 11, 12, 13, 16, 18)	[0.9490327811884629, 0.9500752924823352, 0.950...	0.950267	(Customer Type, Type of Travel, Class, Infligh...	0.001569	0.000979	0.000565

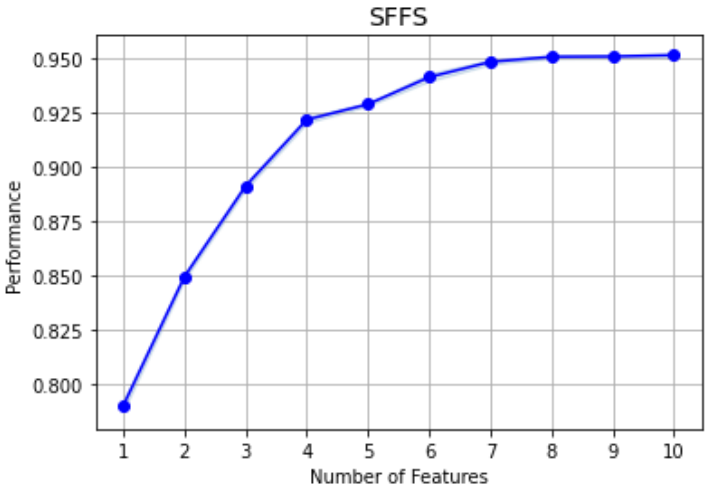
F. Plotting results using plot_sequential_feature_selection :

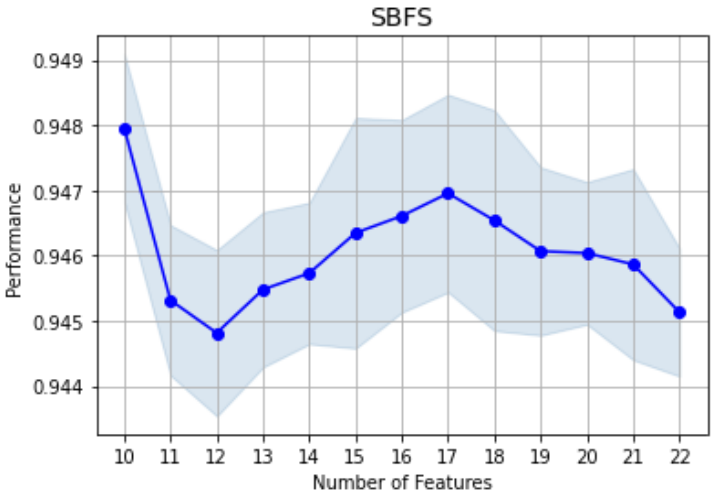
We see that for SFS and SFFS, the accuracy increases with an increase in number of features as they follow forward propagation.

However for SBS and SBFS, accuracy decreases with increasing number of features. This happened because of backward propagation.



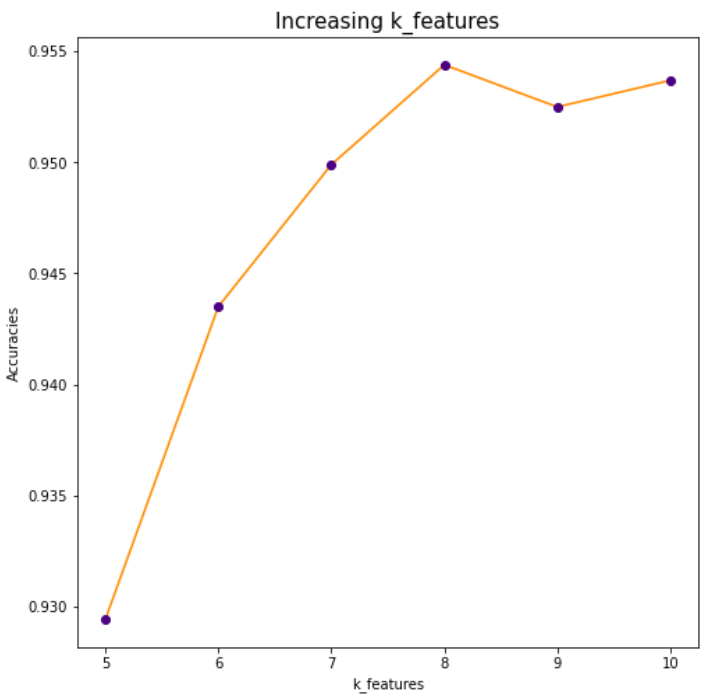


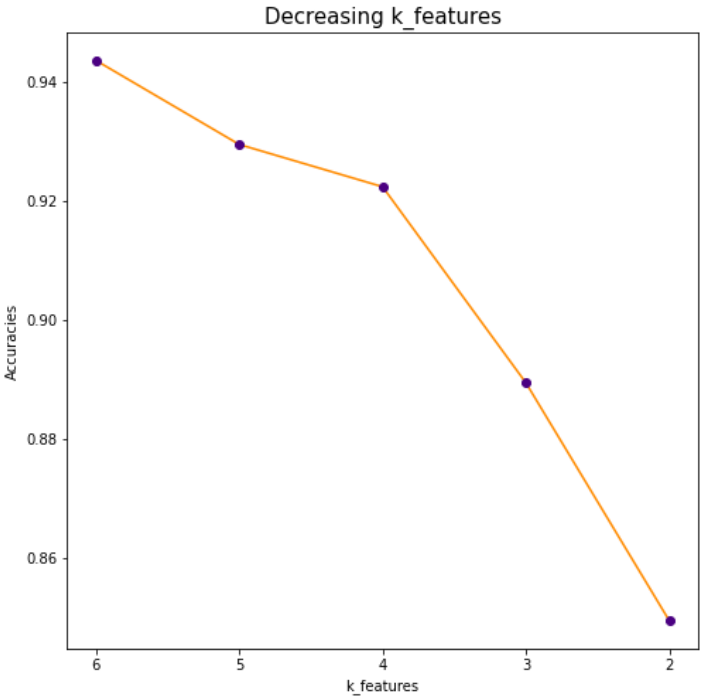




G. Increasing and Decreasing number of Features :

We see that for higher numbers of features, accuracy increases. This is expected as using a lesser number of features would indirectly mean data loss while predicting our class.





End of the Report !