Part A Question 4

In [ ]:
```
import time
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import tensorflow as tf
import shap
shap.initjs()


import IPython.display as ipd

from scipy.io import wavfile as wav

from sklearn import preprocessing
from sklearn.model_selection import KFold
from sklearn.model_selection import train_test_split
from sklearn.metrics import f1_score, precision_score, recall_score, confusion_matrix


import tensorflow.keras as keras
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import *
from tensorflow.keras.regularizers import l2
from tensorflow.keras.callbacks import ReduceLROnPlateau,EarlyStopping,ModelCheckpoint,
from sklearn import datasets
from sklearn.model_selection import KFold
```

```
c:\Users\JoeTe\AppData\Local\Programs\Python\Python310\lib\site-packages\tqdm\auto.py:2
2: TqdmWarning: IProgress not found. Please update jupyter and ipywidgets. See https://
ipywidgets.readthedocs.io/en/stable/user_install.html
  from .autonotebook import tqdm as notebook_tqdm
```

In [ ]:
```
SEED = 42
import os
os.environ['TF_CUDNN_DETERMINISTIC'] = '1'

import random
random.seed(SEED)

import numpy as np
np.random.seed(SEED)

import tensorflow as tf
tf.random.set_seed(SEED)
```

In [ ]:
```
df = pd.read_csv('./full.csv')
df.head()
```

Out[ ]:

	filename	tempo	total_beats	average_beats	chroma_stft_mean	chroma
0	app_3001_4001_phnd_neg_0000.wav	184.570312	623	69.222222	0.515281	
1	app_3001_4001_phnd_neg_0001.wav	151.999081	521	74.428571	0.487201	
2	app_3001_4001_phnd_neg_0002.wav	112.347147	1614	146.727273	0.444244	
3	app_3001_4001_phnd_neg_0003.wav	107.666016	2060	158.461538	0.454156	
4	app_3001_4001_phnd_neg_0004.wav	75.999540	66	33.000000	0.478780	

5 rows × 78 columns

In [ ]:
```
df['label'] = df['filename'].str.split('_').str[-2]
df['label'].value_counts()
```

Out[ ]:
```
pos    92826
neg    89428
Name: label, dtype: int64
```

In [ ]:
```
columns_to_drop = ['label','filename']

def split_dataset(df, columns_to_drop, test_size, random_state):
  label_encoder = preprocessing.LabelEncoder()

  df['label'] = label_encoder.fit_transform(df['label'])

  df_train, df_test = train_test_split(df, test_size=test_size, random_state=random_sta

  df_train2 = df_train.drop(columns_to_drop,axis=1)
  y_train2 = df_train['label'].to_numpy()

  df_test2 = df_test.drop(columns_to_drop,axis=1)
  y_test2 = df_test['label'].to_numpy()

  return df_train2, y_train2, df_test2, y_test2

def preprocess_dataset(df_train, df_test):

  standard_scaler = preprocessing.StandardScaler()
  df_train_scaled = standard_scaler.fit_transform(df_train)

  df_test_scaled = standard_scaler.transform(df_test)

  return df_train_scaled, df_test_scaled

X_train, y_train, X_test, y_test = split_dataset(df, columns_to_drop, test_size=0.3, ra

X_train_scaled, X_test_scaled = preprocess_dataset(X_train, X_test)
```

Question 4A

In [ ]:
```
neg_voice_record_df = pd.read_csv('Q4_neg_voice_record.csv')
pos_voice_record_df = pd.read_csv('Q4_pos_voice_record.csv')
threshold = 0.5
```

In [ ]:
```
neg_voice_record_df = neg_voice_record_df.drop(["filename"],axis = 1)
pos_voice_record_df = pos_voice_record_df.drop(["filename"], axis=1)
```

In [ ]:
```
def process_dataset(df_train, df_test):

  standard_scaler = preprocessing.StandardScaler()
  df_train_scaled = standard_scaler.fit_transform(df_train)

  df_test_scaled = standard_scaler.transform(df_test)

  return df_test_scaled


neg_voice_record_df_scaled = process_dataset(X_train, neg_voice_record_df)
pos_voice_record_df_scaled = process_dataset(X_train, pos_voice_record_df)
```

Question 4B

Do a model prediction on your sample test dataset with threshold = 0.5

In [ ]:
```
optimized_model = keras.models.load_model('optimized_model/')

neg_result_label = (optimized_model.predict(neg_voice_record_df_scaled)>threshold).asty
pos_result_label = (optimized_model.predict(pos_voice_record_df_scaled)>threshold).asty

data = {"Label": ["Negative Voice ", "Positive Voice"],"Result":[neg_result_label, pos_

data_df = pd.DataFrame.from_dict(data)
data_df
```

```
1/1 [==============================] - 1s 1s/step
1/1 [==============================] - 0s 16ms/step
```

Out[ ]:

	Label	Result
0	Negative Voice	[[0]]
1	Positive Voice	[[0]]

Question 4C

Identify most important features using SHAP

In [ ]:
```
tf.compat.v1.disable_v2_behavior()
```

```
WARNING:tensorflow:From c:\Users\JoeTe\AppData\Local\Programs\Python\Python310\lib\site
-packages\tensorflow\python\compat\v2_compat.py:107: disable_resource_variables (from t
ensorflow.python.ops.variable_scope) is deprecated and will be removed in a future vers
ion.
Instructions for updating:
non-resource variables are not supported in the long term
```

Retrieve 1000 samples

In [ ]:
```
X_train_sample = X_train_scaled[np.random.choice(len(X_train_scaled), 1000, replace=Fals
X_test_sample = X_test_scaled[np.random.choice(len(X_test_scaled), 1000, replace=False)
```

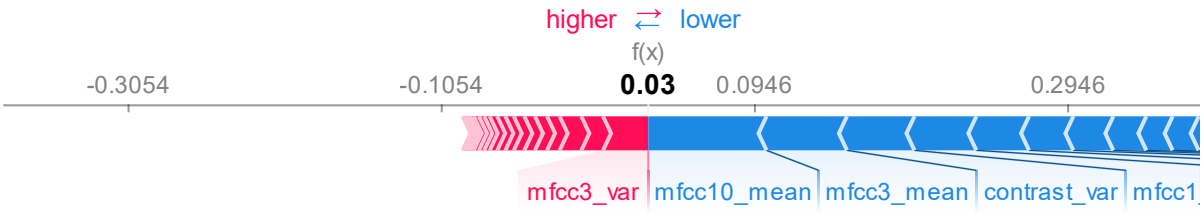
In [ ]:
```
model = keras.models.load_model('optimized_model/')
explainer = shap.DeepExplainer(model , X_train_sample)
```

```
keras is no longer supported, please use tf.keras instead.
Your TensorFlow version is newer than 2.4.0 and so graph support has been removed in ea
ger mode and some static graphs may not be supported. See PR #1483 for discussion.
WARNING:tensorflow:From c:\Users\JoeTe\AppData\Local\Programs\Python\Python310\lib\site
-packages\shap\explainers\tf_utils.py:28: The name tf.keras.backend.get_session is depr
ecated. Please use tf.compat.v1.keras.backend.get_session instead.
```

Force plot of neg voice record

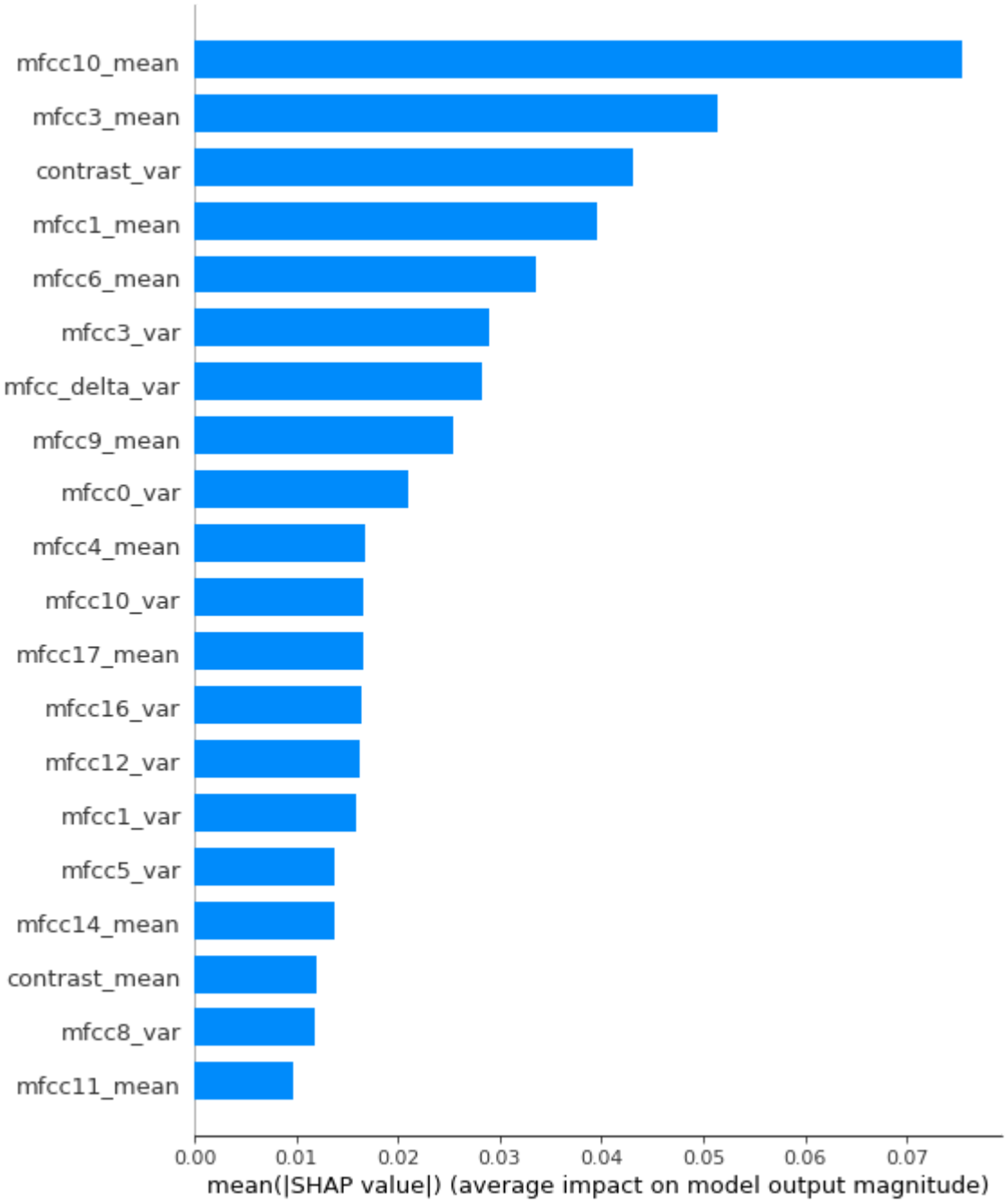
In [ ]:
```
shap_values = explainer.shap_values(neg_voice_record_df_scaled)
shap.force_plot(explainer.expected_value[0], shap_values[0][0], features = X_test.colum
```

Out[ ]:



Summary plot of neg voice record

In [ ]:
```
shap.summary_plot(shap_values[0], plot_type = 'bar', feature_names = X_test.columns)
```



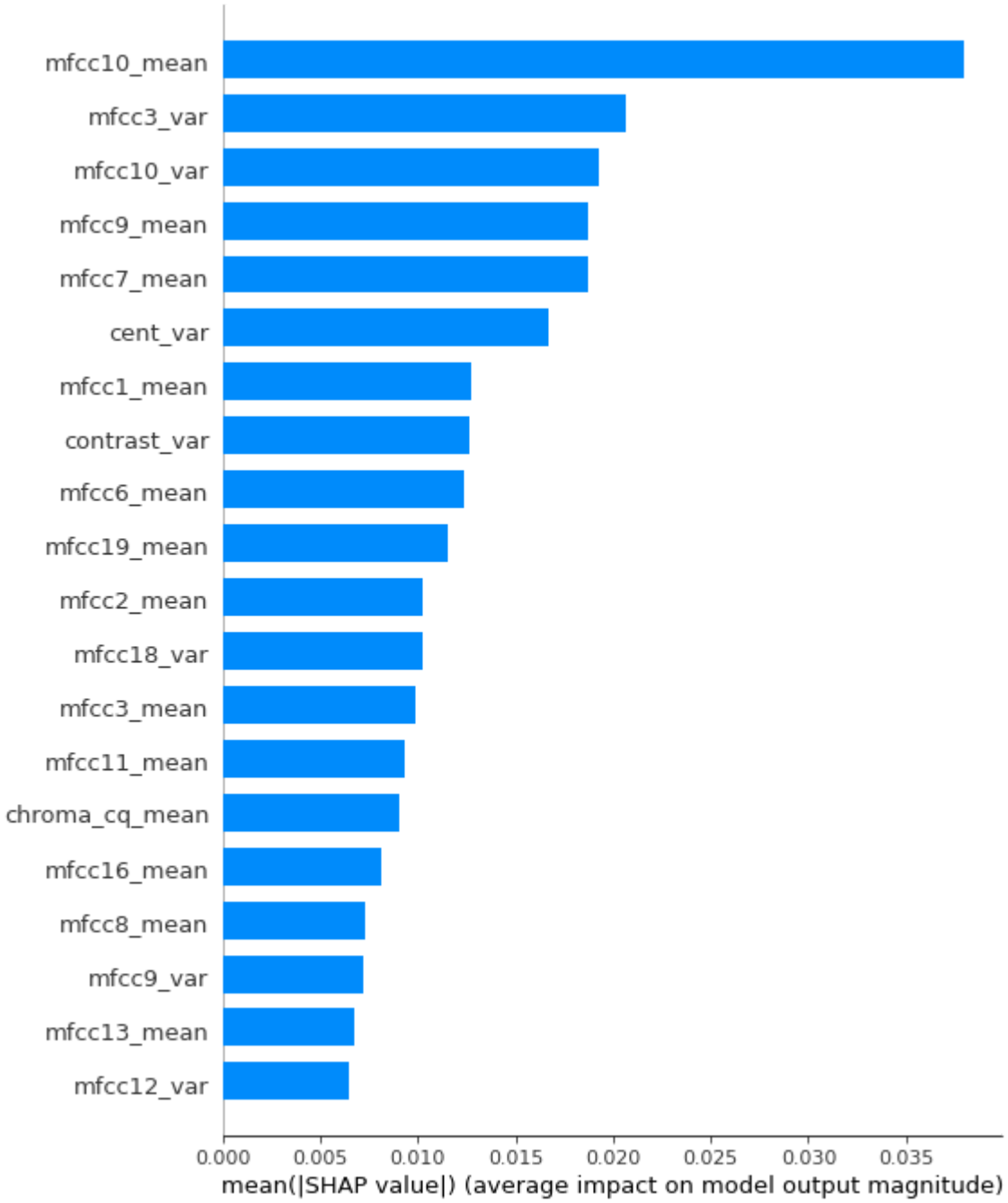
Force plot of pos voice record

```
In [ ]: shap_values = explainer.shap_values(pos_voice_record_df_scaled)
        shap.force_plot(explainer.expected_value[0], shap_values[0][0], features = X_test.columı
```

Out[ ]:

Summary plot of pos voice record

In [ ]:
```
shap.summary_plot(shap_values[0], plot_type = 'bar', feature_names = X_test.columns)
```



Observation

From the force plot of both the pos and neg_voice record, it shows the top most influential features that led to prediction value indicated. The red color features influence positivity(towards the predicted value) and the blue color influence negativity(away from the predicted value). With the force plot and summary, we can identify the magnitude of the features' impact to the resulting predicted value,

Analysis on Positive Voice Record

In the voice recording, we can identify that the features in red to be the most influential features in influencing the prediction to be closer to 1 (positive label) and in blue are the most influential features in influencing the predict to be closer to 0 (negative label)

From the summary plot, we can then identify the magnitude of the the features' impact to the resulting predicted value

In the positive voice recording where there is negative classification(0), we can see that mfcc10_mean, mfcc10_var are the top two most influential feature for a postive prediction and mfcc3_var along with mfcc9_mean are the top two most influential feature for a negative prediction. This shows us that users can observe that given a certain value to a input feature, the user would be able to observe it would fare in its influence towards the prediction outcome.

Discussion Points

1. Limitations of FFN

FFN is prone to overfitting and with given large number of parameters, the model will be more complex and could take a long time to train. In addition, due to the risk of overfitting, the model may lose the ability to generalize to new examples.

Also, feed-forward neural networks may have results that are difficult to interpret due to the complexitiy of the model's architecture.

2. Most impactful parameter

In terms of time taken for every epoch, the batch size is the most impactful parameter as from the table at Q2b, we notice that doubling the batch size shorterns the time taken for the final epoch significantly.

In terms of accuracy, the number of neurons in the first hidden layer is the most impactful paramater as from the table at Q3b, we notice that there is significant rise in accuracy when the number of neurons increases

3. Alternative approaches

We can use CNN model architecture for genre classfication as well. Similar to the assignment, we have to perform feature extraction and define the model architecture. In fact, we can experiment by adding more hidden layers so that the model is able to handle more complex tasks and learn the relationships between features

4. Other dataset

Analysing the audio waveforms to idenitify the species of the subject(animals). Perhaps more hidden layer is required for the model to learn the relationship between the features extracted from the audio of different species

Also, we can do speech enhancement. To improve the quality of the audio, we would need a very large and complex and neural network model which would need an increase in hidden layers and number of units.

5. Neural Network Ensemble

An ensemble of neural network can be done to achieve diversification in order to build models that can generalize better. Ensemble learning combines the prediction from multiple neural network models to reduce the variance of predictions and reduce generalization error.