In [ ]:
```
# General data analysis/plotting
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Data preprocessing
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

# Neural Net modules
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.callbacks import EarlyStopping
```

In [ ]:
```
df = pd.read_csv('boston.csv')
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   longitude           20640 non-null  float64
 1   latitude            20640 non-null  float64
 2   housing_median_age  20640 non-null  float64
 3   total_rooms         20640 non-null  float64
 4   total_bedrooms      20433 non-null  float64
 5   population          20640 non-null  float64
 6   households          20640 non-null  float64
 7   median_income       20640 non-null  float64
 8   median_house_value  20640 non-null  float64
 9   ocean_proximity     20640 non-null  object
dtypes: float64(9), object(1)
memory usage: 1.6+ MB
```

In [ ]:
```
df.head()
```

Out[ ]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	household:
0	-122.23	37.88	41.0	880.0	129.0	322.0	126.(
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.(
2	-122.24	37.85	52.0	1467.0	190.0	496.0	177.(
3	-122.25	37.85	52.0	1274.0	235.0	558.0	219.(
4	-122.25	37.85	52.0	1627.0	280.0	565.0	259.(

In [ ]:
```
df.corr()['median_house_value'].sort_values()
```

Out[ ]:
```
latitude             -0.144160
longitude            -0.045967
population           -0.024650
total_bedrooms        0.049686
households            0.065843
housing_median_age    0.105623
total_rooms           0.134153
median_income         0.688075
median_house_value    1.000000
Name: median_house_value, dtype: float64
```

In [ ]:
```
df.dropna(axis=0, inplace=True)
df = pd.get_dummies(df, columns=['ocean_proximity'])
```

In [ ]:
```
y = df['median_house_value']
X = df.drop('median_house_value', axis=1)
print(X.shape, y.shape)

# convert to numpy array
X = np.array(X)
y = np.array(y)

# split into X_train and X_test
# always split into X_train, X_test first THEN apply minmax scaler
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.2,
                                                    random_state=42)

# use minMax scaler
s_scaler = StandardScaler()
X_train = s_scaler.fit_transform(X_train)
X_test = s_scaler.transform(X_test)
```

```
(20433, 13) (20433,)
```

In [ ]:
```
model = Sequential()
model.add(Dense(256, input_shape=(X_train.shape[1],),input_dim = 13 ,activation='re
model.add(Dense(128, activation='relu'))
model.add(Dense(64, activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(1, activation='linear')) # output node
model.summary() # see what your model looks like

# compile the model
model.compile(optimizer='adam', loss='mean_squared_error', metrics=['mae'])
# es = EarlyStopping(monitor='val_loss',
#                    mode='min',
#                    patience=50,
#                    restore_best_weights = True)

hist =model.fit(X_train, y_train,
            validation_split=0.2,
                    # callbacks=[es],
            epochs=70,
            batch_size=32,
            verbose=1)
```

```
Model: "sequential_7"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_33 (Dense)            (None, 256)               3584

 dense_34 (Dense)            (None, 128)               32896

 dense_35 (Dense)            (None, 64)                8256

 dense_36 (Dense)            (None, 32)                2080

 dense_37 (Dense)            (None, 1)                 33

=================================================================
Total params: 46,849
Trainable params: 46,849
Non-trainable params: 0
_________________________________________________________________
Epoch 1/70
409/409 [==============================] - 5s 4ms/step - loss: 24274036736.0000 -
mae: 114201.8828 - val_loss: 6917057536.0000 - val_mae: 58284.5391
Epoch 2/70
409/409 [==============================] - 2s 4ms/step - loss: 5623338496.0000 - m
ae: 53030.3320 - val_loss: 5379912704.0000 - val_mae: 51872.2773
Epoch 3/70
409/409 [==============================] - 1s 3ms/step - loss: 4915969536.0000 - m
ae: 49851.2656 - val_loss: 4975810560.0000 - val_mae: 50313.9609
Epoch 4/70
409/409 [==============================] - 1s 3ms/step - loss: 4655184896.0000 - m
ae: 48411.6836 - val_loss: 4783638016.0000 - val_mae: 48325.8047
Epoch 5/70
409/409 [==============================] - 2s 4ms/step - loss: 4520173568.0000 - m
ae: 47550.9375 - val_loss: 4686906880.0000 - val_mae: 47503.1328
Epoch 6/70
409/409 [==============================] - 2s 4ms/step - loss: 4437259264.0000 - m
ae: 47009.6992 - val_loss: 4577262080.0000 - val_mae: 47612.0703
Epoch 7/70
409/409 [==============================] - 2s 4ms/step - loss: 4378068992.0000 - m
ae: 46830.8516 - val_loss: 4522796032.0000 - val_mae: 47136.4609
Epoch 8/70
409/409 [==============================] - 1s 3ms/step - loss: 4339048448.0000 - m
ae: 46461.2188 - val_loss: 4456461312.0000 - val_mae: 47471.6367
Epoch 9/70
409/409 [==============================] - 1s 4ms/step - loss: 4303913472.0000 - m
ae: 46286.0664 - val_loss: 4422327808.0000 - val_mae: 46551.8672
Epoch 10/70
409/409 [==============================] - 1s 3ms/step - loss: 4280538112.0000 - m
ae: 46029.6914 - val_loss: 4427043840.0000 - val_mae: 47133.0000
Epoch 11/70
409/409 [==============================] - 1s 3ms/step - loss: 4259996672.0000 - m
ae: 46016.8008 - val_loss: 4403410432.0000 - val_mae: 46114.2148
Epoch 12/70
409/409 [==============================] - 1s 3ms/step - loss: 4234523136.0000 - m
ae: 45667.0234 - val_loss: 4353989632.0000 - val_mae: 46168.0195
Epoch 13/70
409/409 [==============================] - 1s 3ms/step - loss: 4203964672.0000 - m
ae: 45597.8945 - val_loss: 4347664384.0000 - val_mae: 46370.9648
Epoch 14/70
409/409 [==============================] - 2s 4ms/step - loss: 4199000320.0000 - m
ae: 45502.2227 - val_loss: 4355822080.0000 - val_mae: 46640.8281
```

```
Epoch 15/70
409/409 [==============================] - 1s 3ms/step - loss: 4178278400.0000 - m
ae: 45446.6523 - val_loss: 4364462080.0000 - val_mae: 45258.4609
Epoch 16/70
409/409 [==============================] - 1s 3ms/step - loss: 4173323776.0000 - m
ae: 45314.0742 - val_loss: 4309645312.0000 - val_mae: 45316.6289
Epoch 17/70
409/409 [==============================] - 2s 4ms/step - loss: 4161819392.0000 - m
ae: 45212.4180 - val_loss: 4285344512.0000 - val_mae: 45982.6953
Epoch 18/70
409/409 [==============================] - 1s 3ms/step - loss: 4142855424.0000 - m
ae: 45096.4492 - val_loss: 4307572224.0000 - val_mae: 45539.5664
Epoch 19/70
409/409 [==============================] - 2s 5ms/step - loss: 4148440832.0000 - m
ae: 45154.0000 - val_loss: 4263058176.0000 - val_mae: 45173.0938
Epoch 20/70
409/409 [==============================] - 2s 4ms/step - loss: 4125058304.0000 - m
ae: 44957.3906 - val_loss: 4235993344.0000 - val_mae: 45444.3398
Epoch 21/70
409/409 [==============================] - 2s 5ms/step - loss: 4124503040.0000 - m
ae: 45043.2891 - val_loss: 4268925696.0000 - val_mae: 44802.5195
Epoch 22/70
409/409 [==============================] - 2s 6ms/step - loss: 4107105280.0000 - m
ae: 44833.5352 - val_loss: 4320941568.0000 - val_mae: 45099.8984
Epoch 23/70
409/409 [==============================] - 2s 6ms/step - loss: 4109408256.0000 - m
ae: 44907.6484 - val_loss: 4182974976.0000 - val_mae: 45293.5352
Epoch 24/70
409/409 [==============================] - 2s 6ms/step - loss: 4091773440.0000 - m
ae: 44754.4805 - val_loss: 4186414336.0000 - val_mae: 45187.4219
Epoch 25/70
409/409 [==============================] - 3s 7ms/step - loss: 4086612224.0000 - m
ae: 44744.9805 - val_loss: 4153266432.0000 - val_mae: 44700.7969
Epoch 26/70
409/409 [==============================] - 2s 6ms/step - loss: 4063900160.0000 - m
ae: 44641.8203 - val_loss: 4108521728.0000 - val_mae: 45028.0625
Epoch 27/70
409/409 [==============================] - 2s 5ms/step - loss: 4057838080.0000 - m
ae: 44502.5547 - val_loss: 4092457216.0000 - val_mae: 45597.2852
Epoch 28/70
409/409 [==============================] - 2s 5ms/step - loss: 4044464640.0000 - m
ae: 44418.7695 - val_loss: 4066526208.0000 - val_mae: 44522.0156
Epoch 29/70
409/409 [==============================] - 2s 5ms/step - loss: 4039573760.0000 - m
ae: 44588.5039 - val_loss: 4148489472.0000 - val_mae: 44201.4492
Epoch 30/70
409/409 [==============================] - 2s 4ms/step - loss: 4034781440.0000 - m
ae: 44340.6562 - val_loss: 4118276864.0000 - val_mae: 44278.3086
Epoch 31/70
409/409 [==============================] - 2s 4ms/step - loss: 4030390272.0000 - m
ae: 44352.0312 - val_loss: 4034901760.0000 - val_mae: 44673.6719
Epoch 32/70
409/409 [==============================] - 2s 4ms/step - loss: 4019687424.0000 - m
ae: 44452.9844 - val_loss: 4038199808.0000 - val_mae: 44221.9609
Epoch 33/70
409/409 [==============================] - 2s 4ms/step - loss: 4024622080.0000 - m
ae: 44231.2344 - val_loss: 4035862528.0000 - val_mae: 44333.8516
Epoch 34/70
409/409 [==============================] - 2s 4ms/step - loss: 4004206592.0000 - m
ae: 44203.8789 - val_loss: 4036334336.0000 - val_mae: 44156.7109
Epoch 35/70
```

```
409/409 [==============================] - 2s 4ms/step - loss: 4014968832.0000 - m
ae: 44236.4219 - val_loss: 3986566400.0000 - val_mae: 44355.0547
Epoch 36/70
409/409 [==============================] - 2s 5ms/step - loss: 3973395712.0000 - m
ae: 44049.5273 - val_loss: 4030490880.0000 - val_mae: 43969.3164
Epoch 37/70
409/409 [==============================] - 2s 4ms/step - loss: 3962467072.0000 - m
ae: 43875.3086 - val_loss: 3955189504.0000 - val_mae: 44289.5664
Epoch 38/70
409/409 [==============================] - 2s 4ms/step - loss: 3931886336.0000 - m
ae: 43662.4102 - val_loss: 3921278208.0000 - val_mae: 43623.4023
Epoch 39/70
409/409 [==============================] - 2s 4ms/step - loss: 3889708288.0000 - m
ae: 43545.1367 - val_loss: 3906986496.0000 - val_mae: 43703.6680
Epoch 40/70
409/409 [==============================] - 2s 4ms/step - loss: 3874746112.0000 - m
ae: 43393.6602 - val_loss: 3907326464.0000 - val_mae: 43019.7539
Epoch 41/70
409/409 [==============================] - 2s 4ms/step - loss: 3836909568.0000 - m
ae: 43138.1211 - val_loss: 3830507520.0000 - val_mae: 43691.4102
Epoch 42/70
409/409 [==============================] - 2s 4ms/step - loss: 3811752960.0000 - m
ae: 43025.1797 - val_loss: 3823808512.0000 - val_mae: 42736.0195
Epoch 43/70
409/409 [==============================] - 2s 5ms/step - loss: 3784745472.0000 - m
ae: 42864.2734 - val_loss: 3791889664.0000 - val_mae: 42630.1016
Epoch 44/70
409/409 [==============================] - 2s 4ms/step - loss: 3765963520.0000 - m
ae: 42766.3711 - val_loss: 3805528320.0000 - val_mae: 43583.4609
Epoch 45/70
409/409 [==============================] - 1s 3ms/step - loss: 3728362240.0000 - m
ae: 42606.4141 - val_loss: 3808962048.0000 - val_mae: 42433.3242
Epoch 46/70
409/409 [==============================] - 1s 4ms/step - loss: 3693335296.0000 - m
ae: 42396.9805 - val_loss: 3694614016.0000 - val_mae: 42211.8594
Epoch 47/70
409/409 [==============================] - 1s 3ms/step - loss: 3664879872.0000 - m
ae: 42263.3906 - val_loss: 3691926528.0000 - val_mae: 42199.6328
Epoch 48/70
409/409 [==============================] - 1s 3ms/step - loss: 3640350208.0000 - m
ae: 42184.6719 - val_loss: 3628929024.0000 - val_mae: 42129.1211
Epoch 49/70
409/409 [==============================] - 1s 3ms/step - loss: 3595078656.0000 - m
ae: 41791.3984 - val_loss: 3618345216.0000 - val_mae: 42017.4453
Epoch 50/70
409/409 [==============================] - 1s 3ms/step - loss: 3563447552.0000 - m
ae: 41715.0430 - val_loss: 3586153984.0000 - val_mae: 41846.2148
Epoch 51/70
409/409 [==============================] - 1s 3ms/step - loss: 3514232576.0000 - m
ae: 41453.5430 - val_loss: 3605815808.0000 - val_mae: 41578.6367
Epoch 52/70
409/409 [==============================] - 1s 3ms/step - loss: 3515499520.0000 - m
ae: 41376.8555 - val_loss: 3562244096.0000 - val_mae: 41552.6641
Epoch 53/70
409/409 [==============================] - 1s 3ms/step - loss: 3451065344.0000 - m
ae: 40934.7461 - val_loss: 3517224960.0000 - val_mae: 41498.3047
Epoch 54/70
409/409 [==============================] - 1s 3ms/step - loss: 3417441024.0000 - m
ae: 40683.4688 - val_loss: 3558119424.0000 - val_mae: 40717.1992
Epoch 55/70
409/409 [==============================] - 1s 3ms/step - loss: 3386411264.0000 - m
```

```
ae: 40528.3906 - val_loss: 3500677120.0000 - val_mae: 40677.4492
Epoch 56/70
409/409 [==============================] - 1s 3ms/step - loss: 3362473472.0000 - m
ae: 40346.7344 - val_loss: 3462355200.0000 - val_mae: 41052.4414
Epoch 57/70
409/409 [==============================] - 1s 3ms/step - loss: 3339944192.0000 - m
ae: 40248.0117 - val_loss: 3442449408.0000 - val_mae: 40286.1602
Epoch 58/70
409/409 [==============================] - 1s 3ms/step - loss: 3334477056.0000 - m
ae: 40051.9961 - val_loss: 3431271168.0000 - val_mae: 41194.8438
Epoch 59/70
409/409 [==============================] - 1s 3ms/step - loss: 3281990912.0000 - m
ae: 39774.0039 - val_loss: 3525592576.0000 - val_mae: 40266.9023
Epoch 60/70
409/409 [==============================] - 1s 3ms/step - loss: 3272454400.0000 - m
ae: 39610.4023 - val_loss: 3373454336.0000 - val_mae: 39978.1328
Epoch 61/70
409/409 [==============================] - 1s 3ms/step - loss: 3262034176.0000 - m
ae: 39480.6797 - val_loss: 3365679360.0000 - val_mae: 40013.2812
Epoch 62/70
409/409 [==============================] - 1s 3ms/step - loss: 3239940864.0000 - m
ae: 39370.3398 - val_loss: 3386418944.0000 - val_mae: 40480.4297
Epoch 63/70
409/409 [==============================] - 2s 5ms/step - loss: 3211299328.0000 - m
ae: 39153.2734 - val_loss: 3376682240.0000 - val_mae: 39808.6992
Epoch 64/70
409/409 [==============================] - 2s 5ms/step - loss: 3209287936.0000 - m
ae: 39020.8203 - val_loss: 3385104640.0000 - val_mae: 39395.4219
Epoch 65/70
409/409 [==============================] - 2s 5ms/step - loss: 3174726656.0000 - m
ae: 38856.9883 - val_loss: 3298547968.0000 - val_mae: 39523.1172
Epoch 66/70
409/409 [==============================] - 2s 4ms/step - loss: 3165601024.0000 - m
ae: 38765.8984 - val_loss: 3347460608.0000 - val_mae: 38800.0898
Epoch 67/70
409/409 [==============================] - 2s 4ms/step - loss: 3157187072.0000 - m
ae: 38573.0469 - val_loss: 3307460864.0000 - val_mae: 39303.7344
Epoch 68/70
409/409 [==============================] - 2s 4ms/step - loss: 3139415040.0000 - m
ae: 38580.5430 - val_loss: 3314692096.0000 - val_mae: 39787.0898
Epoch 69/70
409/409 [==============================] - 2s 4ms/step - loss: 3128180736.0000 - m
ae: 38383.4648 - val_loss: 3243513088.0000 - val_mae: 39078.0156
Epoch 70/70
409/409 [==============================] - 2s 4ms/step - loss: 3099548928.0000 - m
ae: 38199.7266 - val_loss: 3346754560.0000 - val_mae: 38526.4961
```

In [ ]:

```
hist.history.keys()
```

Out[ ]:

```
dict_keys(['loss', 'mae', 'val_loss', 'val_mae'])
```

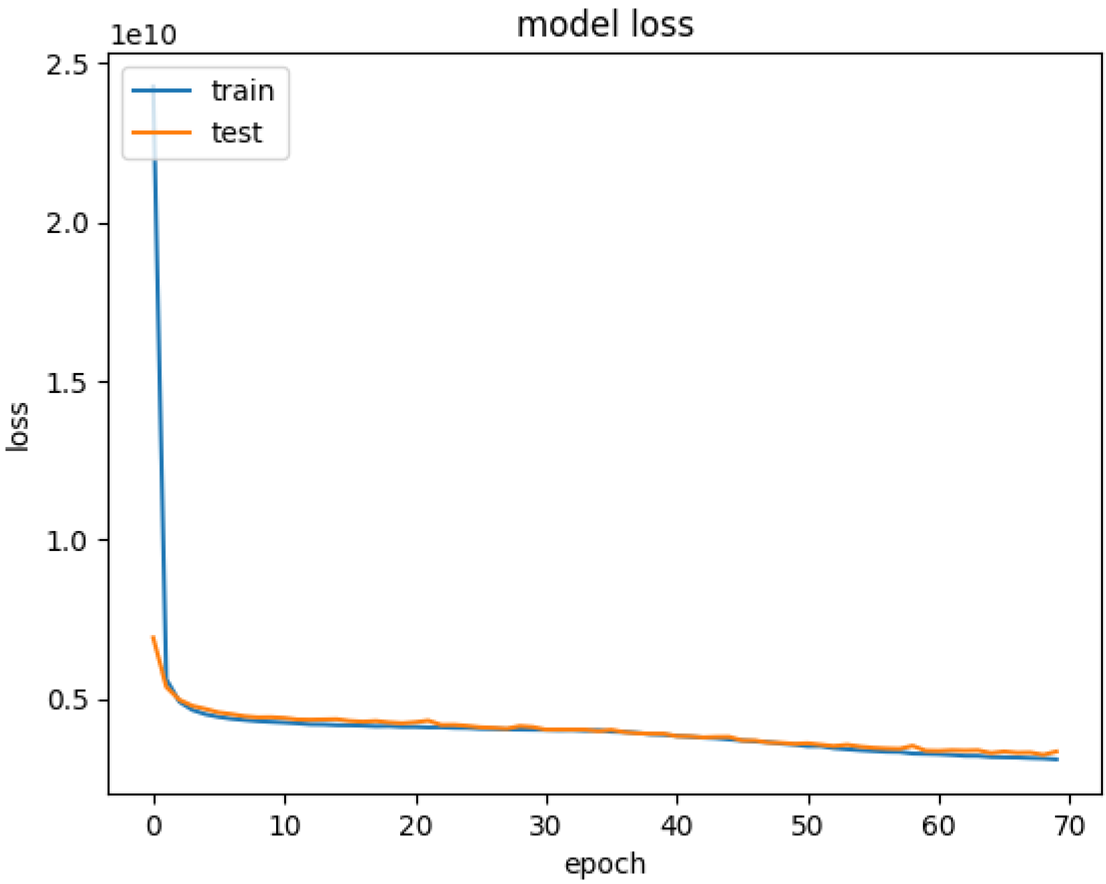
In [ ]:

```
pred = model.predict(X_test)
trainpreds = model.predict(X_train)

from sklearn.metrics import mean_absolute_error
print(mean_absolute_error(y_train, trainpreds)) # train
print(mean_absolute_error(y_test, pred)) # test
```

```
128/128 [==============================] - 0s 2ms/step
511/511 [==============================] - 1s 3ms/step
37641.33522132443
39362.079361772085
```

In [ ]:

```python
# plotting validation and training error
import matplotlib.pyplot as plt
import seaborn as sns

plt.plot(hist.history['loss'])
plt.plot(hist.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```



In [ ]:

```python
df = pd.read_csv("https://github.com/YBI-Foundation/Dataset/raw/main/Titanic.csv")
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1309 entries, 0 to 1308
Data columns (total 14 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   pclass     1309 non-null   int64  
 1   survived   1309 non-null   int64  
 2   name       1309 non-null   object 
 3   sex        1309 non-null   object 
 4   age        1046 non-null   float64
 5   sibsp      1309 non-null   int64  
 6   parch      1309 non-null   int64  
 7   ticket     1309 non-null   object 
 8   fare       1308 non-null   float64
 9   cabin      295 non-null    object 
 10  embarked   1307 non-null   object 
 11  boat       486 non-null    object 
 12  body       121 non-null    float64
 13  home.dest  745 non-null    object 
dtypes: float64(3), int64(4), object(7)
memory usage: 143.3+ KB
```

In [ ]:
```
df.drop(['body','home.dest','boat','cabin'],axis=1,inplace=True)
df.isna().sum()
```

Out[ ]:
```
pclass        0
survived      0
name          0
sex           0
age         263
sibsp         0
parch         0
ticket        0
fare          1
embarked      2
dtype: int64
```

In [ ]:
```
df['age'].fillna(round(df['age'].mean()),inplace=True)
df['fare'].fillna(df['fare'].median(),inplace=True)
df['embarked'].fillna(df['embarked'].mode()[0],inplace=True)
```

In [ ]:
```
df.replace({'sex':{'male':0,'female':1}, 'embarked':{'S':0,'C':1,'Q':2}}, inplace=T
```

In [ ]:
```
X = df.drop(columns = ['name','ticket','survived'],axis=1)
y=df['survived']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_s
```

In [ ]:
```
model = Sequential()
model.add(Dense(64, input_shape=(X_train.shape[1],), activation='relu')) # (feature
model.add(Dense(32, activation='relu'))
model.add(Dense(1, activation='sigmoid')) # output node

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=50, batch_size
```

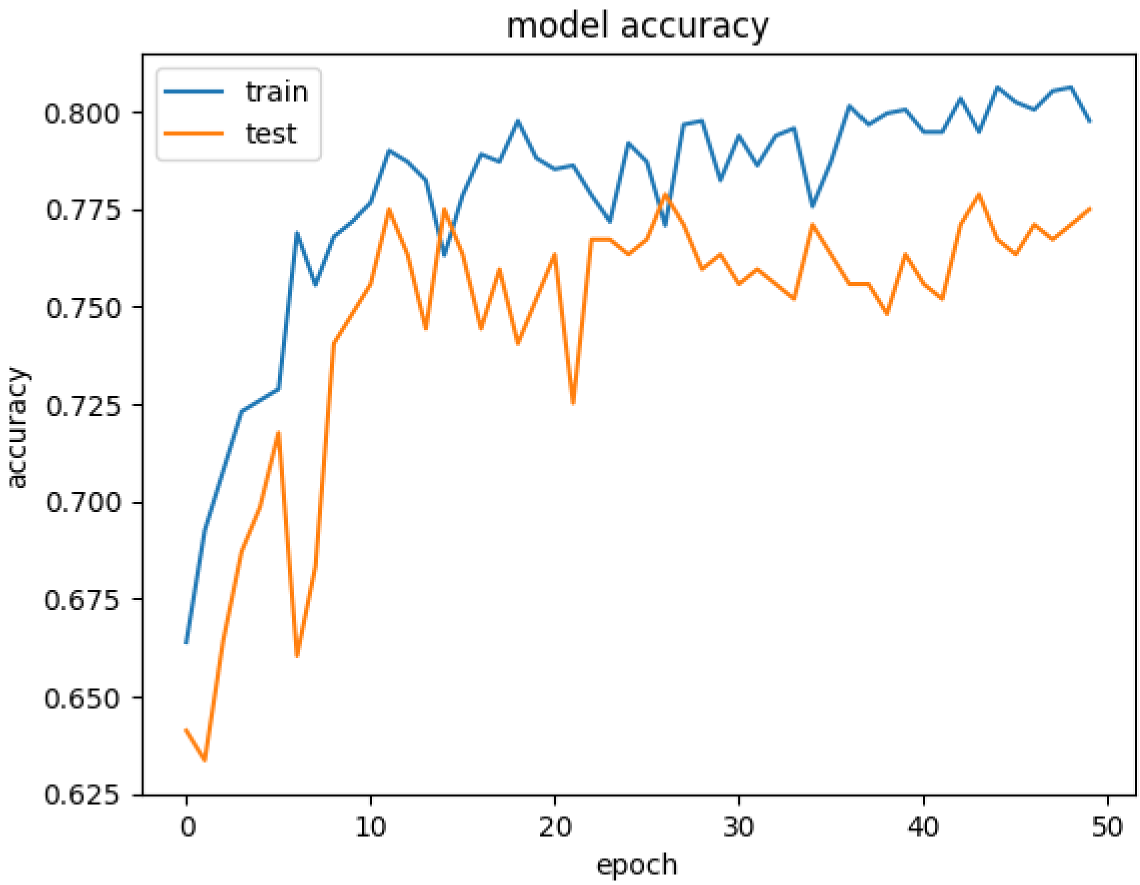
```
Epoch 1/50
66/66 [==============================] - 2s 14ms/step - loss: 0.6732 - accuracy:
0.6638 - val_loss: 0.6384 - val_accuracy: 0.6412
Epoch 2/50
66/66 [==============================] - 0s 4ms/step - loss: 0.6203 - accuracy: 0.
6925 - val_loss: 0.6240 - val_accuracy: 0.6336
Epoch 3/50
66/66 [==============================] - 0s 4ms/step - loss: 0.6307 - accuracy: 0.
7077 - val_loss: 0.5916 - val_accuracy: 0.6641
Epoch 4/50
66/66 [==============================] - 0s 6ms/step - loss: 0.5602 - accuracy: 0.
7230 - val_loss: 0.5630 - val_accuracy: 0.6870
Epoch 5/50
66/66 [==============================] - 0s 4ms/step - loss: 0.5608 - accuracy: 0.
7259 - val_loss: 0.5964 - val_accuracy: 0.6985
Epoch 6/50
66/66 [==============================] - 0s 4ms/step - loss: 0.5534 - accuracy: 0.
7287 - val_loss: 0.5391 - val_accuracy: 0.7176
Epoch 7/50
66/66 [==============================] - 0s 4ms/step - loss: 0.5135 - accuracy: 0.
7689 - val_loss: 0.6156 - val_accuracy: 0.6603
Epoch 8/50
66/66 [==============================] - 0s 4ms/step - loss: 0.5408 - accuracy: 0.
7555 - val_loss: 0.5777 - val_accuracy: 0.6832
Epoch 9/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4987 - accuracy: 0.
7679 - val_loss: 0.5157 - val_accuracy: 0.7405
Epoch 10/50
66/66 [==============================] - 0s 4ms/step - loss: 0.5094 - accuracy: 0.
7717 - val_loss: 0.4928 - val_accuracy: 0.7481
Epoch 11/50
66/66 [==============================] - 0s 4ms/step - loss: 0.5016 - accuracy: 0.
7765 - val_loss: 0.4931 - val_accuracy: 0.7557
Epoch 12/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4809 - accuracy: 0.
7899 - val_loss: 0.4958 - val_accuracy: 0.7748
Epoch 13/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4836 - accuracy: 0.
7870 - val_loss: 0.4862 - val_accuracy: 0.7634
Epoch 14/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4853 - accuracy: 0.
7822 - val_loss: 0.5542 - val_accuracy: 0.7443
Epoch 15/50
66/66 [==============================] - 0s 4ms/step - loss: 0.5104 - accuracy: 0.
7631 - val_loss: 0.5176 - val_accuracy: 0.7748
Epoch 16/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4832 - accuracy: 0.
7784 - val_loss: 0.5131 - val_accuracy: 0.7634
Epoch 17/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4719 - accuracy: 0.
7889 - val_loss: 0.4950 - val_accuracy: 0.7443
Epoch 18/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4739 - accuracy: 0.
7870 - val_loss: 0.4888 - val_accuracy: 0.7595
Epoch 19/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4744 - accuracy: 0.
7975 - val_loss: 0.5013 - val_accuracy: 0.7405
Epoch 20/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4687 - accuracy: 0.
7880 - val_loss: 0.4943 - val_accuracy: 0.7519
Epoch 21/50
```

```
66/66 [==============================] - 0s 4ms/step - loss: 0.4649 - accuracy: 0.
7851 - val_loss: 0.4851 - val_accuracy: 0.7634
Epoch 22/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4754 - accuracy: 0.
7861 - val_loss: 0.5944 - val_accuracy: 0.7252
Epoch 23/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4864 - accuracy: 0.
7784 - val_loss: 0.5238 - val_accuracy: 0.7672
Epoch 24/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4983 - accuracy: 0.
7717 - val_loss: 0.5077 - val_accuracy: 0.7672
Epoch 25/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4913 - accuracy: 0.
7918 - val_loss: 0.4887 - val_accuracy: 0.7634
Epoch 26/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4690 - accuracy: 0.
7870 - val_loss: 0.5741 - val_accuracy: 0.7672
Epoch 27/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4753 - accuracy: 0.
7708 - val_loss: 0.5120 - val_accuracy: 0.7786
Epoch 28/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4543 - accuracy: 0.
7966 - val_loss: 0.5287 - val_accuracy: 0.7710
Epoch 29/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4680 - accuracy: 0.
7975 - val_loss: 0.4848 - val_accuracy: 0.7595
Epoch 30/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4917 - accuracy: 0.
7822 - val_loss: 0.5414 - val_accuracy: 0.7634
Epoch 31/50
66/66 [==============================] - 0s 5ms/step - loss: 0.4587 - accuracy: 0.
7937 - val_loss: 0.4912 - val_accuracy: 0.7557
Epoch 32/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4647 - accuracy: 0.
7861 - val_loss: 0.5383 - val_accuracy: 0.7595
Epoch 33/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4462 - accuracy: 0.
7937 - val_loss: 0.5177 - val_accuracy: 0.7557
Epoch 34/50
66/66 [==============================] - 0s 3ms/step - loss: 0.4572 - accuracy: 0.
7956 - val_loss: 0.4946 - val_accuracy: 0.7519
Epoch 35/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4695 - accuracy: 0.
7755 - val_loss: 0.5363 - val_accuracy: 0.7710
Epoch 36/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4558 - accuracy: 0.
7870 - val_loss: 0.5060 - val_accuracy: 0.7634
Epoch 37/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4417 - accuracy: 0.
8013 - val_loss: 0.4888 - val_accuracy: 0.7557
Epoch 38/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4563 - accuracy: 0.
7966 - val_loss: 0.5019 - val_accuracy: 0.7557
Epoch 39/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4569 - accuracy: 0.
7994 - val_loss: 0.5196 - val_accuracy: 0.7481
Epoch 40/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4568 - accuracy: 0.
8004 - val_loss: 0.5168 - val_accuracy: 0.7634
Epoch 41/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4396 - accuracy: 0.
```

```
7947 - val_loss: 0.4904 - val_accuracy: 0.7557
Epoch 42/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4425 - accuracy: 0.
7947 - val_loss: 0.5046 - val_accuracy: 0.7519
Epoch 43/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4419 - accuracy: 0.
8032 - val_loss: 0.4980 - val_accuracy: 0.7710
Epoch 44/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4408 - accuracy: 0.
7947 - val_loss: 0.4936 - val_accuracy: 0.7786
Epoch 45/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4404 - accuracy: 0.
8061 - val_loss: 0.5278 - val_accuracy: 0.7672
Epoch 46/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4392 - accuracy: 0.
8023 - val_loss: 0.4881 - val_accuracy: 0.7634
Epoch 47/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4465 - accuracy: 0.
8004 - val_loss: 0.5276 - val_accuracy: 0.7710
Epoch 48/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4457 - accuracy: 0.
8052 - val_loss: 0.4998 - val_accuracy: 0.7672
Epoch 49/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4375 - accuracy: 0.
8061 - val_loss: 0.5007 - val_accuracy: 0.7710
Epoch 50/50
66/66 [==============================] - 0s 4ms/step - loss: 0.4519 - accuracy: 0.
7975 - val_loss: 0.4957 - val_accuracy: 0.7748
```

Out[ ]:
```
<keras.callbacks.History at 0x22f455c0c40>
```

In [ ]:
```
# plot the accuracy
plt.plot(model.history.history['accuracy'])
plt.plot(model.history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```


model accuracy
train
test
0.800
0.775
0.750
0.725
0.700
0.675
0.650
0.625
accuracy
0
10
20
30
40
50
epoch

In [ ]: