

Lymphography Data Set- Neural Networks

```
In [1]: ##### Standard Libraries #####
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

sns.set_style("whitegrid")
sns.set_context("poster")

%matplotlib inline
```

EDA

```
In [2]: ### List columns names based on the description
col_names = ['class', 'lymphatics', 'block of affere', 'bl. of lymph. c', 'bl. of ly
    'extravasates', 'regeneration of', 'early uptake in', 'lym.nodes dimin', 'lym.nodes
    'changes in lym.', 'defect in node', 'changes in node', 'changes in stru', 'special
    'dislocation of', 'exclusion of no', 'no. of nodes in']
```

```
In [3]: ### Load the data
df = pd.read_csv(r"C:\Users\ADMIN\Desktop\Projects\Data sets\lymphography.csv", name
print("Size of dataset:", df.shape)
df.head()
```

Size of dataset: (148, 19)

In [4]: `df.describe()`

	class	lymphatics	block of affere	bl. of lymph. c	bl. of lymph. s	by pass	extravasates	regene
50%	2.000000	3.000000	2.000000	1.000000	1.000000	1.000000	2.000000	1.0
75%	3.000000	3.000000	2.000000	1.000000	1.000000	1.000000	2.000000	1.0
max	4.000000	4.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.0

In [5]: `df["class"].value_counts()`

Out[5]:

2	81
3	61
4	4
1	2

Name: class, dtype: int64

In [6]: `df.isnull().sum()`

Out[6]:

class	0
lymphatics	0
block of affere	0
bl. of lymph. c	0
bl. of lymph. s	0
by pass	0
extravasates	0
regeneration of	0
early uptake in	0
lym.nodes dimin	0
lym.nodes enlar	0
changes in lym.	0
defect in node	0
changes in node	0
changes in stru	0
special forms	0
dislocation of	0
exclusion of no	0
no. of nodes in	0

dtype: int64

In [7]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 148 entries, 0 to 147
Data columns (total 19 columns):
 #   Column           Non-Null Count  Dtype  
 --- 
 0   class            148 non-null    int64  
 1   lymphatics       148 non-null    int64  
 2   block of affere 148 non-null    int64  
 3   bl. of lymph. c  148 non-null    int64  
 4   bl. of lymph. s  148 non-null    int64  
 5   by pass          148 non-null    int64  
 6   extravasates    148 non-null    int64  
 7   regeneration of 148 non-null    int64  
 8   early uptake in 148 non-null    int64  
 9   lym.nodes dimin 148 non-null    int64  
 10  lym.nodes enlar 148 non-null    int64  
 11  changes in lym. 148 non-null    int64  
 12  defect in node  148 non-null    int64
```

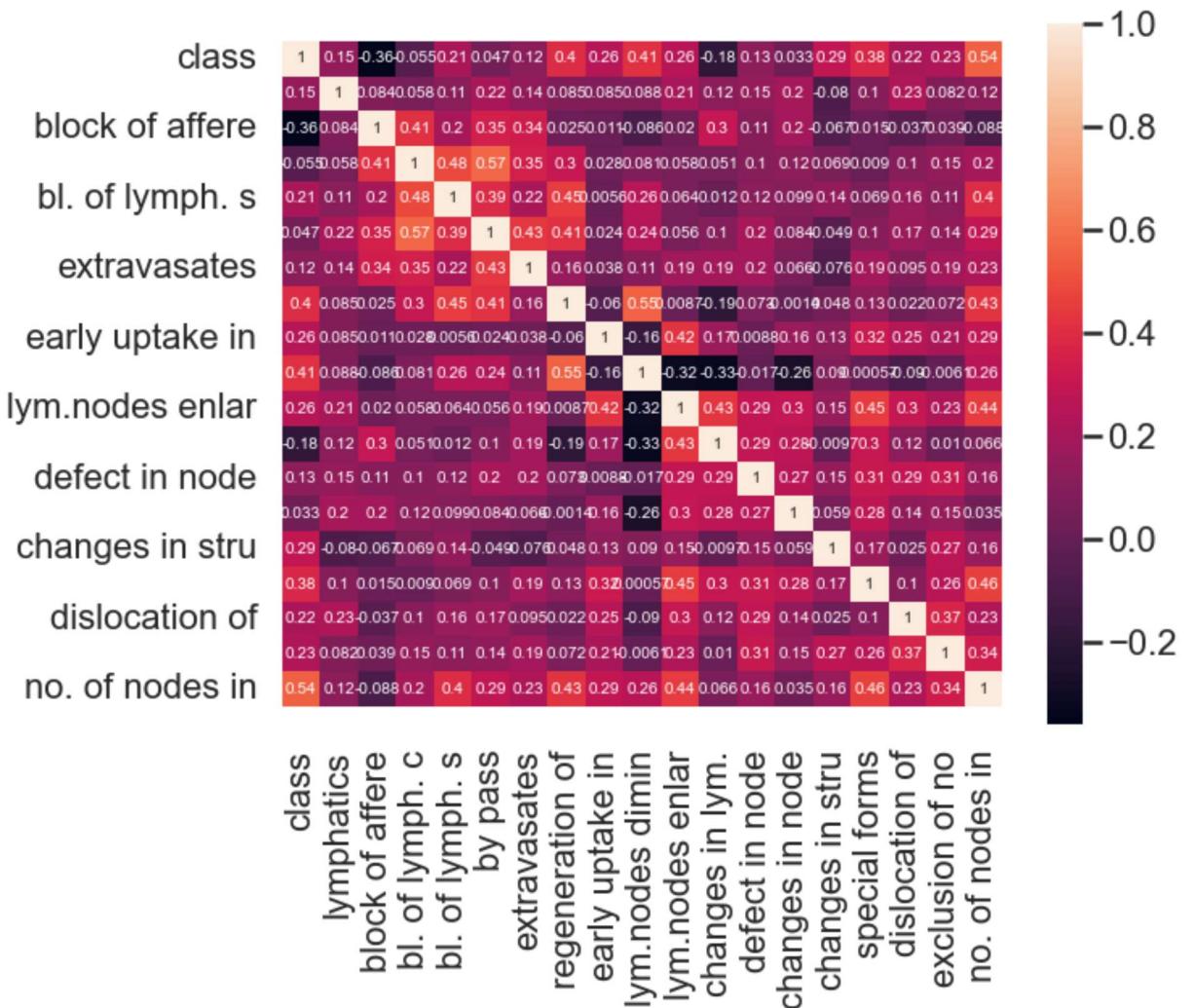
```

13 changes in node    148 non-null      int64
14 changes in stru   148 non-null      int64
15 special forms    148 non-null      int64
16 dislocation of   148 non-null      int64
17 exclusion of no  148 non-null      int64
18 no. of nodes in  148 non-null      int64
dtypes: int64(19)
memory usage: 22.1 KB

```

In [8]:

```
#Heatmap to show the correlation between various variables of the dataset
plt.figure(figsize=(10, 8))
cor = df.corr()
ax = sns.heatmap(cor, annot=True)
bottom, top = ax.get_ylim()
ax.set_ylim(bottom + 0.5, top - 0.5)
plt.show()
```



In [9]:

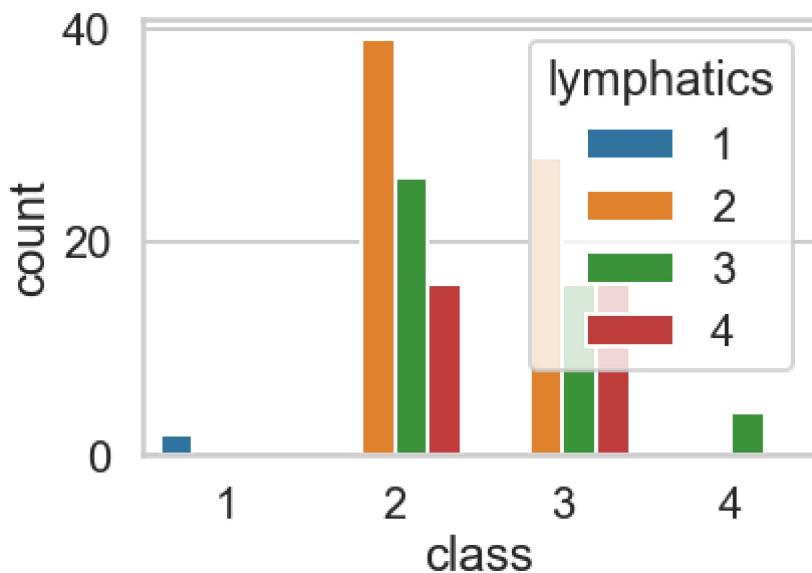
```
sns.countplot(df['class'], hue=df['lymphatics'])
```

C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```

Out[9]:

```
<AxesSubplot:xlabel='class', ylabel='count'>
```

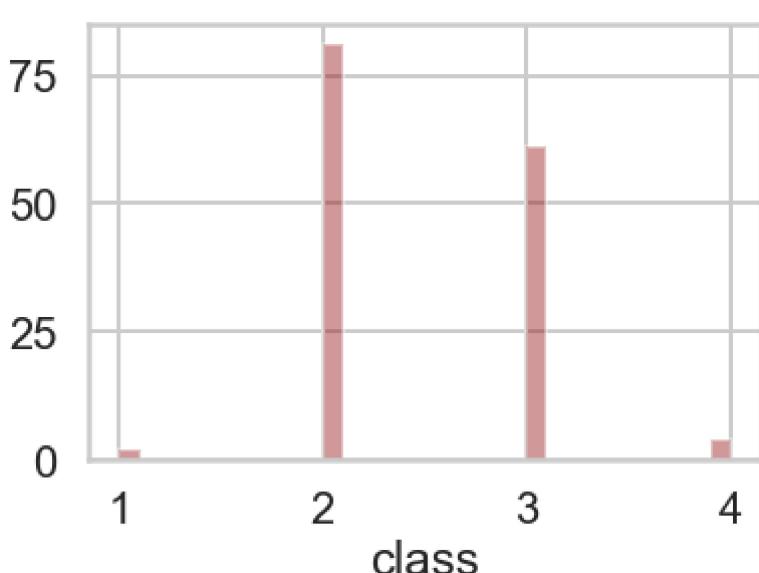


```
In [10]: sns.distplot(df['class'], kde=False, color='darkred', bins=30)
```

C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

```
    warnings.warn(msg, FutureWarning)
```

```
Out[10]: <AxesSubplot:xlabel='class'>
```



Test the model

```
In [11]: from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

```
In [12]: X = df.drop(columns=['class'])
y = df['class']

x_train, x_test, y_train, y_test = train_test_split(X, y, random_state=100, test_size=0.3,
```

```
In [13]: model = LogisticRegression()
model.fit(x_train,y_train)
pred = model.predict(x_test)
print('Accuracy ',accuracy_score(y_test,pred))
print(classification_report(y_test,pred))
sns.heatmap(confusion_matrix(y_test,pred),annot=True,fmt=' .2g ')
```

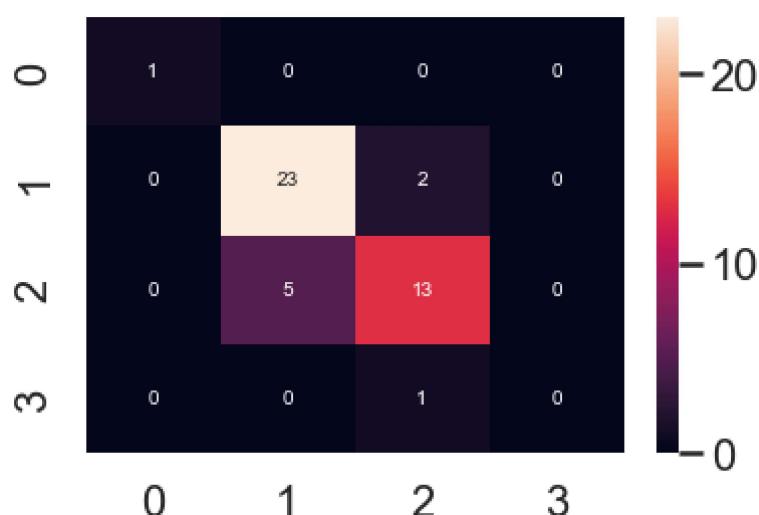
C:\Users\ADMIN\anaconda3\lib\site-packages\sklearn\linear_model_logistic.py:444: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
<https://scikit-learn.org/stable/modules/preprocessing.html>
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(
C:\Users\ADMIN\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1327:
UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
C:\Users\ADMIN\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1327:
UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
C:\Users\ADMIN\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1327:
UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
Accuracy  0.8222222222222222
```

	precision	recall	f1-score	support
1	1.00	1.00	1.00	1
2	0.82	0.92	0.87	25
3	0.81	0.72	0.76	18
4	0.00	0.00	0.00	1
accuracy			0.82	45
macro avg	0.66	0.66	0.66	45
weighted avg	0.80	0.82	0.81	45

Out[13]: <AxesSubplot:>



In [14]:

```
np.bincount(y_train)

Out[14]: array([ 0,  1, 56, 43,  3], dtype=int64)
```

```
In [15]: !pip install imblearn
```

```
Requirement already satisfied: imblearn in c:\users\admin\anaconda3\lib\site-packages (0.0)
Requirement already satisfied: imbalanced-learn in c:\users\admin\anaconda3\lib\site-packages (from imblearn) (0.9.1)
Requirement already satisfied: numpy>=1.17.3 in c:\users\admin\anaconda3\lib\site-packages (from imbalanced-learn->imblearn) (1.20.3)
Requirement already satisfied: scikit-learn>=1.1.0 in c:\users\admin\anaconda3\lib\site-packages (from imbalanced-learn->imblearn) (1.1.1)
Requirement already satisfied: scipy>=1.3.2 in c:\users\admin\anaconda3\lib\site-packages (from imbalanced-learn->imblearn) (1.7.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\admin\anaconda3\lib\site-packages (from imbalanced-learn->imblearn) (2.2.0)
Requirement already satisfied: joblib>=1.0.0 in c:\users\admin\anaconda3\lib\site-packages (from imbalanced-learn->imblearn) (1.1.0)
```

```
In [22]: from imblearn.over_sampling import SMOTE
```

```
In [24]: # transform the dataset
oversample = SMOTE(k_neighbors = 1)
X_train_res,y_train_res = oversample.fit_resample(X,y)
```

```
In [25]: lr = LogisticRegression()
lr.fit(X_train_res, y_train_res.ravel())
predictions = lr.predict(x_test)

print('Accuracy ',accuracy_score(y_test,predictions))
print(classification_report(y_test, predictions))
sns.heatmap(confusion_matrix(y_test,predictions),annot=True,fmt=' .2g ')
```

```
Accuracy 0.9333333333333333
          precision    recall  f1-score   support
          1         1.00     1.00     1.00      1
          2         0.92     0.96     0.94     25
          3         0.94     0.89     0.91     18
          4         1.00     1.00     1.00      1

      accuracy                           0.93      45
   macro avg       0.97     0.96     0.96      45
weighted avg       0.93     0.93     0.93      45
```

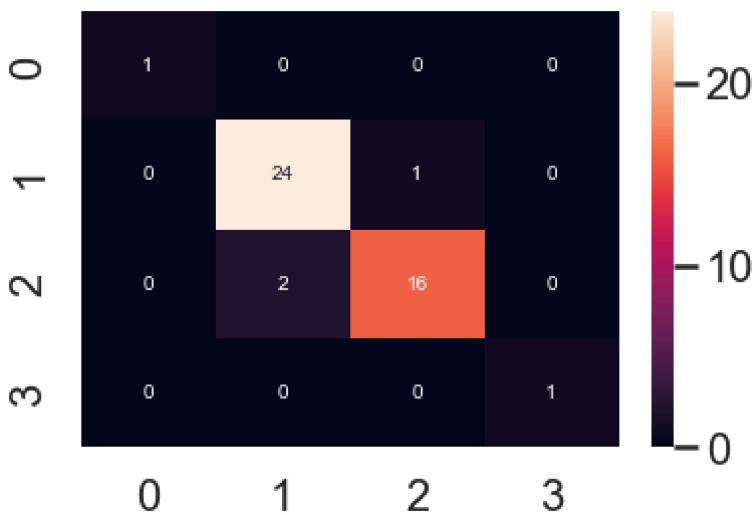
```
C:\Users\ADMIN\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:444: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT.
```

```
Increase the number of iterations (max_iter) or scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
```

```
Please also refer to the documentation for alternative solver options:
```

```
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
n_iter_i = _check_optimize_result(
```

```
<AxesSubplot:>
```



```
In [26]: np.bincount(y_train_res)
```

```
Out[26]: array([ 0, 81, 81, 81, 81], dtype=int64)
```

Tensorflow keras neural networks

```
In [27]: !pip install tensorflow  
!pip install keras
```

```
Requirement already satisfied: tensorflow in c:\users\admin\anaconda3\lib\site-packages (2.9.1)
Requirement already satisfied: setuptools in c:\users\admin\anaconda3\lib\site-packages (from tensorflow) (58.0.4)
Requirement already satisfied: h5py>=2.9.0 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow) (3.2.1)
Requirement already satisfied: wrapt>=1.11.0 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow) (1.12.1)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow) (1.46.3)
Requirement already satisfied: opt-einsum>=2.3.2 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow) (3.3.0)
Requirement already satisfied: google-pasta>=0.1.1 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow) (0.2.0)
Requirement already satisfied: keras<2.10.0,>=2.9.0rc0 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow) (2.9.0)
Requirement already satisfied: libclang>=13.0.0 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow) (14.0.1)
Requirement already satisfied: six>=1.12.0 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow) (1.16.0)
Requirement already satisfied: tensorflow-estimator<2.10.0,>=2.9.0rc0 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow) (2.9.0)
Requirement already satisfied: termcolor>=1.1.0 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow) (1.1.0)
Requirement already satisfied: tensorboard<2.10,>=2.9 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow) (2.9.0)
Requirement already satisfied: astunparse>=1.6.0 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow) (1.6.3)
Requirement already satisfied: protobuf<3.20,>=3.9.2 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow) (3.19.4)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow) (0.26.0)
Requirement already satisfied: gast<=0.4.0,>=0.2.1 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow) (0.4.0)
```

```
Requirement already satisfied: flatbuffers<2,>=1.12 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow) (1.12)
Requirement already satisfied: keras-preprocessing>=1.1.1 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow) (1.1.2)
Requirement already satisfied: absl-py>=1.0.0 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow) (1.1.0)
Requirement already satisfied: numpy>=1.20 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow) (1.20.3)
Requirement already satisfied: packaging in c:\users\admin\anaconda3\lib\site-packages (from tensorflow) (21.0)
Requirement already satisfied: typing-extensions>=3.6.6 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow) (3.10.0.2)
Requirement already satisfied: wheel<1.0,>=0.23.0 in c:\users\admin\anaconda3\lib\site-packages (from astunparse>=1.6.0->tensorflow) (0.37.0)
Requirement already satisfied: markdown>=2.6.8 in c:\users\admin\anaconda3\lib\site-packages (from tensorboard<2.10,>=2.9->tensorflow) (3.3.7)
Requirement already satisfied: werkzeug>=1.0.1 in c:\users\admin\anaconda3\lib\site-packages (from tensorboard<2.10,>=2.9->tensorflow) (2.0.2)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in c:\users\admin\anaconda3\lib\site-packages (from tensorboard<2.10,>=2.9->tensorflow) (0.4.6)
Requirement already satisfied: google-auth<3,>=1.6.3 in c:\users\admin\anaconda3\lib\site-packages (from tensorboard<2.10,>=2.9->tensorflow) (2.6.6)
Requirement already satisfied: requests<3,>=2.21.0 in c:\users\admin\anaconda3\lib\site-packages (from tensorboard<2.10,>=2.9->tensorflow) (2.26.0)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in c:\users\admin\anaconda3\lib\site-packages (from tensorboard<2.10,>=2.9->tensorflow) (1.8.1)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in c:\users\admin\anaconda3\lib\site-packages (from tensorboard<2.10,>=2.9->tensorflow) (0.6.1)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in c:\users\admin\anaconda3\lib\site-packages (from google-auth<3,>=1.6.3->tensorboard<2.10,>=2.9->tensorflow) (5.2.0)
Requirement already satisfied: rsa<5,>=3.1.4 in c:\users\admin\anaconda3\lib\site-packages (from google-auth<3,>=1.6.3->tensorboard<2.10,>=2.9->tensorflow) (4.8)
Requirement already satisfied: pyasn1-modules>=0.2.1 in c:\users\admin\anaconda3\lib\site-packages (from google-auth<3,>=1.6.3->tensorboard<2.10,>=2.9->tensorflow) (0.2.8)
Requirement already satisfied: requests-oauthlib>=0.7.0 in c:\users\admin\anaconda3\lib\site-packages (from google-auth-oauthlib<0.5,>=0.4.1->tensorboard<2.10,>=2.9->tensorflow) (1.3.1)
Requirement already satisfied: importlib-metadata>=4.4 in c:\users\admin\anaconda3\lib\site-packages (from markdown>=2.6.8->tensorboard<2.10,>=2.9->tensorflow) (4.8.1)
Requirement already satisfied: zipp>=0.5 in c:\users\admin\anaconda3\lib\site-packages (from importlib-metadata>=4.4->markdown>=2.6.8->tensorboard<2.10,>=2.9->tensorflow) (3.6.0)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in c:\users\admin\anaconda3\lib\site-packages (from pyasn1-modules>=0.2.1->google-auth<3,>=1.6.3->tensorboard<2.10,>=2.9->tensorflow) (0.4.8)
Requirement already satisfied: idna<4,>=2.5 in c:\users\admin\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorboard<2.10,>=2.9->tensorflow) (3.2)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\admin\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorboard<2.10,>=2.9->tensorflow) (2.0.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\admin\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorboard<2.10,>=2.9->tensorflow) (1.26.7)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\admin\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorboard<2.10,>=2.9->tensorflow) (2021.10.8)
Requirement already satisfied: oauthlib>=3.0.0 in c:\users\admin\anaconda3\lib\site-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1->tensorboard<2.10,>=2.9->tensorflow) (3.2.0)
Requirement already satisfied: pyparsing>=2.0.2 in c:\users\admin\anaconda3\lib\site-packages (from packaging->tensorflow) (3.0.4)
```

```
Requirement already satisfied: keras in c:\users\admin\anaconda3\lib\site-packages  
(2.9.0)
```

```
In [28]:  
from tensorflow.keras.layers import Dense #NN  
from tensorflow.keras.models import Sequential  
from tensorflow.keras.utils import to_categorical
```

```
In [29]:  
features = df.drop(columns=['class'])  
target = df['class']  
from sklearn.model_selection import train_test_split  
  
X_train, X_test, y_train, y_test = train_test_split(features, target)  
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train)
```

```
In [30]:  
model = Sequential()  
  
model.add(Dense(100, input_shape=(features.shape[1],)))  
model.add(Dense(30, activation="relu"))  
# model.add(Dense(45, activation="relu"))  
# model.add(Dense(32, activation="relu"))  
# model.add(Dense(23, activation="relu"))  
model.add(Dense(3, activation="softmax"))
```

```
In [31]:  
model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
<hr/>		
dense (Dense)	(None, 100)	1900
dense_1 (Dense)	(None, 30)	3030
dense_2 (Dense)	(None, 3)	93
<hr/>		
Total params: 5,023		
Trainable params: 5,023		
Non-trainable params: 0		

```
In [32]:  
import tensorflow  
  
model.compile(optimizer="sgd",  
              loss= tensorflow.keras.losses.CategoricalCrossentropy(),  
              metrics=['accuracy'])
```

```
In [33]:  
X_train.shape
```

```
Out[33]: (83, 18)
```

```
In [34]:  
y_train.shape
```

```
Out[34]: (83,)
```

```
In [35]: y_pred = model.predict(X_test)

2/2 [=====] - 0s 0s/step
```

```
In [36]: y_pred
```

```
Out[36]: array([[0.01332371, 0.63358355, 0.35309282],
 [0.0113563 , 0.7007211 , 0.28792265],
 [0.01051941, 0.7464888 , 0.24299186],
 [0.00334291, 0.6833873 , 0.31326988],
 [0.00552439, 0.7415094 , 0.25296625],
 [0.00476152, 0.60363483, 0.3916037 ],
 [0.0116059 , 0.7376119 , 0.25078216],
 [0.01593926, 0.647072 , 0.33698866],
 [0.00875815, 0.74124414, 0.2499977 ],
 [0.00625171, 0.6231458 , 0.3706024 ],
 [0.01424773, 0.6963945 , 0.28935775],
 [0.00682 , 0.73687726, 0.25630265],
 [0.01460958, 0.7293699 , 0.25602052],
 [0.03671781, 0.516455 , 0.44682717],
 [0.00898977, 0.66857946, 0.32243073],
 [0.1227474 , 0.4616481 , 0.4156045 ],
 [0.00650634, 0.68207216, 0.31142148],
 [0.00641715, 0.7753066 , 0.21827629],
 [0.00267669, 0.6922422 , 0.3050812 ],
 [0.00490085, 0.82086897, 0.17423016],
 [0.00485401, 0.780489 , 0.21465696],
 [0.03734061, 0.57403785, 0.38862157],
 [0.03271843, 0.7939493 , 0.17333226],
 [0.00700042, 0.67201483, 0.32098466],
 [0.01022391, 0.70538104, 0.284395 ],
 [0.0226694 , 0.68663234, 0.29069823],
 [0.02572731, 0.76105094, 0.21322171],
 [0.01606635, 0.46163625, 0.52229744],
 [0.00589236, 0.79203176, 0.2020759 ],
 [0.00453373, 0.67180175, 0.32366464],
 [0.00246788, 0.7848282 , 0.21270391],
 [0.02715443, 0.43016136, 0.5426842 ],
 [0.00508959, 0.7792952 , 0.21561518],
 [0.00249271, 0.7866736 , 0.21083367],
 [0.00854404, 0.75358874, 0.23786724],
 [0.01433059, 0.6312669 , 0.35440248],
 [0.00785476, 0.6840458 , 0.3080994 ]], dtype=float32)
```

Tune the Model using GridSearchCv

```
In [37]: from sklearn.pipeline import Pipeline
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV

# Create first pipeline for base without reducing features.

pipe = Pipeline([('classifier', RandomForestClassifier())])
# pipe = Pipeline([('classifier', RandomForestClassifier())])

# Create param grid.

param_grid = [
    {'classifier' : [LogisticRegression()],
     'classifier_penalty' : ['l1', 'l2'],
     'classifier_C' : np.logspace(-4, 4, 20),
```

```

'classifier__solver' : ['liblinear']},
{'classifier' : [RandomForestClassifier()],
 'classifier__n_estimators' : list(range(10,101,10)),
 'classifier__max_features' : list(range(6,32,5))}

]

# Create grid search object

clf = GridSearchCV(pipe, param_grid = param_grid, cv = 5, verbose=True, n_jobs=-1)

# Fit on data

best_clf = clf.fit(X_train, y_train)

```

Fitting 5 folds for each of 100 candidates, totalling 500 fits

C:\Users\ADMIN\anaconda3\lib\site-packages\sklearn\model_selection_split.py:680: UserWarning: The least populated class in y has only 1 members, which is less than n_splits=5.
 warnings.warn(

In [38]:

```

print(clf.best_params_)

print(clf.best_score_)

{'classifier': RandomForestClassifier(max_features=11, n_estimators=80), 'classifier__max_features': 11, 'classifier__n_estimators': 80}
0.7955882352941176

```

Plot the Accuracy and Loss graphs

In [40]:

```

X = df.drop(columns=['class'])
y = df['class']
model.compile(optimizer='adam',
              loss='mse',
              metrics=['accuracy'])

history = model.fit(X,y,steps_per_epoch=3,
                     epochs=3,validation_steps=5)
print(history.history.keys())
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
# summarize history for loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()

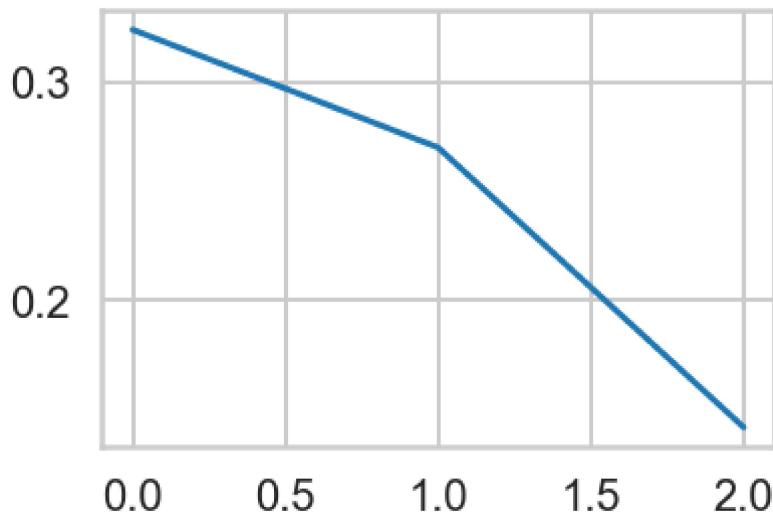
```

Epoch 1/3
3/3 [=====] - 2s 8ms/step - loss: 4.8834 - accuracy: 0.3243
Epoch 2/3
3/3 [=====] - 0s 4ms/step - loss: 4.8642 - accuracy: 0.2703
Epoch 3/3

```
3/3 [=====] - 0s 5ms/step - loss: 4.8430 - accuracy: 0.1419
dict_keys(['loss', 'accuracy'])
```

```
-----  
KeyError Traceback (most recent call last)  
~\AppData\Local\Temp\ipykernel_11880/2332361932.py in <module>  
    10 print(history.history.keys())  
    11 plt.plot(history.history['accuracy'])  
---> 12 plt.plot(history.history['val_accuracy'])  
    13 plt.title('model accuracy')  
    14 plt.ylabel('accuracy')
```

```
KeyError: 'val_accuracy'
```



In []:

In []:

In []: