“Dry Beans”

About Data set
correlation Matrix
Model Building and Enhancing
Support vector classifier
Accuracy of SVM
Decision tree
Accuracy of decision tree
Classification Report
ANN
Accuracy of ANN
Summary of ANN model
visualization ANN
Summary of Models

About Data set

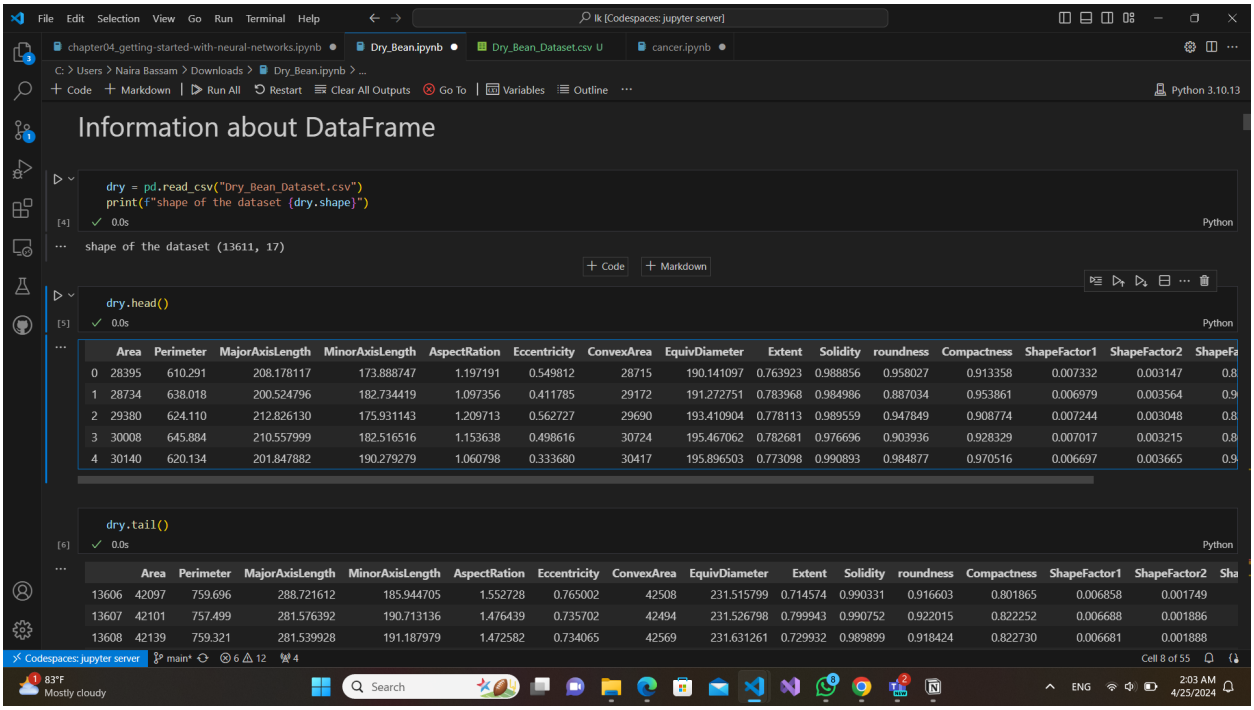
Dry beans is a classification data set about: distinguish seven different registered varieties of dry beans with similar features in order to obtain uniform seed classification.

Total of 16 features; 12 dimensions and 4 shape forms, were obtained from the grains.

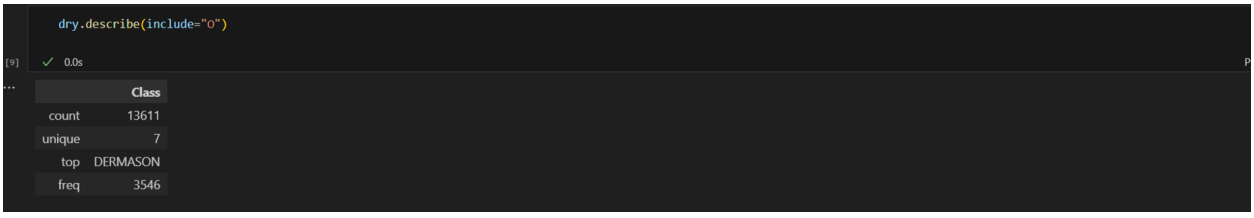
Cleaning and Preparing the data

- Feature Selection
- Dealing with missing data
- Dealing with outliers
- Feature Encoding
- **After drop duplicated:** The shape after drop duplicates is: (13543, 17)

`Information about data set`



`Describe data`



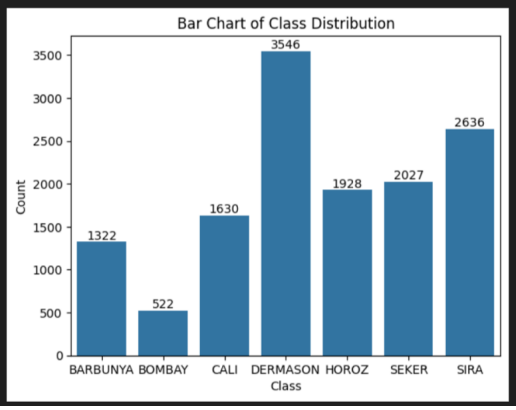
	Class
count	13611
unique	7
top	DERMASON
freq	3546

`Graph classes`

```
ax = sns.barplot(x='Class', y='Count', data=dry.groupby('Class').size().reset_index(name='Count'))
ax.bar_label(ax.containers[0])
plt.xlabel('Class')
plt.ylabel('Count')
plt.title('Bar Chart of Class Distribution')
```

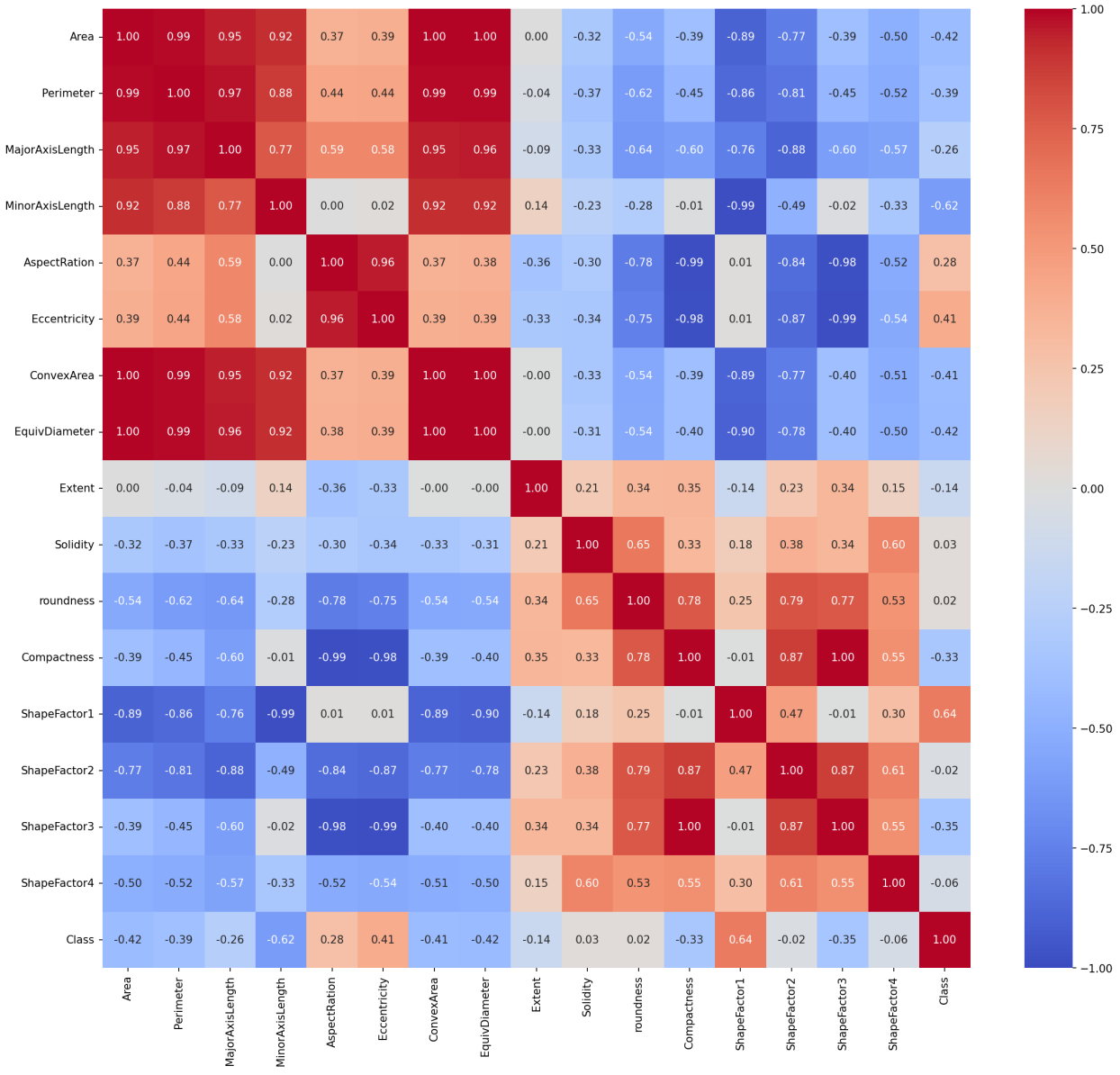
0.4s Python

```
Text(0.5, 1.0, 'Bar Chart of Class Distribution')
```



correlation Matrix

```
dry.select_dtypes(include=['number']).columns
# correlation matrix
corr = dry.corr(numeric_only=True)

# figure settings
plt.figure(figsize=(18,16), dpi=150)

# corr matrix "heatmap"
# sns.heatmap(corr, cmap="Blues")
sns.heatmap(corr, fmt='.2f', annot=True, vmin=-1,center=0, vmax=
```



Model Building and Enhancing

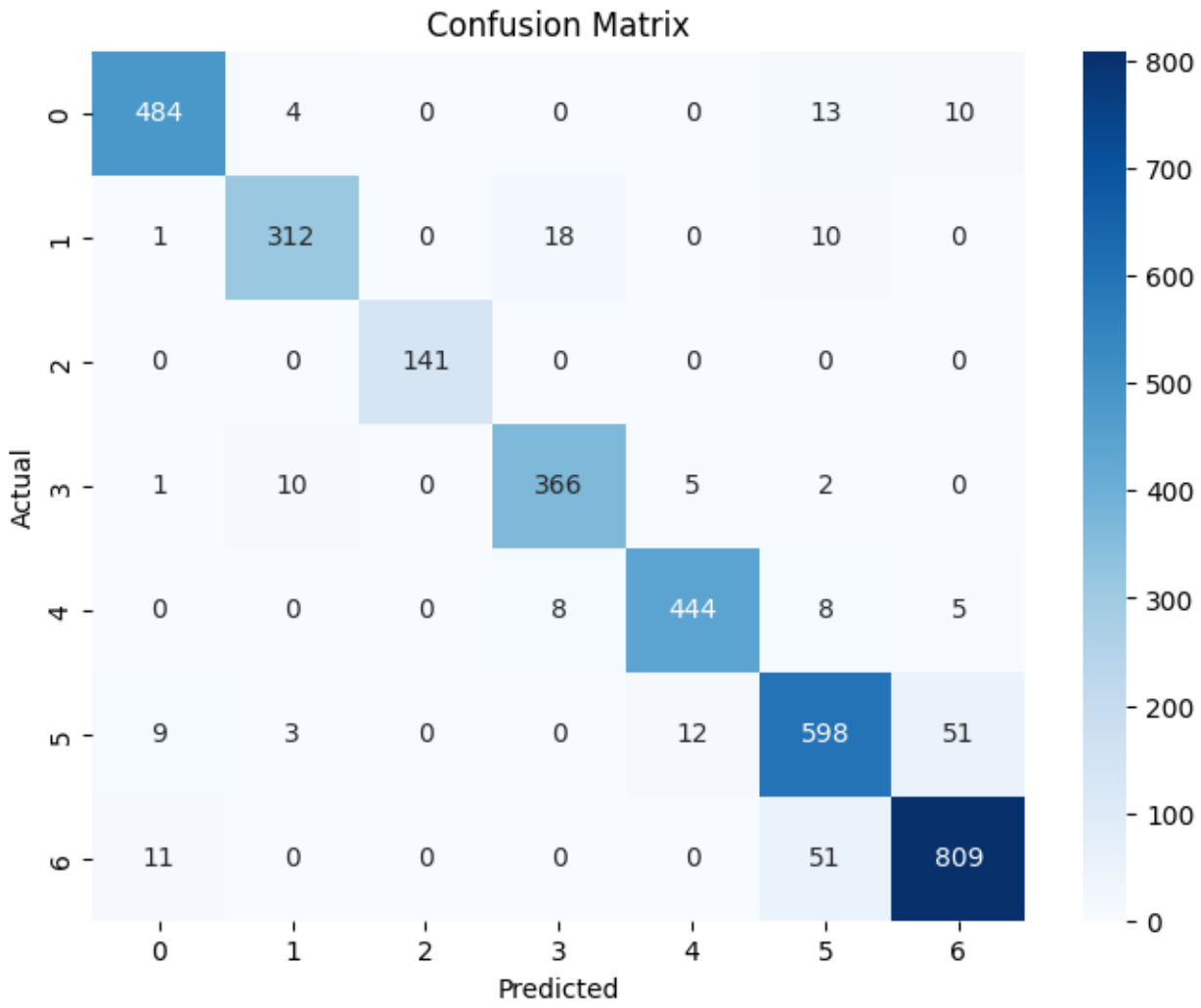
1-Standard scaler

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_s
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler(copy=True,with_std=True)
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Support vector classifier

```
from sklearn.svm import SVC
from sklearn.model_selection import cross_val_score, KFold

svc = SVC(kernel="rbf", C=1)
svc.fit(X_train, y_train)

cv = KFold(n_splits=5, shuffle=True, random_state=42)
scores_svc = cross_val_score(svc, X_train, y_train, cv=cv)
print("Cross-validated scores:", scores_svc)
print(f"Mean Cross-validated-accuracy: {scores_svc.mean()}")
```

confusion matrix for SVM



Accuracy of SVM

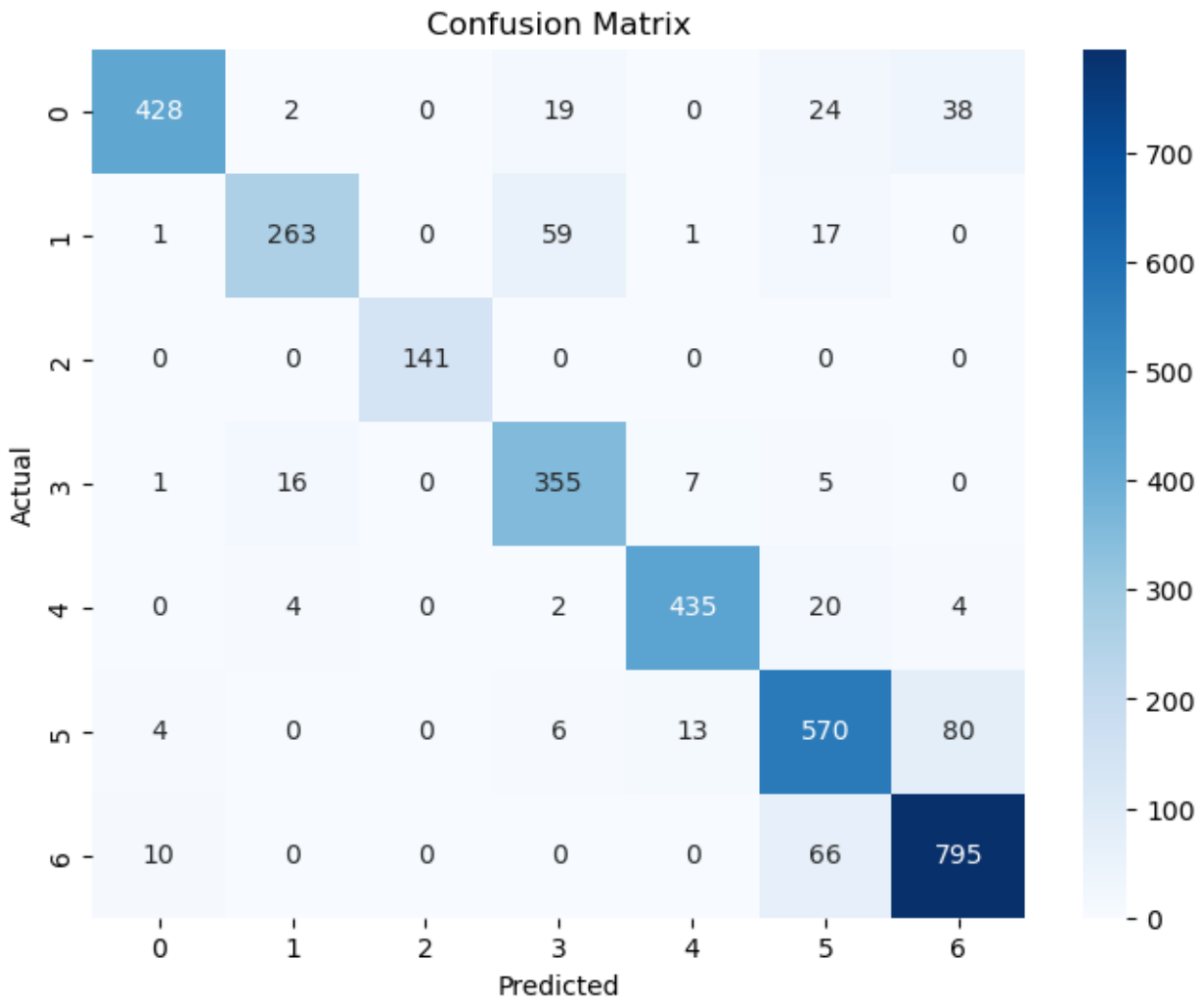
train score: 0.9318696465491779
test score: 0.9314825753101004

Decision tree

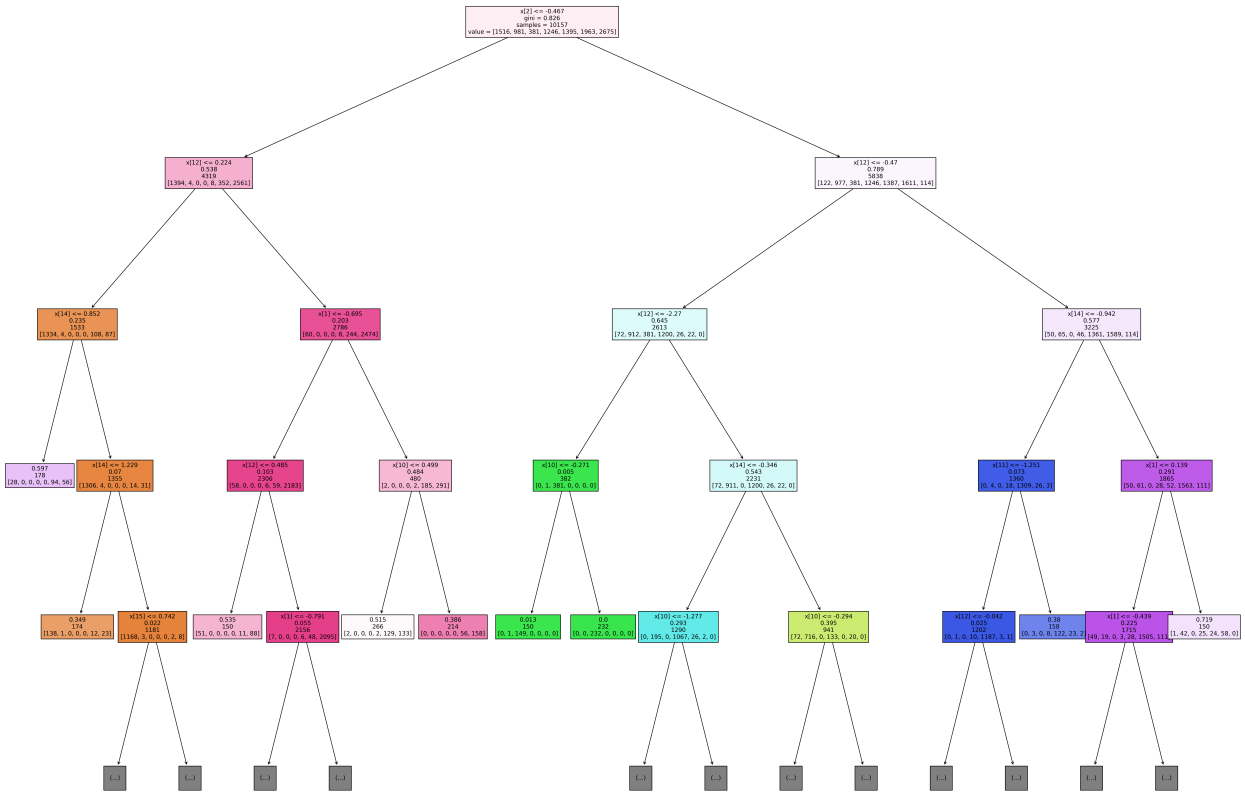
```
from sklearn.tree import DecisionTreeClassifier
# Create a decision tree classifier with customized hyperparamet
tree_clf = DecisionTreeClassifier(
    max_depth=550,  # Maximum depth of the tree
    min_samples_split=350,  # Minimum number of samples required
```

```
    min_samples_leaf=150,  # Minimum number of samples required
    random_state=42  # Random state for reproducibility
)
# Train the classifier on the training data
tree_clf.fit(X_train, y_train)

cv = KFold(n_splits=5, shuffle=True, random_state=42)
scores_tree = cross_val_score(tree_clf, X_train, y_train, cv=cv)
print("Cross-validated scores:", scores_tree)
print(f"Mean Cross-validated-accuracy: {scores_tree.mean()}")
```

Confusion matrix for Decision tree



visualization of decision tree



Accuracy of decision tree

train score: 0.8885497686324703
test score: 0.8821618428824571

Classification Report

	precision	recall	f1-score	support
0	0.96	0.84	0.90	511
1	0.92	0.77	0.84	341
2	1.00	1.00	1.00	141

```
           3       0.80      0.92      0.86       384
           4       0.95      0.94      0.94       465
           5       0.81      0.85      0.83       673
           6       0.87      0.91      0.89       871

accuracy                             0.88      3386
macro avg      0.90      0.89      0.89      3386
weighted avg   0.89      0.88      0.88      3386
```

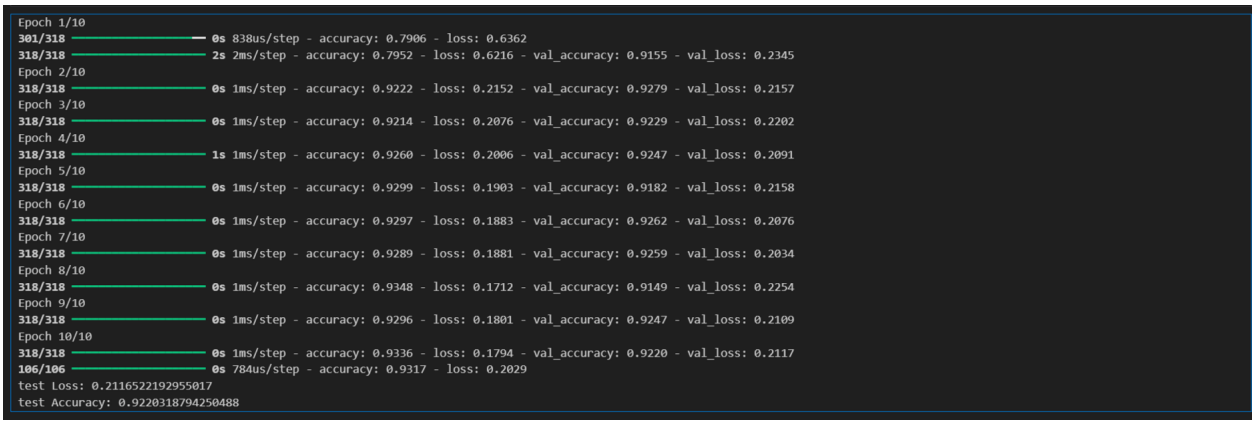
ANN

```
import tensorflow as tf
from tensorflow import keras
model = keras.Sequential([
    keras.layers.Dense(128, activation='relu', input_shape=(X_tr
    keras.layers.Dense(110, activation='relu'),
    keras.layers.Dense(110, activation='relu'),
    keras.layers.Dense(7, activation='softmax')#  # Number of un
])
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',  # Use 'sp
              metrics=['accuracy'])
```

Accuracy of ANN

```
history = model.fit(X_train, y_train, epochs=10, batch_size=32,
test_loss, test_accuracy = model.evaluate(X_test, y_test)

# Evaluate the model
print('test Loss:',test_loss)
print('test Accuracy:',test_accuracy)
```

test Loss: 0.2116522192955017
test Accuracy: 0.9220318794250488



Summary of ANN model

```
model.summary()
```

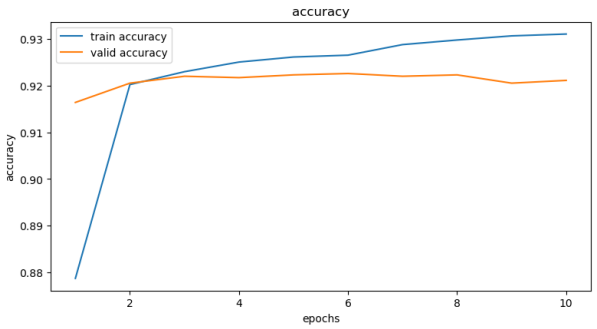
[70] ✓ 0.0s

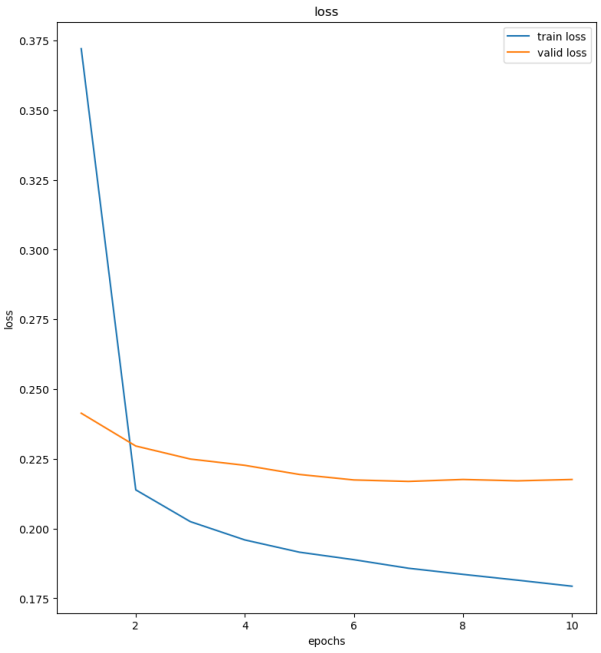
```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 128)	2,176
dense_1 (Dense)	(None, 110)	14,190
dense_2 (Dense)	(None, 110)	12,210
dense_3 (Dense)	(None, 7)	777

```
Total params: 88,061 (343.99 KB)

Trainable params: 29,353 (114.66 KB)

Non-trainable params: 0 (0.00 B)

Optimizer params: 58,708 (229.33 KB)
```

visualization ANN





Summary of Models

SVC	Decision Tree	ANN
train score: 0.9318696465491779	train score: 0.8885497686324703	test Loss: 0.2116522192955017
test score: 0.9314825753101004	test score: 0.8821618428824571	test score: 0.9220318794250488