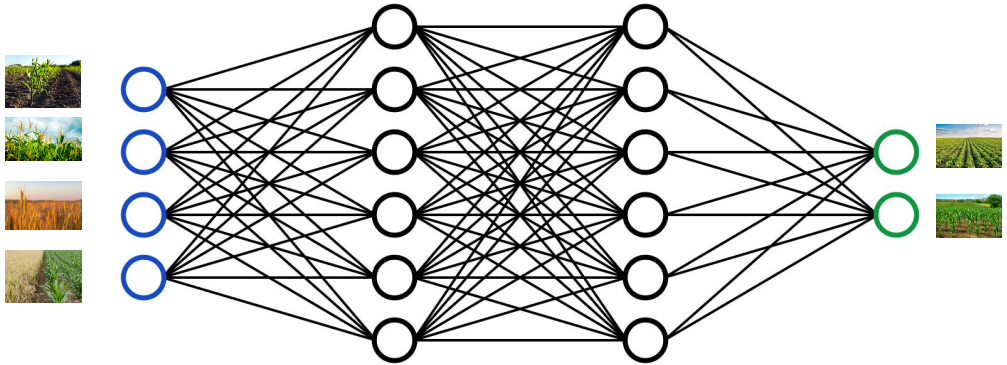
Project DSE316

Crop Recommendation using Machine Learning



By:
Pragati Nayak
Gaurav Kumar Jha
Rachit Bakshi
Mayank Srivastava

Dataset

	N	P	K	temperature	humidity	ph	rainfall	label
0	90	42	43	20.879744	82.002744	6.502985	202.935536	rice
1	85	58	41	21.770462	80.319644	7.038096	226.655537	rice
2	60	55	44	23.004459	82.320763	7.840207	263.964248	rice
3	74	35	40	26.491096	80.158363	6.980401	242.864034	rice
4	78	42	42	20.130175	81.604873	7.628473	262.717340	rice
...	...	...	...	...	...	...	...	...
2195	107	34	32	26.774637	66.413269	6.780064	177.774507	coffee
2196	99	15	27	27.417112	56.636362	6.086922	127.924610	coffee
2197	118	33	30	24.131797	67.225123	6.362608	173.322839	coffee
2198	117	32	34	26.272418	52.127394	6.758793	127.175293	coffee
2199	104	18	30	23.603016	60.396475	6.779833	140.937041	coffee

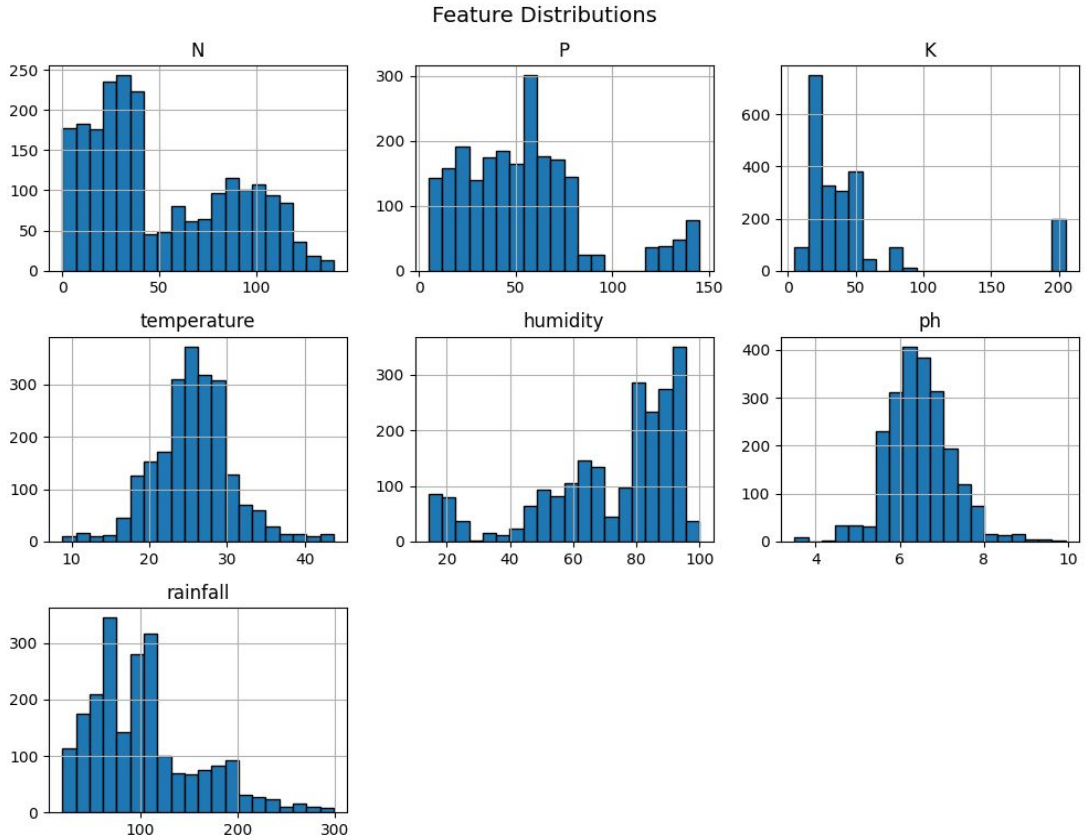
2200 rows × 8 columns

Data Pre-Processing

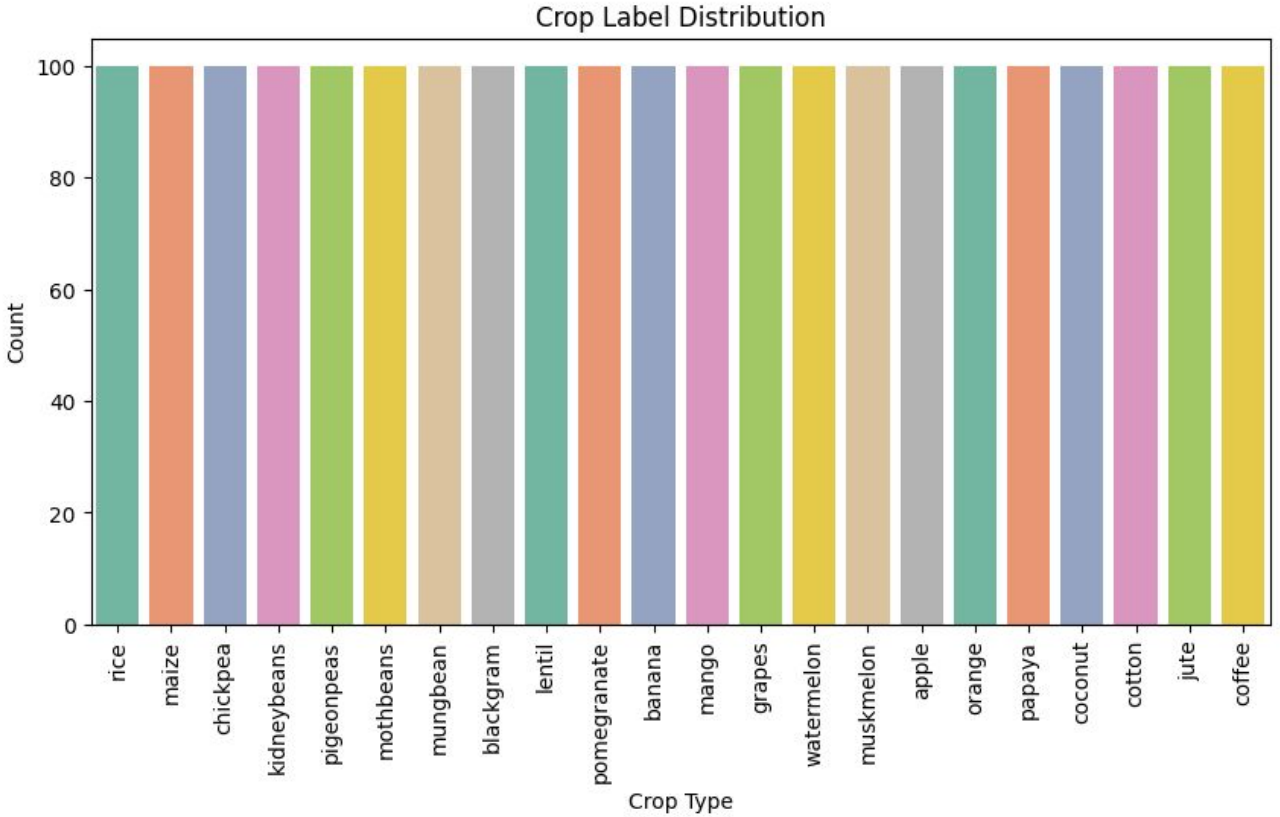
No Missing Values

No Duplicate rows

Number of unique labels: 22



Data Pre-Processing



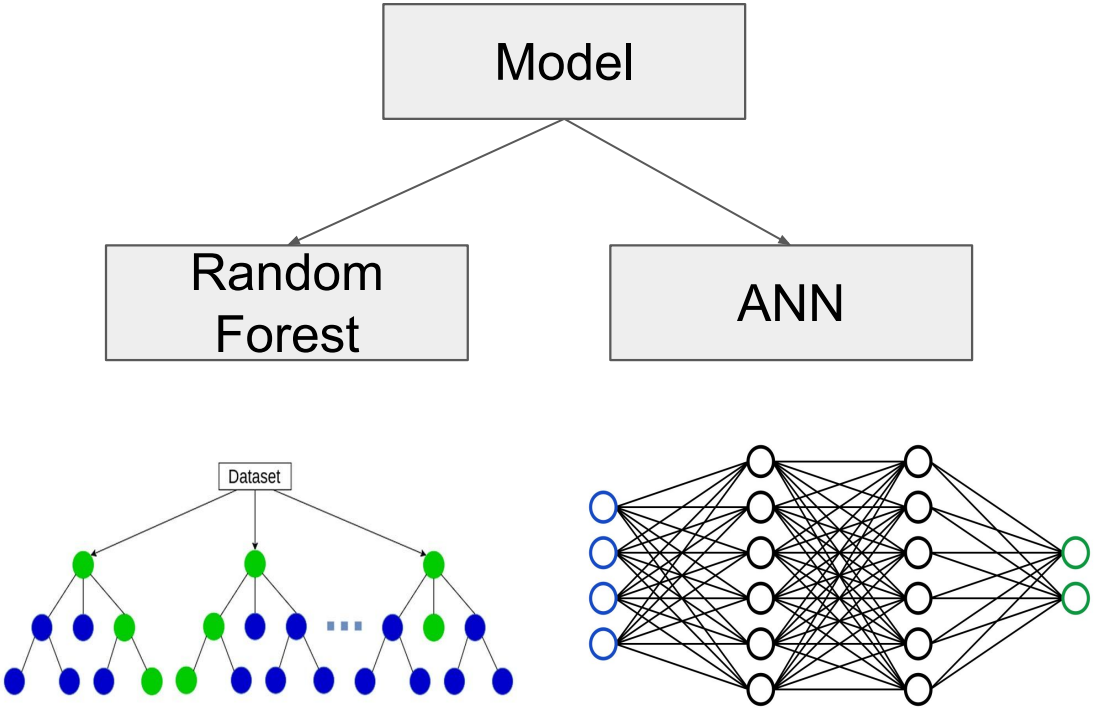
Features

Input Features:
N, P, K,
Temperature,
Humidity, Ph,
Rainfall

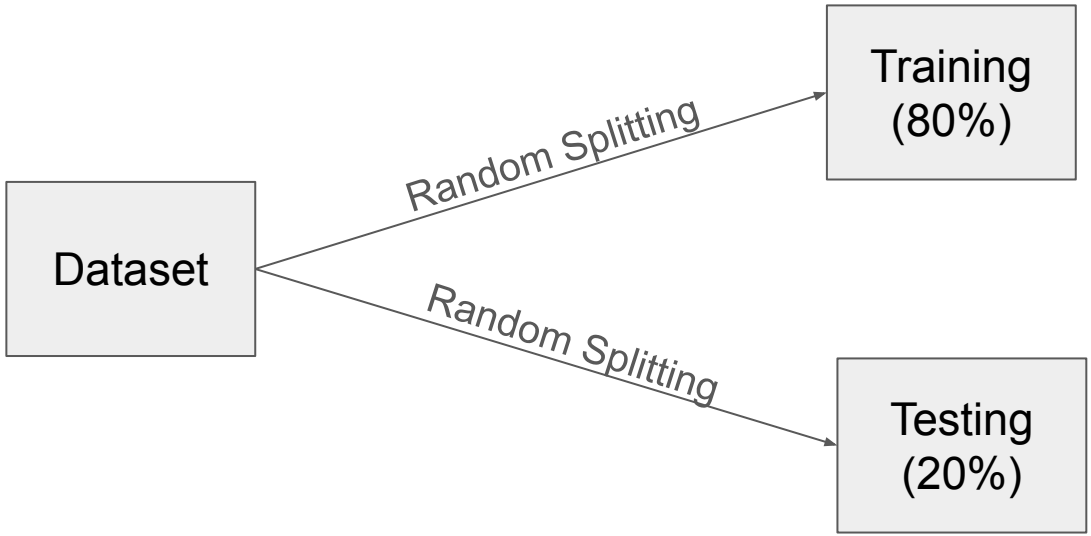
Target Feature:
label

N	P	K	temperature	humidity	ph	rainfall
90	42	43	20.879744	82.002744	6.502985	202.935536
85	58	41	21.770462	80.319644	7.038096	226.655537
60	55	44	23.004459	82.320763	7.840207	263.964248
74	35	40	26.491096	80.158363	6.980401	242.864034
78	42	42	20.130175	81.604873	7.628473	262.717340
...	...	...	...	...	...	...
107	34	32	26.774637	66.413269	6.780064	177.774507
99	15	27	27.417112	56.636362	6.086922	127.924610
118	33	30	24.131797	67.225123	6.362608	173.322839
117	32	34	26.272418	52.127394	6.758793	127.175293
104	18	30	23.603016	60.396475	6.779833	140.937041

```
label
rice          100
maize         100
chickpea      100
kidneybeans   100
pigeonpeas    100
mothbeans     100
mungbean      100
blackgram     100
lentil        100
pomegranate   100
banana        100
mango         100
grapes        100
watermelon    100
muskmelon     100
apple         100
orange        100
papaya        100
coconut       100
cotton        100
jute          100
coffee        100
Name: count, dtype: int64
```


Model
Random Forest
ANN
Dataset

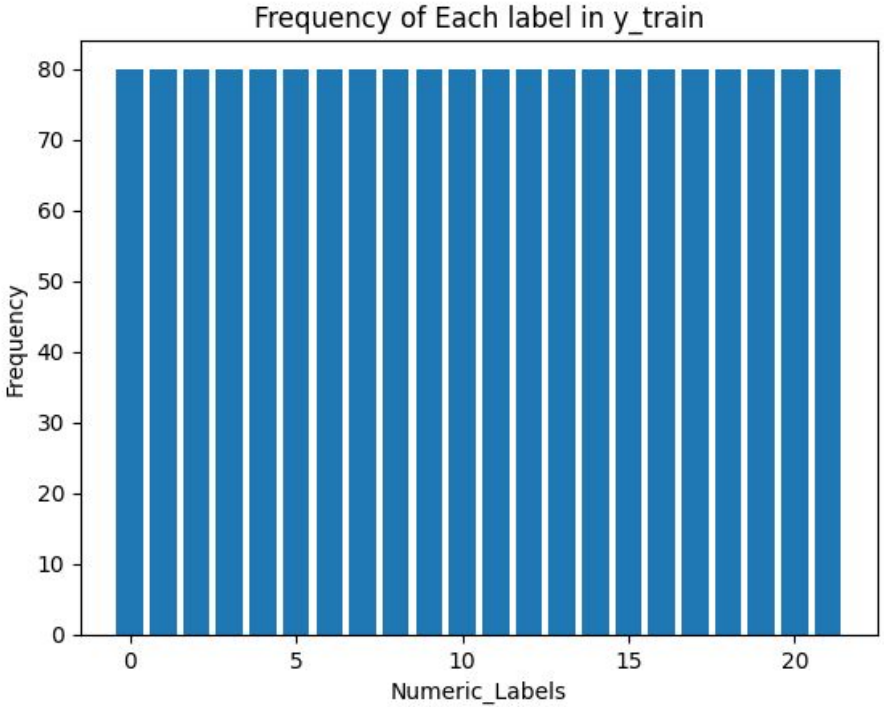
Random Forest

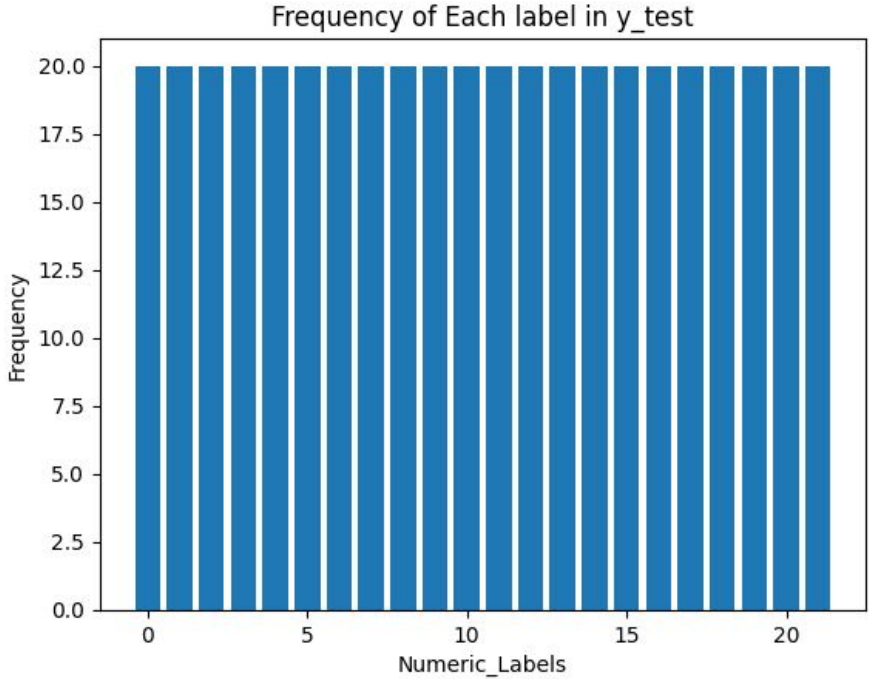


Data	Shape
X_train	(1760, 7)
y_train	(1760,)
X_test	(440, 7)
y_test	(440)

Labels are encoded by Numerical Encoding

Random Forest





Random Forest

N_estimators = 10

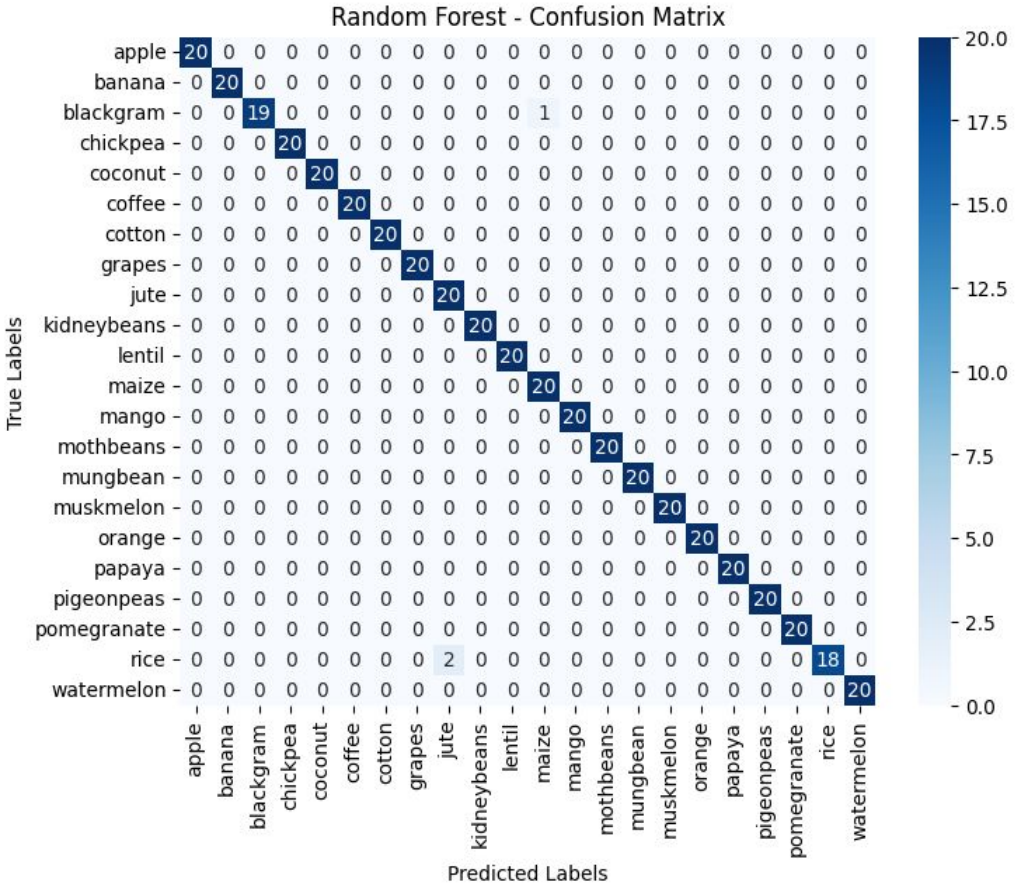
Train the model

Testing

Evaluation Metrics

```
Accuracy: 0.9931818181818182
Precision (macro): 0.9937032664305392
Recall (macro): 0.993181818181818
F1 (macro): 0.9931690047222781
F1 (micro): 0.9931818181818182
```

Random Forest



Tune Random Forest

```
param_grid = {
    'rf__n_estimators': [10,20, 30, 40, 50, 70, 80, 100],
    'rf__max_depth': [None, 20, 30],
    'rf__min_samples_split': [2, 3, 4],
    'rf__min_samples_leaf': [1, 2],
    'rf__class_weight': [None, 'balanced']
}
```

Stratified K Fold : number of splits = 5, Shuffle = True

GridsearchCV

Fitting 5 folds for each of 288 candidates, totalling 1440 fits

Tune Random Forest

```
Best hyperparameters:
{'rf__class_weight': None, 'rf__max_depth': None, 'rf__min_samples_leaf': 1, 'rf__min_samples_split': 4, 'rf__n_estimators': 30}
```

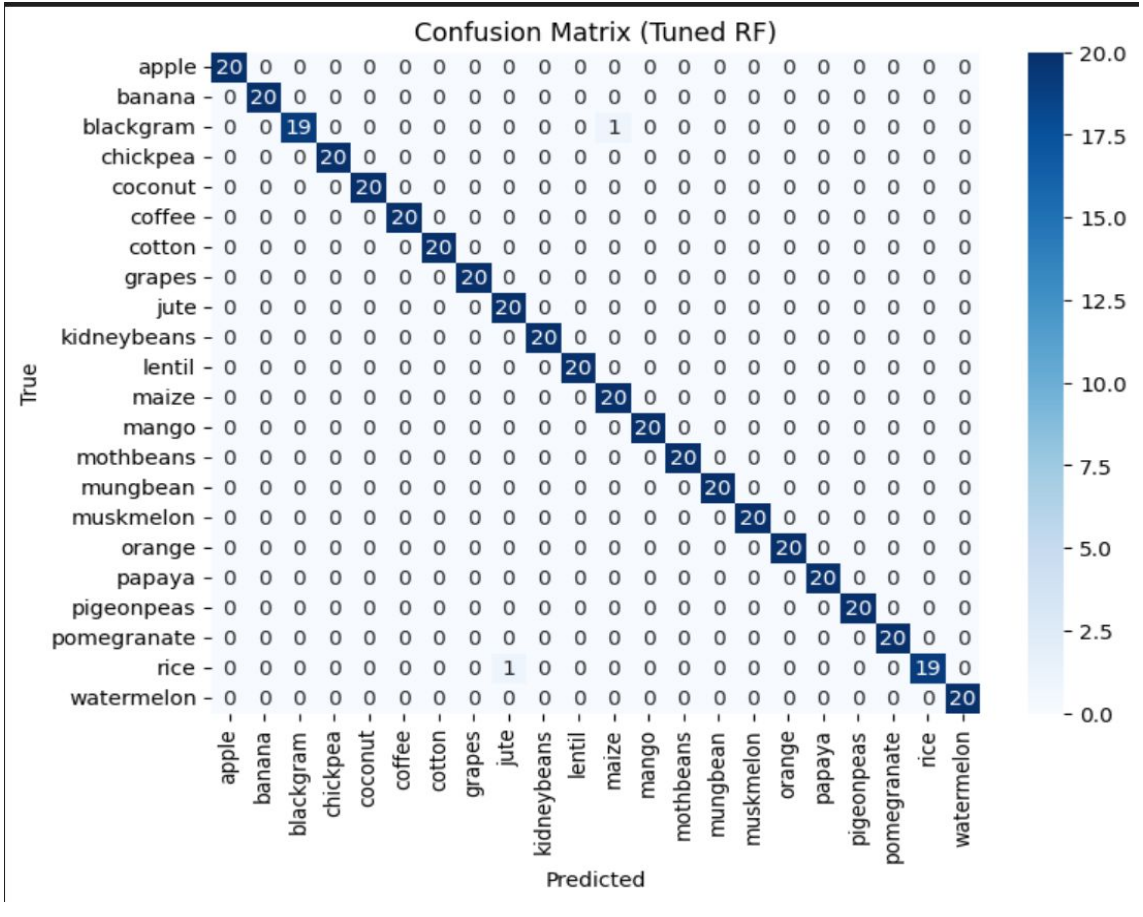
N_estimators = 30

Train the tuned model

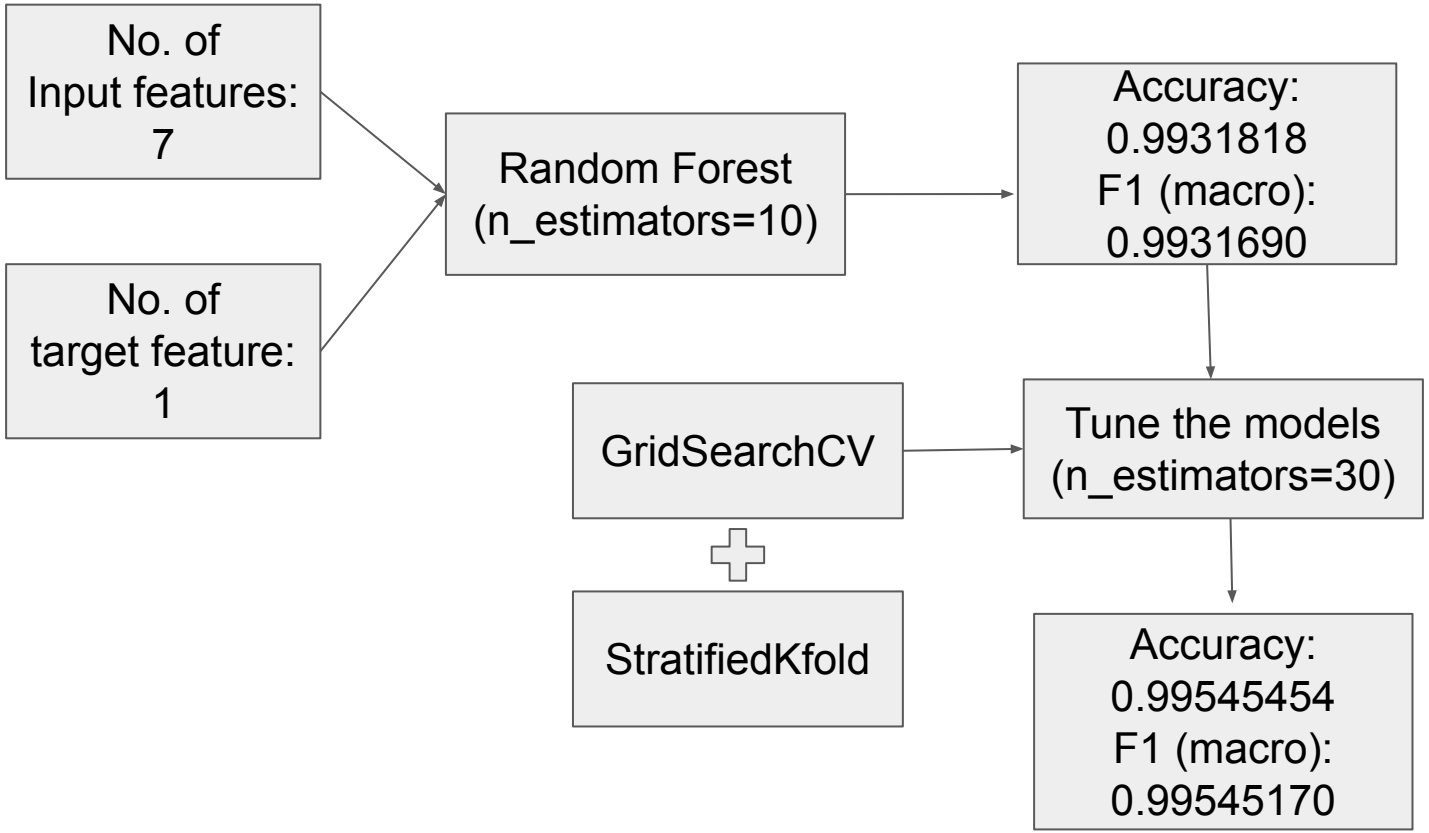
Testing

```
Accuracy: 0.9954545454545455
Precision (macro): 0.9956709956709957
Recall (macro): 0.9954545454545454
F1 (macro): 0.9954517027687758
```

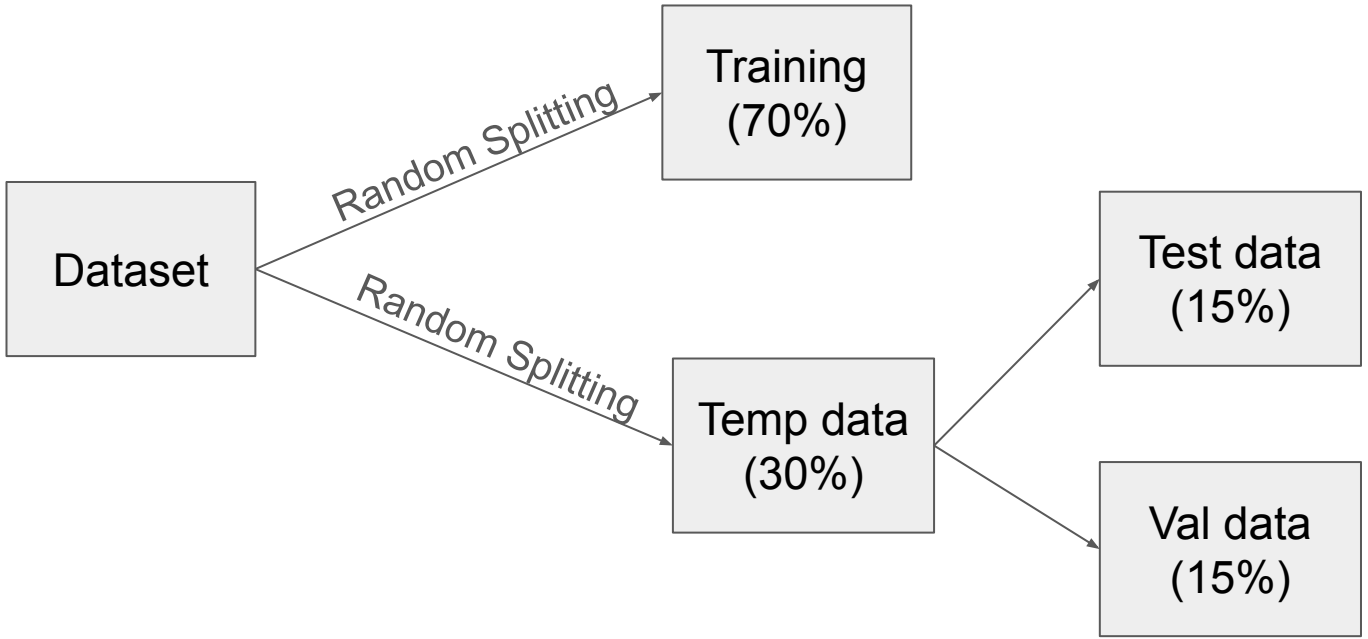
Tune Random Forest



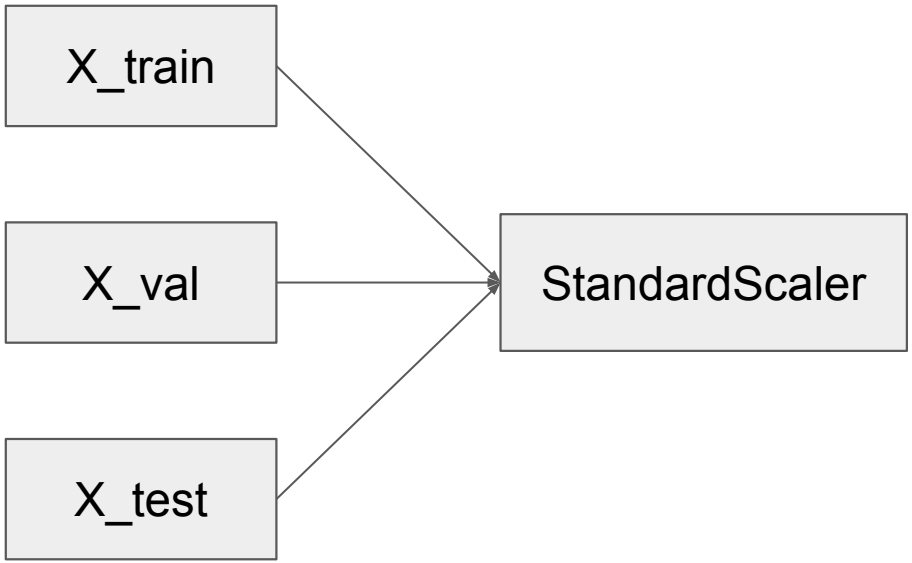
Pipeline of Random Forest



ANNs
Data Pre-processing

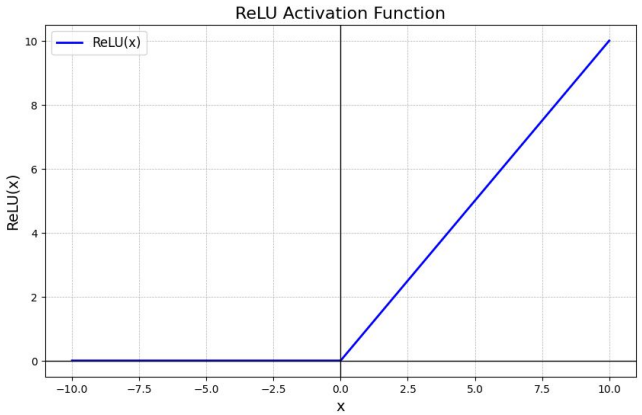


ANNs
Data Pre-processing



ANNs Model

```
model = keras.Sequential([
    layers.Dense(8, activation='relu', input_shape=(X_train.shape[1],)),
    layers.Dropout(0.1),
    layers.Dense(8, activation='relu'),
    layers.Dropout(0.1),
    layers.Dense(8, activation='relu'),
    layers.Dense(len(np.unique(y_enc)), activation='softmax')
])
```



$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

ANNs Model

Optimizer: Adam

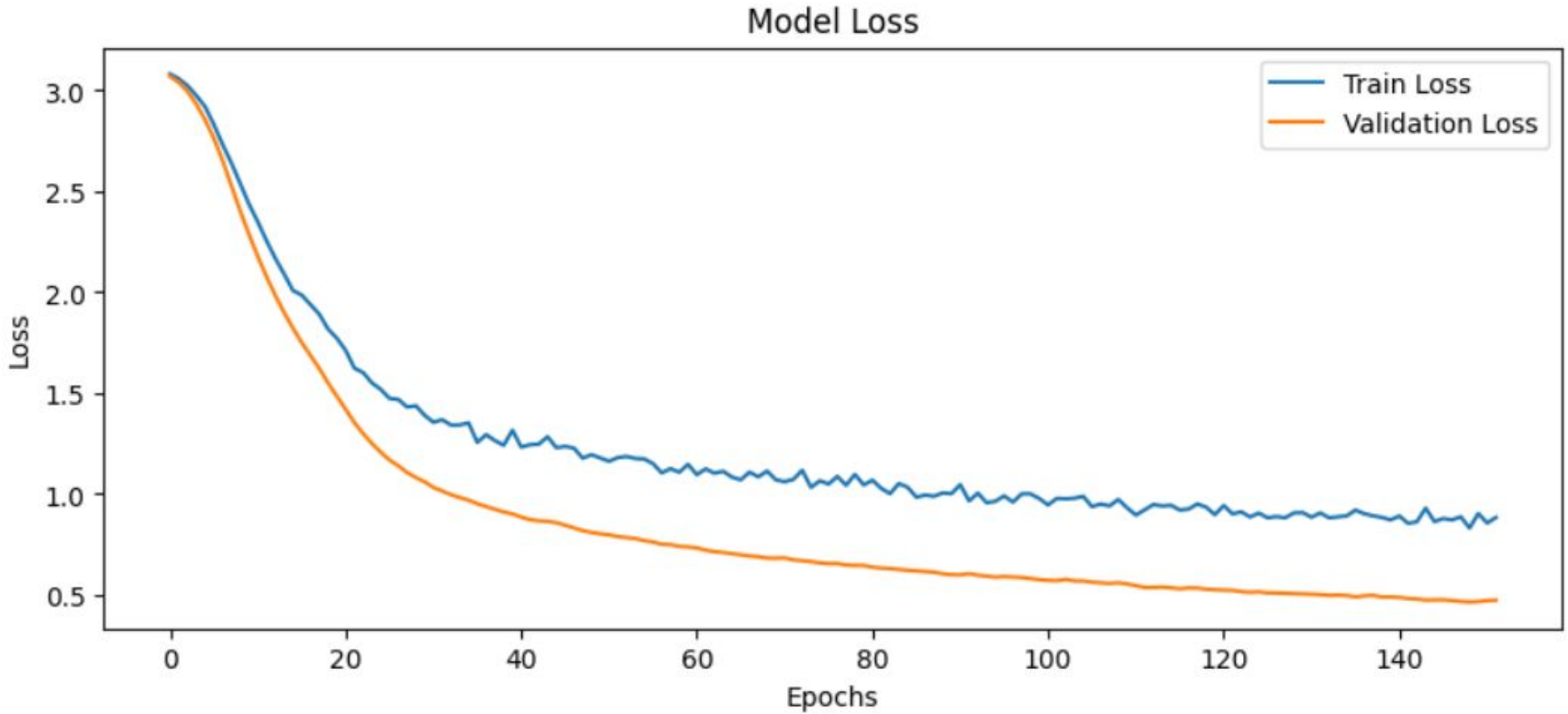
Loss: Sparse categorical crossentropy

Metrics: Accuracy

Early Stopping

```
history = model.fit(
    X_train, y_train,
    epochs=200,
    batch_size=64,
    validation_data=(X_val, y_val),
    callbacks=[early_stop],
    verbose=1
)
```

ANNs Model


Model Loss
Train Loss
Validation Loss
Loss
Epochs
3.0
2.5
2.0
1.5
1.0
0.5
0
20
40
60
80
100
120
140

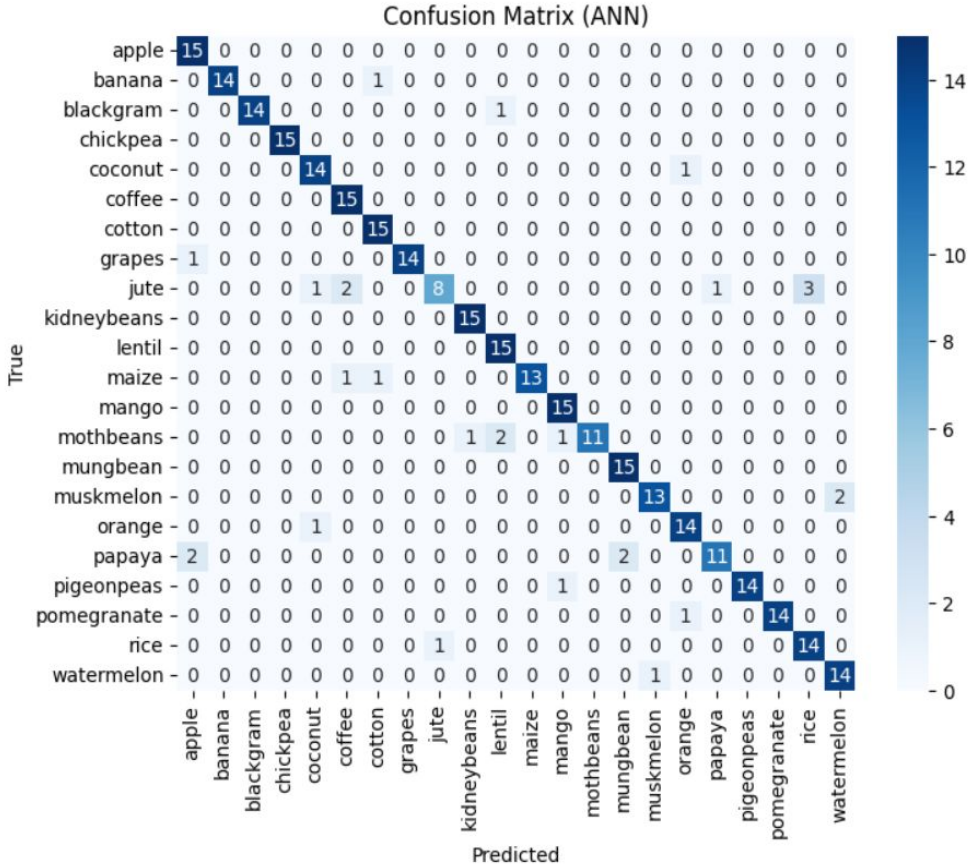
ANNs Model

Accuracy: 0.9151515151515152

Precision (macro): 0.921237055428232

Recall (macro): 0.9151515151515153

F1 Score: 0.9123880349705759



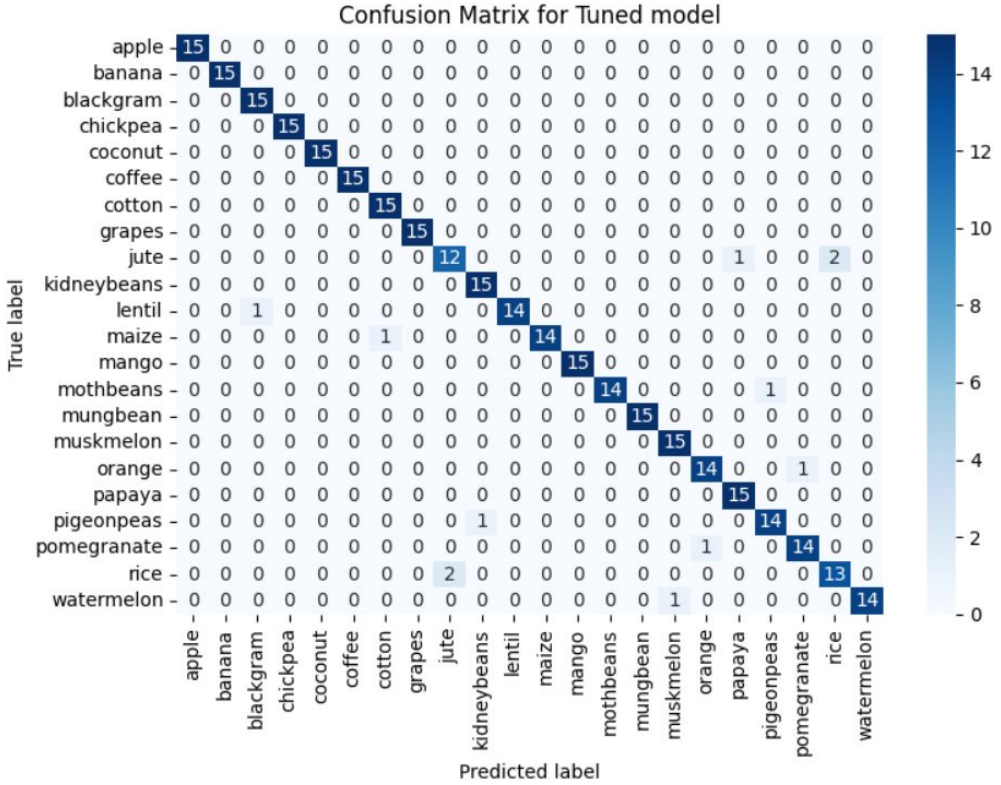
Tuned ANNs Model

```
param_space = {
    'units': [16, 32, 64],
    'dropout': [0.1, 0.2],
    'lr': [1e-3, 1e-4],
    'n_hidden': [ 2, 3, 4],
    'batch_size': [32, 64],
}
```

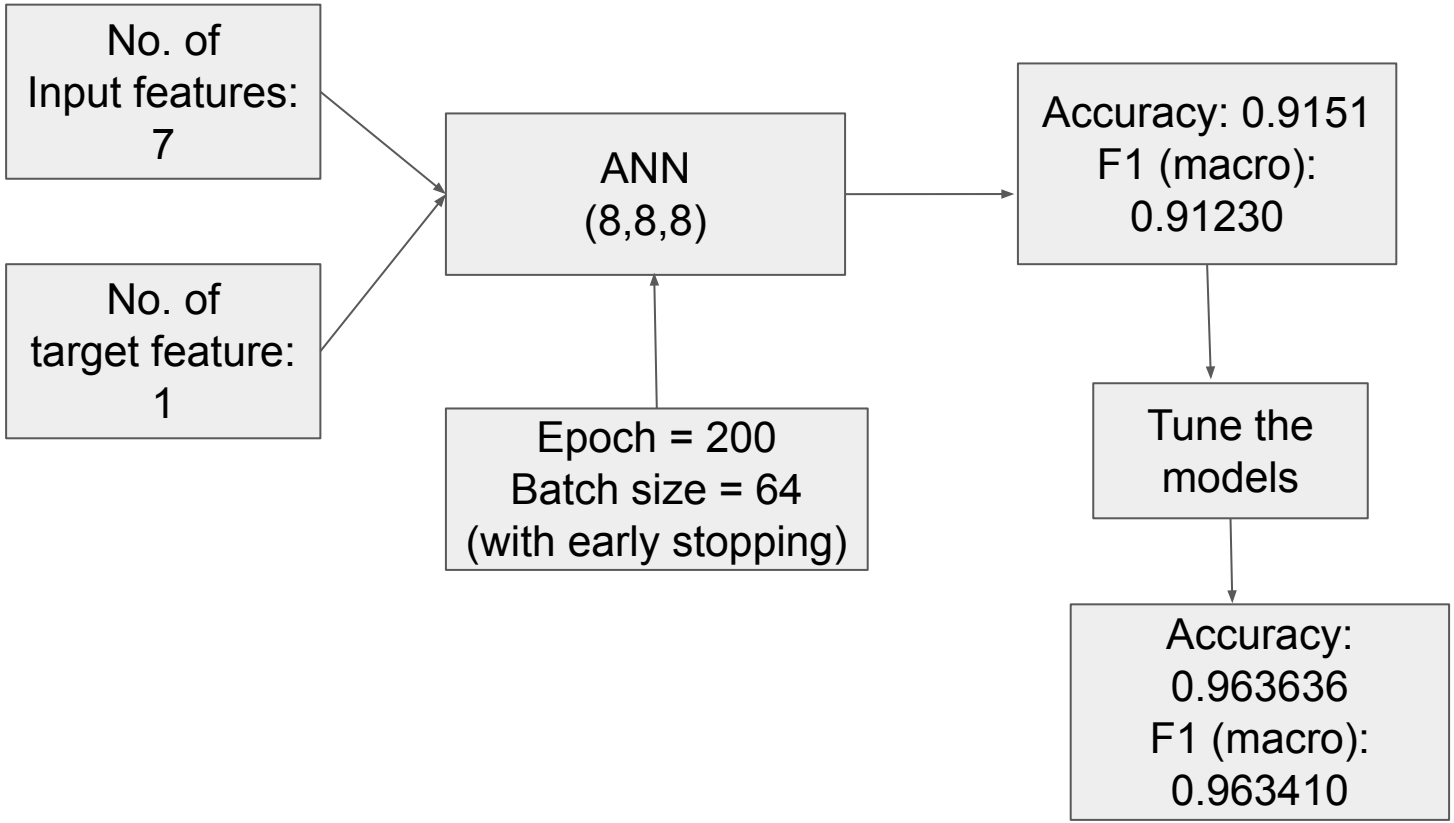
```
n_splits = 3
skf = StratifiedKFold(n_splits=n_splits, shuffle=True, random_state=42)
```

Tuned ANNs Model

```
Test Accuracy: 0.9636363636363636
Precision (macro): 0.964150432900433
Recall (macro): 0.9636363636363636
Test F1 (macro): 0.9634105234772644
```



Pipeline of ANNs



Comparison

Random Forest

Accuracy:
0.995454
F1 (macro):
0.995451

ANNs

Accuracy:
0.963636
F1 (macro):
0.963410

Best model performance: Random Forest


Thank
You