0.) Import and Clean data

In [ ]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

In [ ]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import BaggingClassifier
from sklearn.datasets import make_classification
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.tree import plot_tree
from sklearn.metrics import confusion_matrix
import seaborn as sns
```

In [ ]:
```
#drive.mount('/content/gdrive/', force_remount = True)
```

```
Mounted at /content/gdrive/
```

In [ ]:
```
df = pd.read_csv("bank-additional-full (1).csv",delimiter = ";")
```

In [ ]:
```
df = pd.read_csv('bank-additional-full (1).csv', sep = ';')
df.head()
```

Out[ ]:

	age	job	marital	education	default	housing	loan	contact	month	day_of_v
0	56	housemaid	married	basic.4y	no	no	no	telephone	may	
1	57	services	married	high.school	unknown	no	no	telephone	may	
2	37	services	married	high.school	no	yes	no	telephone	may	
3	40	admin.	married	basic.6y	no	no	no	telephone	may	
4	56	services	married	high.school	no	no	yes	telephone	may	

5 rows × 21 columns

In [ ]:
```
df = df.drop(["default", "pdays",       "previous",     "poutcome",     "emp
df = pd.get_dummies(df, columns = ["loan", "job","marital","housing","contac
```

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

Out[ ]:

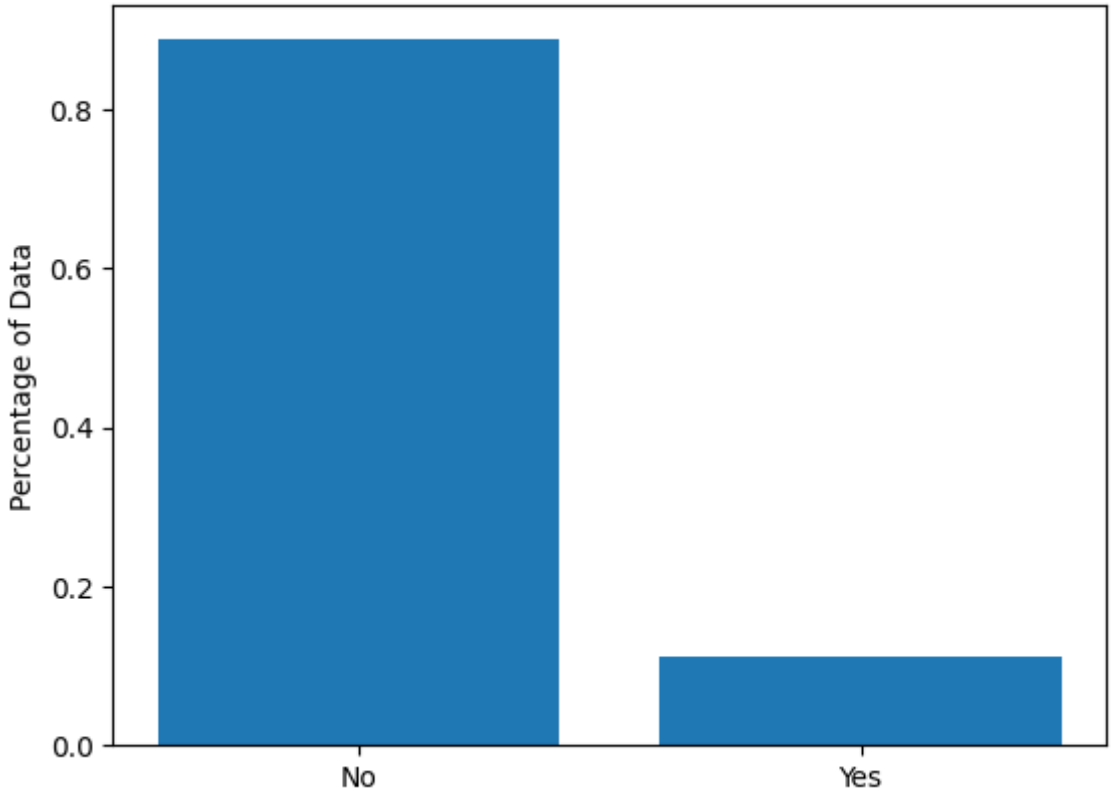
	age	duration	y	loan_unknown	loan_yes	job_blue-collar	job_entrepreneur	job_housemaid
0	56	261	no	False	False	False	False	True
1	57	149	no	False	False	False	False	False
2	37	226	no	False	False	False	False	False
3	40	151	no	False	False	False	False	False
4	56	307	no	False	True	False	False	False

5 rows × 83 columns

In [ ]:
```
y = pd.get_dummies(df["y"], drop_first = True)
X = df.drop(["y"], axis = 1)
```

In [ ]:
```
```

In [ ]:
```
obs = len(y)
plt.bar(["No","Yes"],[len(y[y.yes==0])/obs,len(y[y.yes==1])/obs])
plt.ylabel("Percentage of Data")
plt.show()
```



In [ ]:
```
# Train Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, ra

scaler = StandardScaler().fit(X_train)

X_scaled = scaler.transform(X_train)
X_test = scaler.transform(X_test)
```

1.) Based on the visualization above, use your expert opinion to transform the data based on what we learned this quarter

In [ ]:
```
###############
###TRANSFORM###
###############
from imblearn.over_sampling import RandomOverSampler
from imblearn.under_sampling import RandomUnderSampler
from imblearn.over_sampling import SMOTE

ros = RandomOverSampler()
over_X, over_y = ros.fit_resample(X_train, y_train)
```

```
X_scaled = over_X
y_train = over_y
```

2.) Build and visualize a decision tree of Max Depth 3. Show the confusion matrix.

In [ ]:

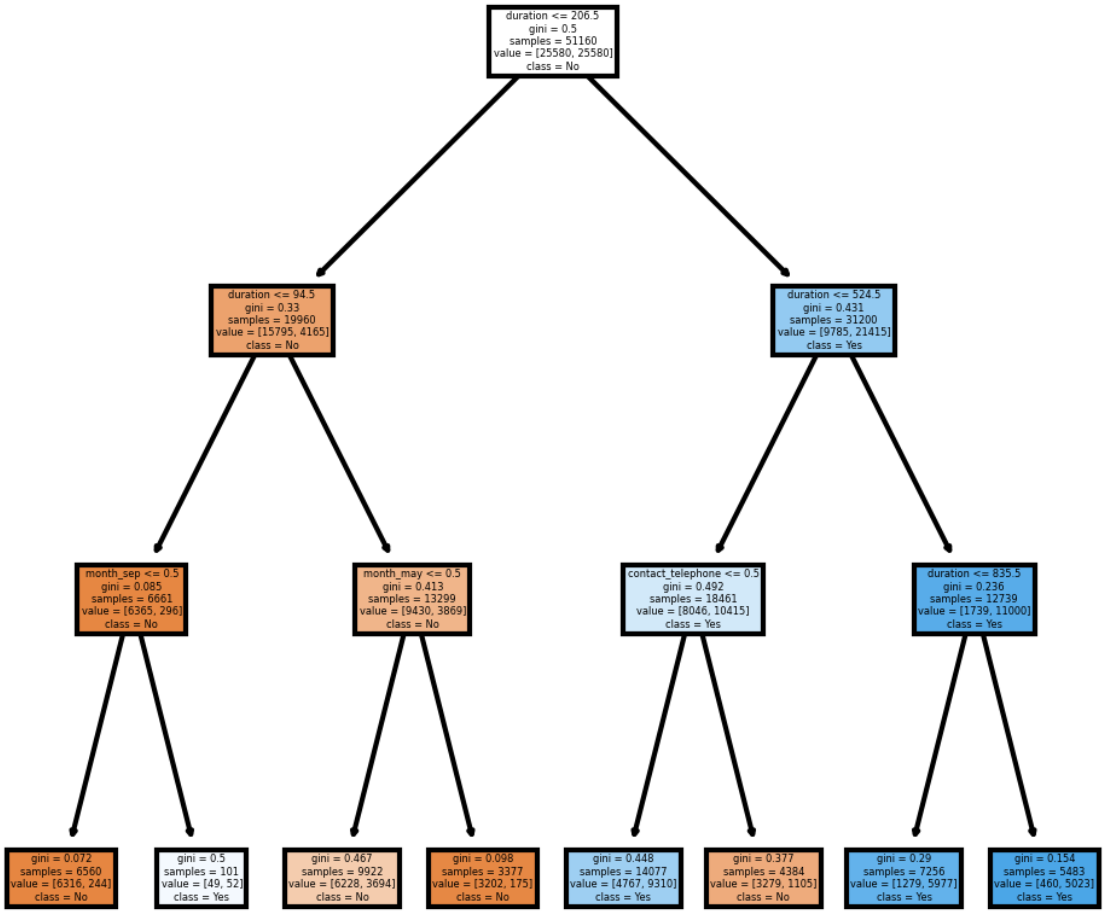
In [ ]:
```
dtree_main = DecisionTreeClassifier(max_depth = 3)
dtree_main.fit(X_scaled, y_train)
```

Out[ ]:
```
DecisionTreeClassifier
DecisionTreeClassifier(max_depth=3)
```

In [ ]:
```
fig, axes = plt.subplots(nrows = 1,ncols = 1,figsize = (4,4), dpi=300)
plot_tree(dtree_main, filled = True, feature_names = X.columns, class_names=


#fig.savefig('imagename.png')
```

Out[ ]:
```
[Text(0.5, 0.875, 'duration <= 206.5\ngini = 0.5\nsamples = 51160\nvalue =
[25580, 25580]\nclass = No'),
 Text(0.25, 0.625, 'duration <= 94.5\ngini = 0.33\nsamples = 19960\nvalue =
[15795, 4165]\nclass = No'),
 Text(0.125, 0.375, 'month_sep <= 0.5\ngini = 0.085\nsamples = 6661\nvalue
= [6365, 296]\nclass = No'),
 Text(0.0625, 0.125, 'gini = 0.072\nsamples = 6560\nvalue = [6316, 244]\ncl
ass = No'),
 Text(0.1875, 0.125, 'gini = 0.5\nsamples = 101\nvalue = [49, 52]\nclass =
Yes'),
 Text(0.375, 0.375, 'month_may <= 0.5\ngini = 0.413\nsamples = 13299\nvalue
= [9430, 3869]\nclass = No'),
 Text(0.3125, 0.125, 'gini = 0.467\nsamples = 9922\nvalue = [6228, 3694]\nc
lass = No'),
 Text(0.4375, 0.125, 'gini = 0.098\nsamples = 3377\nvalue = [3202, 175]\ncl
ass = No'),
 Text(0.75, 0.625, 'duration <= 524.5\ngini = 0.431\nsamples = 31200\nvalue
= [9785, 21415]\nclass = Yes'),
 Text(0.625, 0.375, 'contact_telephone <= 0.5\ngini = 0.492\nsamples = 1846
1\nvalue = [8046, 10415]\nclass = Yes'),
 Text(0.5625, 0.125, 'gini = 0.448\nsamples = 14077\nvalue = [4767, 9310]\n
class = Yes'),
 Text(0.6875, 0.125, 'gini = 0.377\nsamples = 4384\nvalue = [3279, 1105]\nc
lass = No'),
 Text(0.875, 0.375, 'duration <= 835.5\ngini = 0.236\nsamples = 12739\nvalu
e = [1739, 11000]\nclass = Yes'),
 Text(0.8125, 0.125, 'gini = 0.29\nsamples = 7256\nvalue = [1279, 5977]\ncl
ass = Yes'),
 Text(0.9375, 0.125, 'gini = 0.154\nsamples = 5483\nvalue = [460, 5023]\ncl
ass = Yes')]
```



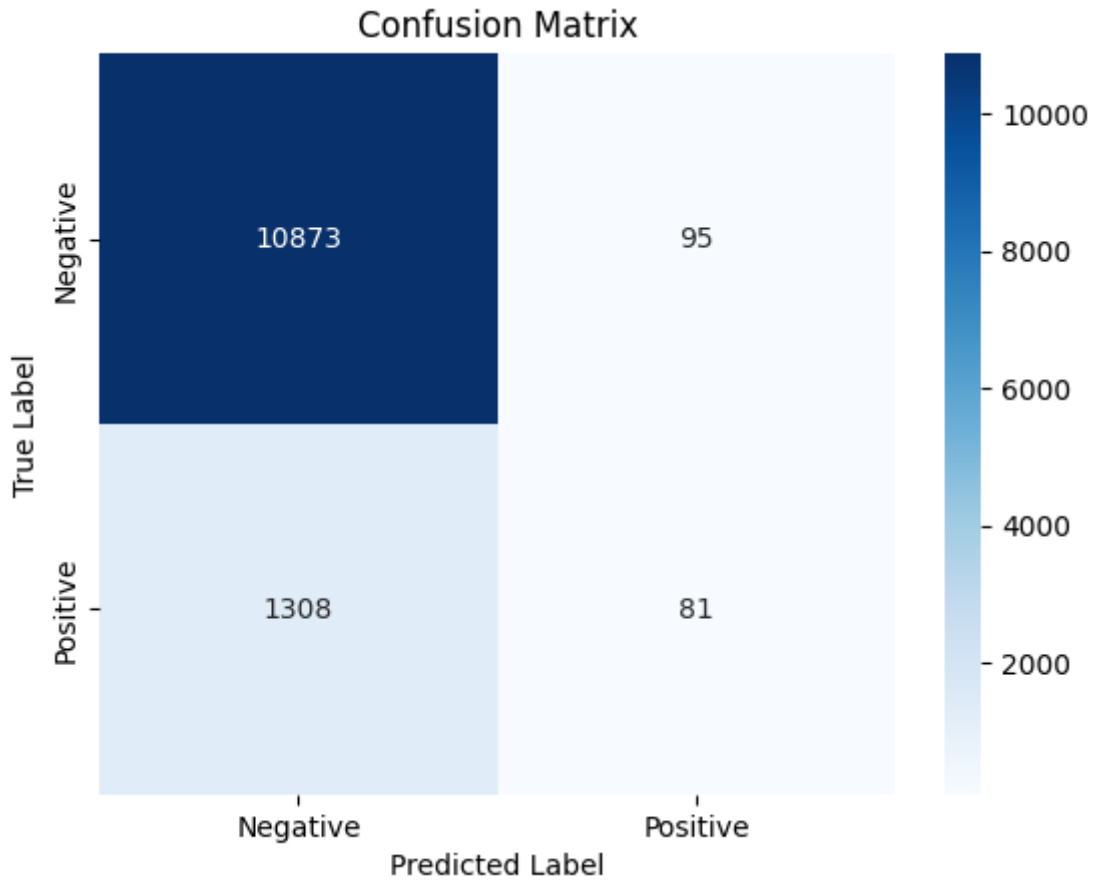
1b.) Confusion matrix on out of sample data. Visualize and store as variable

In [ ]:
```
y_pred = dtree_main.predict(X_test)
y_true = y_test
cm_raw = confusion_matrix(y_true, y_pred)
```

```
/Users/laoga/anaconda3/lib/python3.11/site-packages/sklearn/base.py:439: Us
erWarning: X does not have valid feature names, but DecisionTreeClassifier
was fitted with feature names
  warnings.warn(
```

In [ ]:
```
class_labels = ['Negative', 'Positive']

# Plot the confusion matrix as a heatmap
sns.heatmap(cm_raw, annot=True, fmt='d', cmap='Blues', xticklabels=class_lab
plt.title('Confusion Matrix')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.show()
```



3.) Use bagging on your descision tree

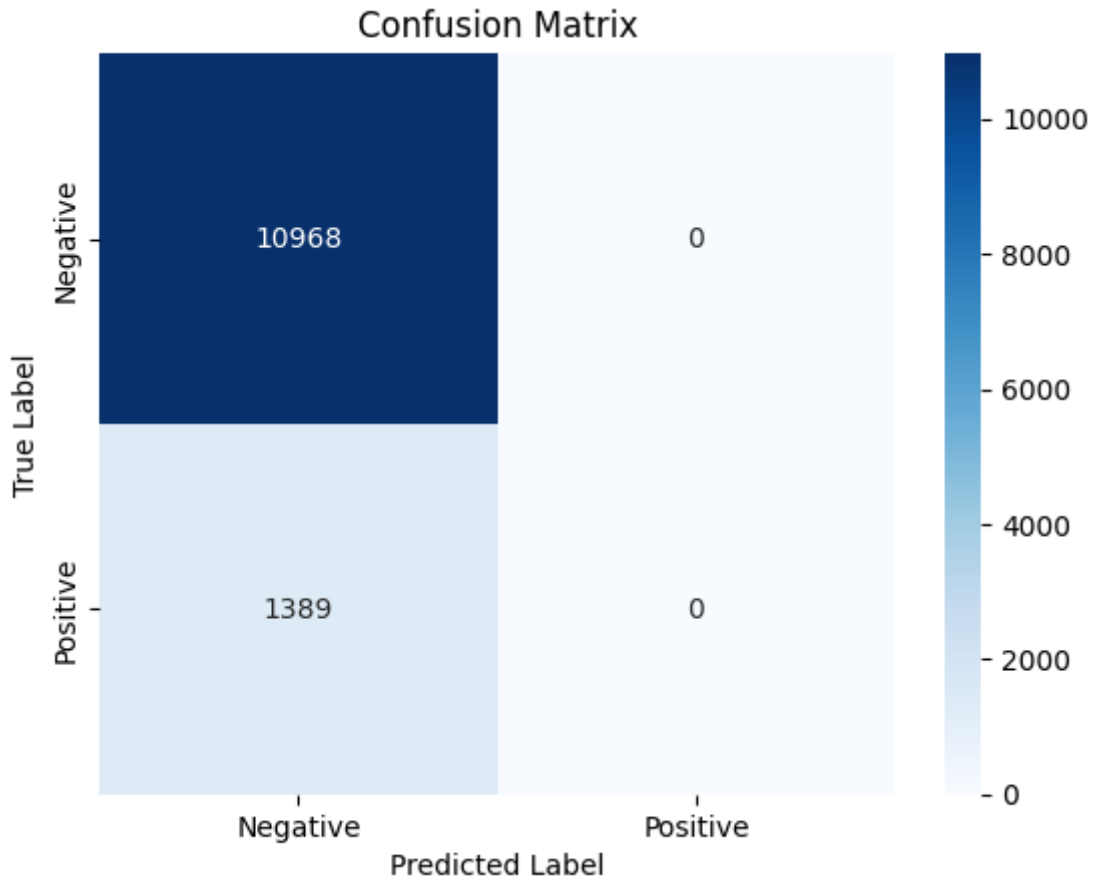
In [ ]:
```
dtree = DecisionTreeClassifier(max_depth=3)
```

In [ ]:
```
bagging = BaggingClassifier(estimator = dtree,
                  n_estimators= 100,
                  max_samples=.5,
                  max_features=1.)
bagging.fit(X_scaled, y_train)
y_pred=bagging.predict(X_test)
y_true = y_test
cm_raw = confusion_matrix(y_true, y_pred)
```

```
/Users/laoga/anaconda3/lib/python3.11/site-packages/sklearn/ensemble/_baggi
ng.py:802: DataConversionWarning: A column-vector y was passed when a 1d ar
ray was expected. Please change the shape of y to (n_samples, ), for exampl
e using ravel().
  y = column_or_1d(y, warn=True)
/Users/laoga/anaconda3/lib/python3.11/site-packages/sklearn/base.py:439: Us
erWarning: X does not have valid feature names, but BaggingClassifier was f
itted with feature names
  warnings.warn(
```

In [ ]:
```
class_labels = ['Negative', 'Positive']

# Plot the confusion matrix as a heatmap
sns.heatmap(cm_raw, annot=True, fmt='d', cmap='Blues', xticklabels=class_lab
plt.title('Confusion Matrix')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.show()
```



In [ ]:

4.) Boost your tree

In [ ]:
```
from sklearn.ensemble import AdaBoostClassifier
```

In [ ]:
```
dtree = DecisionTreeClassifier(max_depth=3)
```

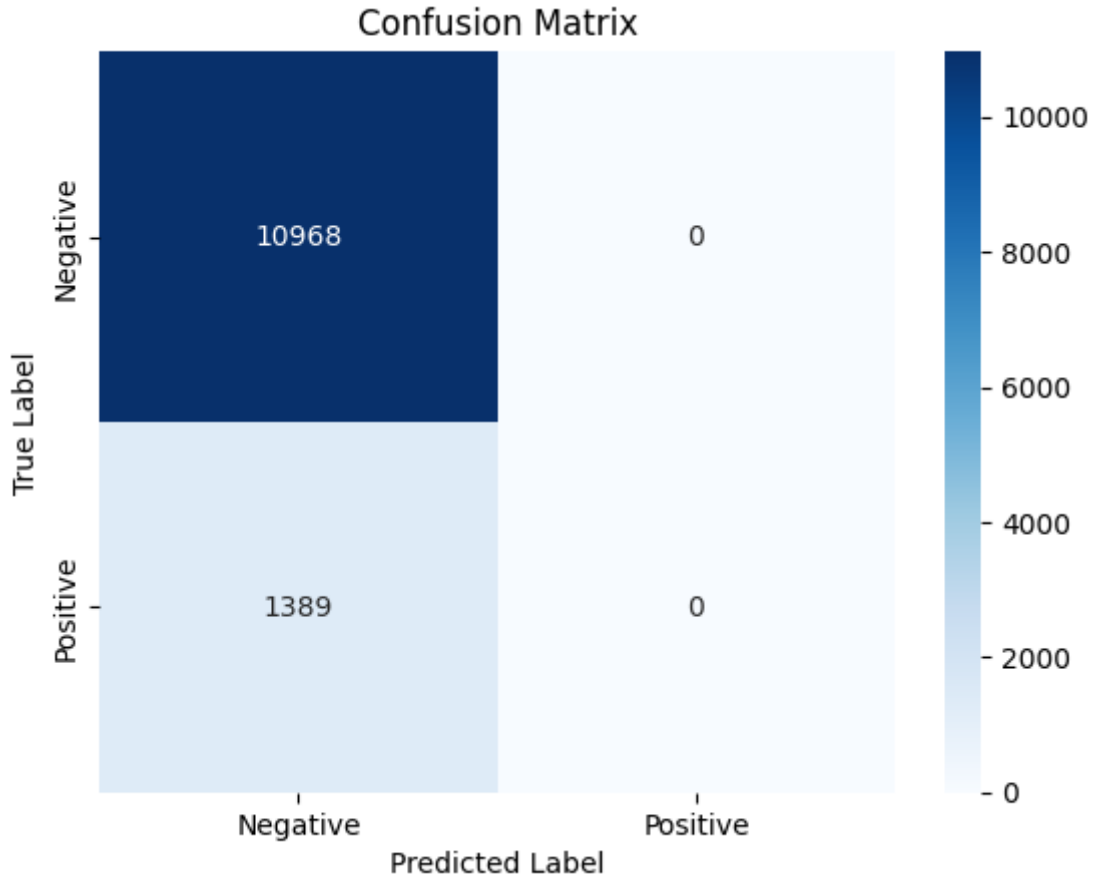
In [ ]:
```
boost=AdaBoostClassifier(estimator= dtree,
                         n_estimators=50)
boost.fit(X_scaled, y_train)
y_pred=bagging.predict(X_test)
y_true = y_test
cm_raw = confusion_matrix(y_true, y_pred)
```

```
/Users/laoga/anaconda3/lib/python3.11/site-packages/sklearn/utils/validatio
n.py:1143: DataConversionWarning: A column-vector y was passed when a 1d ar
ray was expected. Please change the shape of y to (n_samples, ), for exampl
e using ravel().
  y = column_or_1d(y, warn=True)
/Users/laoga/anaconda3/lib/python3.11/site-packages/sklearn/base.py:439: Us
erWarning: X does not have valid feature names, but BaggingClassifier was f
itted with feature names
  warnings.warn(
```

In [ ]:
```
class_labels = ['Negative', 'Positive']

# Plot the confusion matrix as a heatmap
sns.heatmap(cm_raw, annot=True, fmt='d', cmap='Blues', xticklabels=class_lab
plt.title('Confusion Matrix')
plt.xlabel('Predicted Label')
```

```
plt.ylabel('True Label')
plt.show()
```



In [ ]:

5.) Create a superlearner with at least 4 base learner models. Use a logistic reg for your metalearner. Interpret your coefficients and save your CM.

In [ ]:
```
pip install mlens
```

```
Collecting mlens
  Downloading mlens-0.2.3-py2.py3-none-any.whl (227 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 227.7/227.7 kB 2.0 MB/s eta 0:
00:0000:0100:01
Requirement already satisfied: numpy>=1.11 in /Users/laoga/anaconda3/lib/py
thon3.11/site-packages (from mlens) (1.23.5)
Requirement already satisfied: scipy>=0.17 in /Users/laoga/anaconda3/lib/py
thon3.11/site-packages (from mlens) (1.10.1)
Installing collected packages: mlens
Successfully installed mlens-0.2.3
Note: you may need to restart the kernel to use updated packages.
```

In [ ]:
```
from sklearn.linear_model import LogisticRegression
import numpy as np
```

In [ ]:
```
X_base_learners=[list(bagging.predict(X_scaled)),list(boost.predict(X_scaled
```

In [ ]:
```
super_learner=LogisticRegression()
super_learner.fit(np.column_stack(X_base_learners), y_train)
super_learner.coef_
```

```
/Users/laoga/anaconda3/lib/python3.11/site-packages/sklearn/utils/validation.py:1143: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

Out[ ]:
```
array([[0.83634915, 3.11668978, 0.05835552]])
```