HW 8

March 3, 2024

1 0.) Import and Clean data

[22]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
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
from imblearn.over_sampling import KMeansSMOTE
from sklearn.ensemble import AdaBoostClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
```

[2]:
```
data = pd.read_csv('bank-additional-full (1).csv', sep=";")
data
```

[2]:

	age	job	marital	education	default	housing	loan	\
0	56	housemaid	married	basic.4y	no	no	no	
1	57	services	married	high.school	unknown	no	no	
2	37	services	married	high.school	no	yes	no	
3	40	admin.	married	basic.6y	no	no	no	
4	56	services	married	high.school	no	no	yes	
…	…	…	…	…	…	…	…	…
41183	73	retired	married	professional.course	no	yes	no	
41184	46	blue-collar	married	professional.course	no	no	no	
41185	56	retired	married	university.degree	no	yes	no	
41186	44	technician	married	professional.course	no	no	no	
41187	74	retired	married	professional.course	no	yes	no	

```
         contact month day_of_week  …  campaign  pdays  previous  \
0      telephone   may         mon  …         1    999         0
1      telephone   may         mon  …         1    999         0
2      telephone   may         mon  …         1    999         0
3      telephone   may         mon  …         1    999         0
4      telephone   may         mon  …         1    999         0
…            …   …           …  …  …         …    …       …
41183   cellular   nov         fri  …         1    999         0
41184   cellular   nov         fri  …         1    999         0
41185   cellular   nov         fri  …         2    999         0
41186   cellular   nov         fri  …         1    999         0
41187   cellular   nov         fri  …         3    999         1

          poutcome emp.var.rate  cons.price.idx  cons.conf.idx  euribor3m  \
0      nonexistent          1.1          93.994          -36.4      4.857
1      nonexistent          1.1          93.994          -36.4      4.857
2      nonexistent          1.1          93.994          -36.4      4.857
3      nonexistent          1.1          93.994          -36.4      4.857
4      nonexistent          1.1          93.994          -36.4      4.857
…              …          …              …              …        …
41183  nonexistent         -1.1          94.767          -50.8      1.028
41184  nonexistent         -1.1          94.767          -50.8      1.028
41185  nonexistent         -1.1          94.767          -50.8      1.028
41186  nonexistent         -1.1          94.767          -50.8      1.028
41187      failure         -1.1          94.767          -50.8      1.028

       nr.employed    y
0           5191.0   no
1           5191.0   no
2           5191.0   no
3           5191.0   no
4           5191.0   no
…              …  …
41183       4963.6  yes
41184       4963.6   no
41185       4963.6   no
41186       4963.6  yes
41187       4963.6   no

[41188 rows x 21 columns]
```

[4]:
```
data = data.drop(["default",␣
 ↪"pdays",        "previous",        "poutcome",        "emp.var.
 ↪rate",        "cons.price.idx",        "cons.conf.
 ↪idx",        "euribor3m",        "nr.employed"], axis = 1)
```

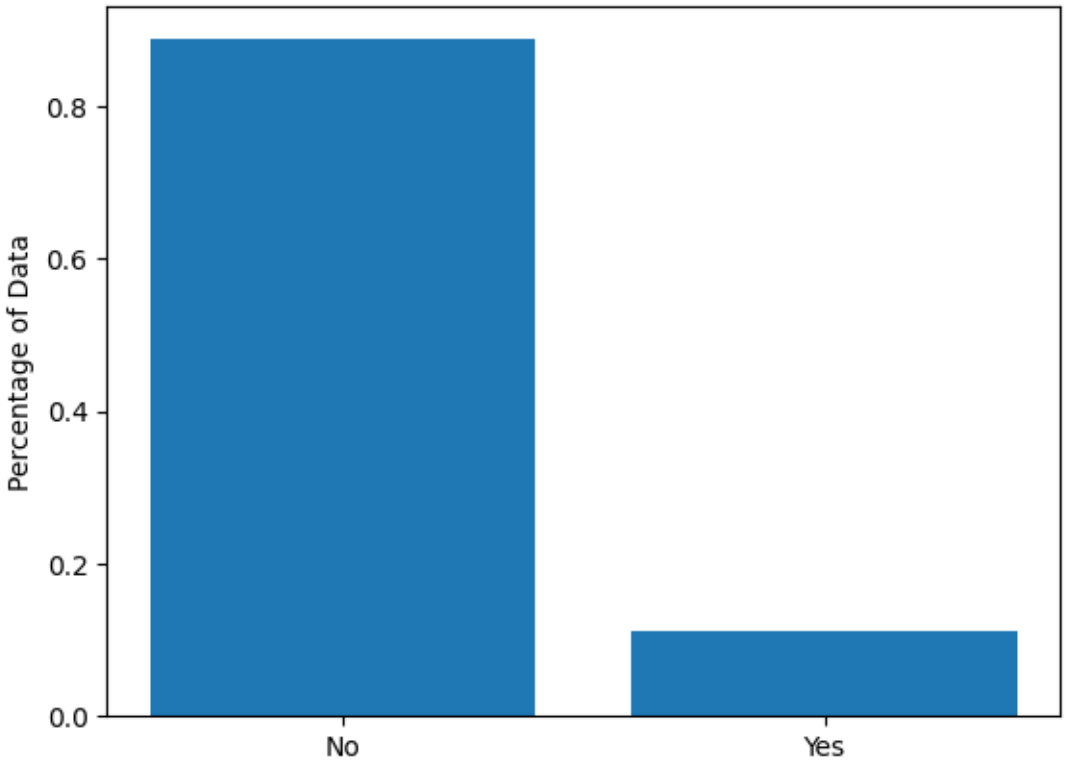
```
data = pd.get_dummies(data, columns = ["loan",␣
 ↪"job","marital","housing","contact","day_of_week", "campaign", "month",␣
 ↪"education"],drop_first = True)

data
```

[4]:

```
       age  duration    y  loan_unknown  loan_yes  job_blue-collar  \
0       56       261   no         False     False            False
1       57       149   no         False     False            False
2       37       226   no         False     False            False
3       40       151   no         False     False            False
4       56       307   no         False      True            False
…      …        …  …           …        …              …
41183   73       334  yes         False     False            False
41184   46       383   no         False     False             True
41185   56       189   no         False     False            False
41186   44       442  yes         False     False            False
41187   74       239   no         False     False            False

       job_entrepreneur  job_housemaid  job_management  job_retired  …  \
0                 False           True           False        False  …
1                 False          False           False        False  …
2                 False          False           False        False  …
3                 False          False           False        False  …
4                 False          False           False        False  …
…                 …              …               …            …  …
41183             False          False           False         True  …
41184             False          False           False        False  …
41185             False          False           False         True  …
41186             False          False           False        False  …
41187             False          False           False         True  …

       month_nov  month_oct  month_sep  education_basic.6y  \
0          False      False      False               False
1          False      False      False               False
2          False      False      False               False
3          False      False      False                True
4          False      False      False               False
…          …          …          …                   …
41183       True      False      False               False
41184       True      False      False               False
41185       True      False      False               False
41186       True      False      False               False
41187       True      False      False               False

       education_basic.9y  education_high.school  education_illiterate  \
0                   False                  False                 False
```

```
1                   False                    True                   False
2                   False                    True                   False
3                   False                   False                   False
4                   False                    True                   False
…                   …                        …                      …
41183               False                   False                   False
41184               False                   False                   False
41185               False                   False                   False
41186               False                   False                   False
41187               False                   False                   False

       education_professional.course  education_university.degree  \
0                              False                         False
1                              False                         False
2                              False                         False
3                              False                         False
4                              False                         False
…                              …                             …
41183                           True                         False
41184                           True                         False
41185                          False                          True
41186                           True                         False
41187                           True                         False

       education_unknown
0                  False
1                  False
2                  False
3                  False
4                  False
…                  …
41183              False
41184              False
41185              False
41186              False
41187              False

[41188 rows x 83 columns]
```

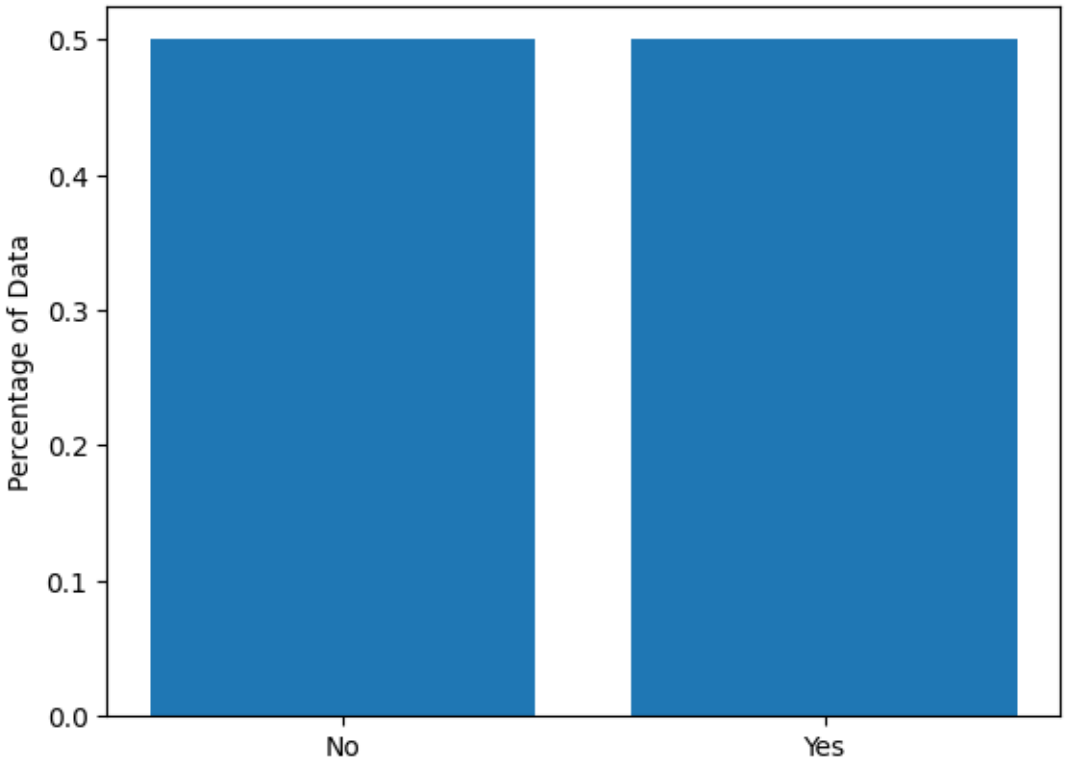
[5]:
```
y = pd.get_dummies(data["y"], drop_first = True)
X = data.drop(["y"], axis = 1)

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()
```



[6]:
```
# Train Test Split
X_train, X_test, y_train, y_test = train_test_split(X.astype(int), y.
 ↪astype(int), test_size=0.3, random_state=42, )
```

[7]:
```
# We use the SMOTE algorithm to balance the data set

smote = KMeansSMOTE(random_state = 42)

X_train_smote, y_train_smote = smote.fit_resample(X_train, y_train)
```

[8]:
```
obs = len(y_train_smote)
plt.bar(["No","Yes"],[len(y_train_smote[y_train_smote.yes==0])/
 ↪obs,len(y_train_smote[y_train_smote.yes==1])/obs])
plt.ylabel("Percentage of Data")
plt.show()
```



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

[9]:
```
dtree_main = DecisionTreeClassifier(max_depth = 3)
dtree_main.fit(X_train_smote, y_train_smote)
```

[9]:
```
DecisionTreeClassifier(max_depth=3)
```

[10]:
```
X_train_smote.head()
```

[10]:

	age	duration	loan_unknown	loan_yes	job_blue-collar	job_entrepreneur	\
0	29	77	0	0	0	0	
1	29	12	0	0	0	0	
2	45	277	0	0	1	0	
3	34	70	0	0	0	0	
4	32	1181	0	0	0	0	

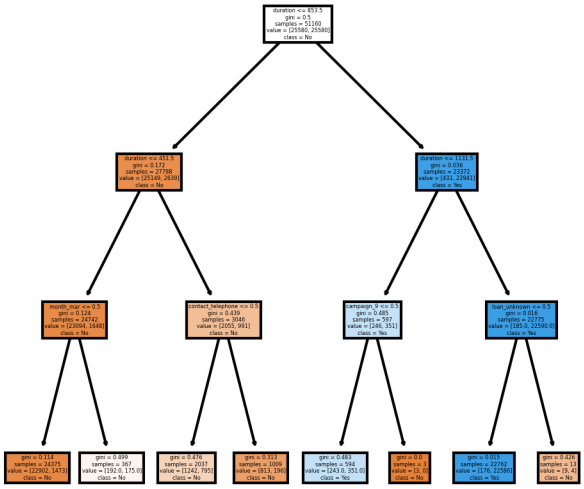
	job_housemaid	job_management	job_retired	job_self-employed	…	\
0	0	0	0	0	…	
1	0	0	0	0	…	
2	0	0	0	0	…	

```
3              0              0              0                  0  …
4              0              0              0                  0  …

   month_nov  month_oct  month_sep  education_basic.6y  education_basic.9y  \
0          0          0          0                   0                   0
1          0          0          0                   0                   0
2          0          0          0                   1                   0
3          0          0          0                   0                   0
4          0          0          0                   0                   0

   education_high.school  education_illiterate  education_professional.course  \
0                      0                     0                              0
1                      0                     0                              0
2                      0                     0                              0
3                      0                     0                              0
4                      1                     0                              0

   education_university.degree  education_unknown
0                            1                  0
1                            1                  0
2                            0                  0
3                            1                  0
4                            0                  0

[5 rows x 82 columns]
```

[11]:
```
fig, axes = plt.subplots(nrows = 1,ncols = 1,figsize = (4,4), dpi=300)
plot_tree(dtree_main, filled = True, feature_names = list(X_train_smote.
 ↪columns), class_names=["No","Yes"])
```

[11]:
```
[Text(0.5, 0.875, 'duration <= 853.5\ngini = 0.5\nsamples = 51160\nvalue =
[25580, 25580]\nclass = No'),
 Text(0.25, 0.625, 'duration <= 451.5\ngini = 0.172\nsamples = 27788\nvalue =
[25149, 2639]\nclass = No'),
 Text(0.125, 0.375, 'month_mar <= 0.5\ngini = 0.124\nsamples = 24742\nvalue =
[23094, 1648]\nclass = No'),
 Text(0.0625, 0.125, 'gini = 0.114\nsamples = 24375\nvalue = [22902,
1473]\nclass = No'),
 Text(0.1875, 0.125, 'gini = 0.499\nsamples = 367\nvalue = [192.0, 175.0]\nclass
= No'),
 Text(0.375, 0.375, 'contact_telephone <= 0.5\ngini = 0.439\nsamples =
3046\nvalue = [2055, 991]\nclass = No'),
 Text(0.3125, 0.125, 'gini = 0.476\nsamples = 2037\nvalue = [1242, 795]\nclass =
No'),
 Text(0.4375, 0.125, 'gini = 0.313\nsamples = 1009\nvalue = [813, 196]\nclass =
No'),
 Text(0.75, 0.625, 'duration <= 1131.5\ngini = 0.036\nsamples = 23372\nvalue =
```

```
[431, 22941]\nclass = Yes'),
 Text(0.625, 0.375, 'campaign_9 <= 0.5\ngini = 0.485\nsamples = 597\nvalue =
[246, 351]\nclass = Yes'),
 Text(0.5625, 0.125, 'gini = 0.483\nsamples = 594\nvalue = [243.0, 351.0]\nclass
= Yes'),
 Text(0.6875, 0.125, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]\nclass = No'),
 Text(0.875, 0.375, 'loan_unknown <= 0.5\ngini = 0.016\nsamples = 22775\nvalue =
[185.0, 22590.0]\nclass = Yes'),
 Text(0.8125, 0.125, 'gini = 0.015\nsamples = 22762\nvalue = [176, 22586]\nclass
= Yes'),
 Text(0.9375, 0.125, 'gini = 0.426\nsamples = 13\nvalue = [9, 4]\nclass = No')]
```

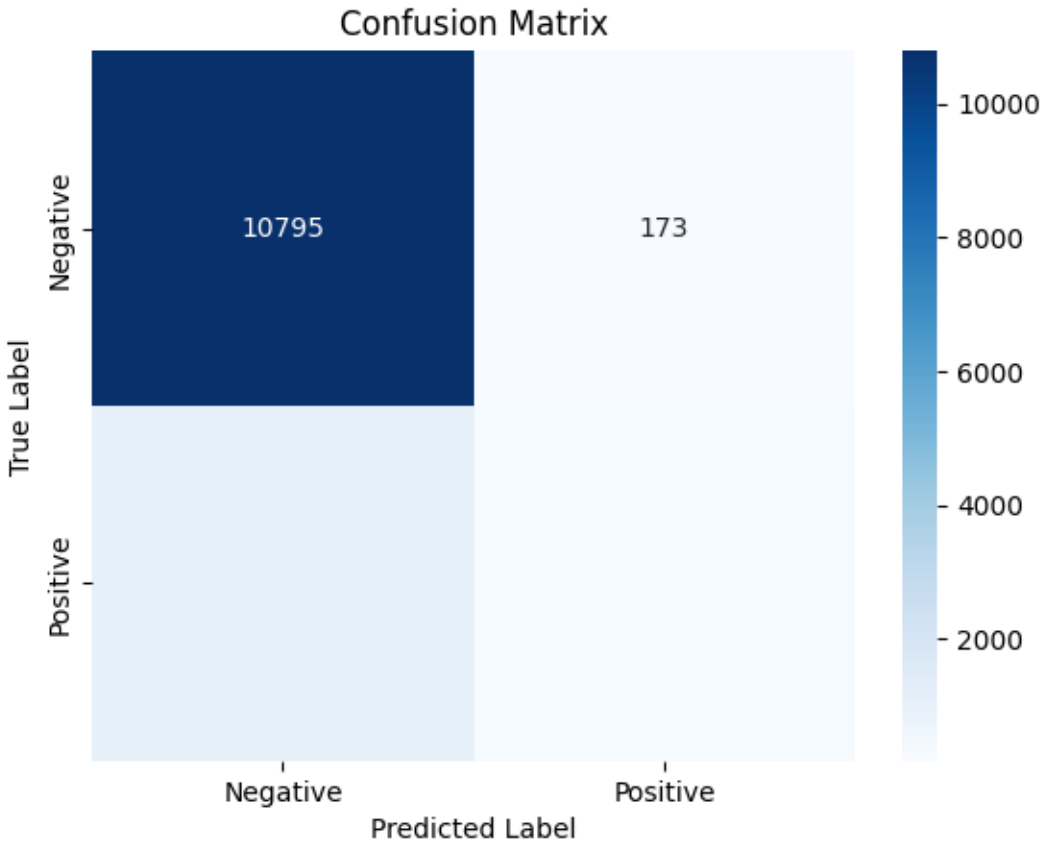


3 1b.) Confusion matrix on out of sample data. Visualize and store as variable

[12]:
```
y_pred_main = dtree_main.predict(X_test)
y_true = y_test
cm_raw = confusion_matrix(y_true, y_pred_main)
```

[13]:
```
class_labels = ['Negative', 'Positive']

# Plot the confusion matrix as a heatmap
sns.heatmap(cm_raw, annot=True, fmt='d', cmap='Blues',␣
 ↪xticklabels=class_labels, yticklabels=class_labels)
plt.title('Confusion Matrix')
```

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



4 3.) Use bagging on your descision tree

[14]:
```
bagging_dtree = DecisionTreeClassifier(max_depth=3)
```

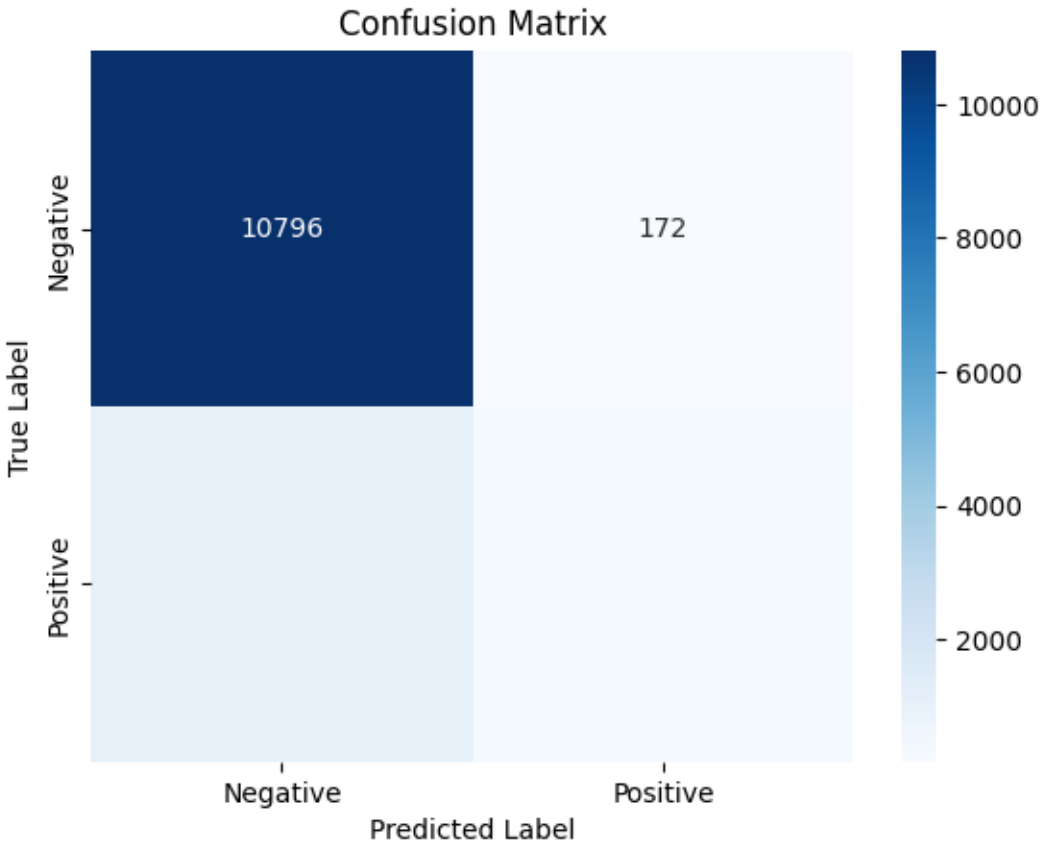
[15]:
```
bagging= BaggingClassifier(estimator=bagging_dtree,
                  n_estimators= 100,
                  max_samples= .5,
                  max_features=1.)

bagging.fit(X_train_smote, y_train_smote)

y_pred_bagging = bagging.predict
```

```
/Users/raghavirajumohan/anaconda3/lib/python3.11/site-
packages/sklearn/ensemble/_bagging.py:782: 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)
```

[16]:
```
y_pred_bagging = bagging.predict(X_test)
y_true = y_test
cm_raw = confusion_matrix(y_true, y_pred_bagging)
```

[17]:
```
class_labels = ['Negative', 'Positive']

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



5 4.) Boost your tree

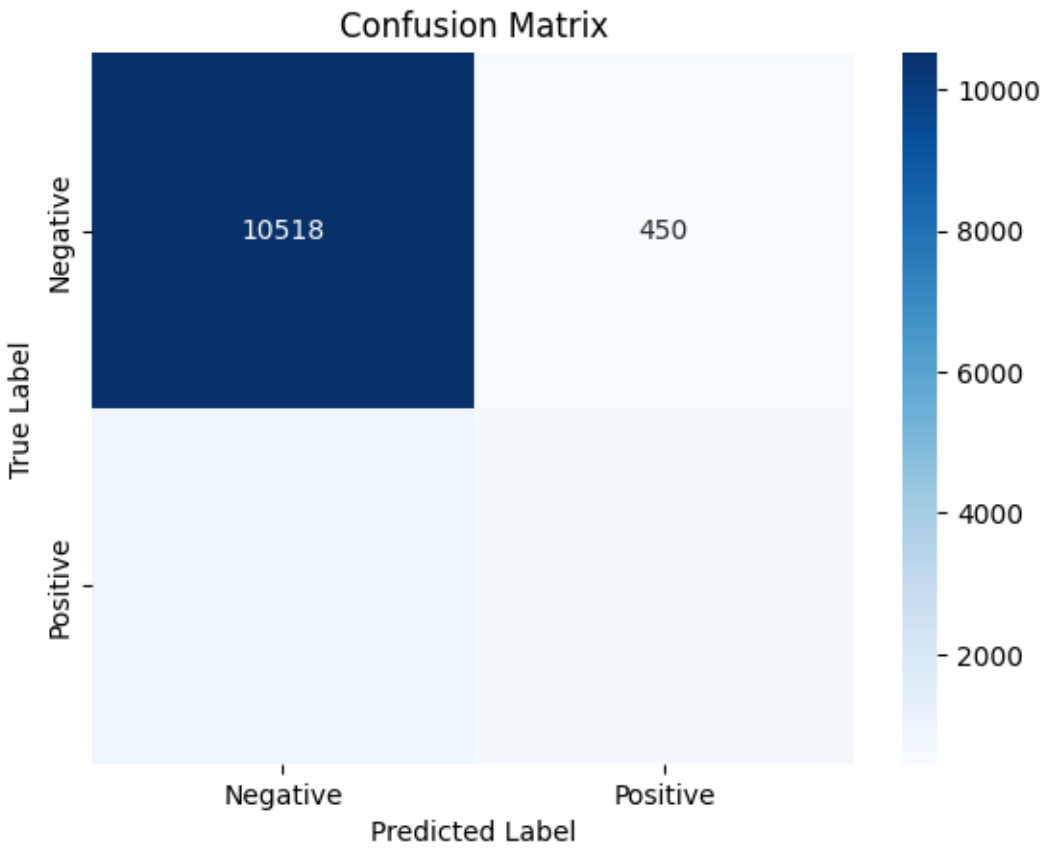
[27]:
```
boost_dtree = DecisionTreeClassifier(max_depth=3)
```

[28]:
```
boost = AdaBoostClassifier(estimator=boost_dtree,
                   n_estimators= 100,
                   )

boost.fit(X_train_smote, y_train_smote)

y_pred_boost = boost.predict
```

```
/Users/raghavirajumohan/anaconda3/lib/python3.11/site-
packages/sklearn/utils/validation.py:1229: 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)
/Users/raghavirajumohan/anaconda3/lib/python3.11/site-
packages/sklearn/ensemble/_weight_boosting.py:519: FutureWarning: The SAMME.R
algorithm (the default) is deprecated and will be removed in 1.6. Use the SAMME
algorithm to circumvent this warning.
  warnings.warn(
```

[29]:
```
y_pred_boost = boost.predict(X_test)
y_true = y_test
cm_raw = confusion_matrix(y_true, y_pred_boost)
```

[30]:
```
class_labels = ['Negative', 'Positive']

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



6 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.

[31]:
```
bagging.predict(X_train_smote)
boost.predict(X_train_smote)
dtree_main.predict(X_train_smote)
```

[31]:
```
array([0, 0, 0, …, 1, 1, 1])
```

[32]:
```
# Put the in a list
X_base_learners = np.array(
    [
        (bagging.predict_proba(X_train_smote)[:,1]),
        boost.predict_proba(X_train_smote)[:,1],
        dtree_main.predict_proba(X_train_smote)[:,1],
     ]).T
```

[33]:
```
X_base_learners = pd.DataFrame(X_base_learners)
supper_learner = LogisticRegression()
supper_learner.fit(X_base_learners, y_train_smote)
```

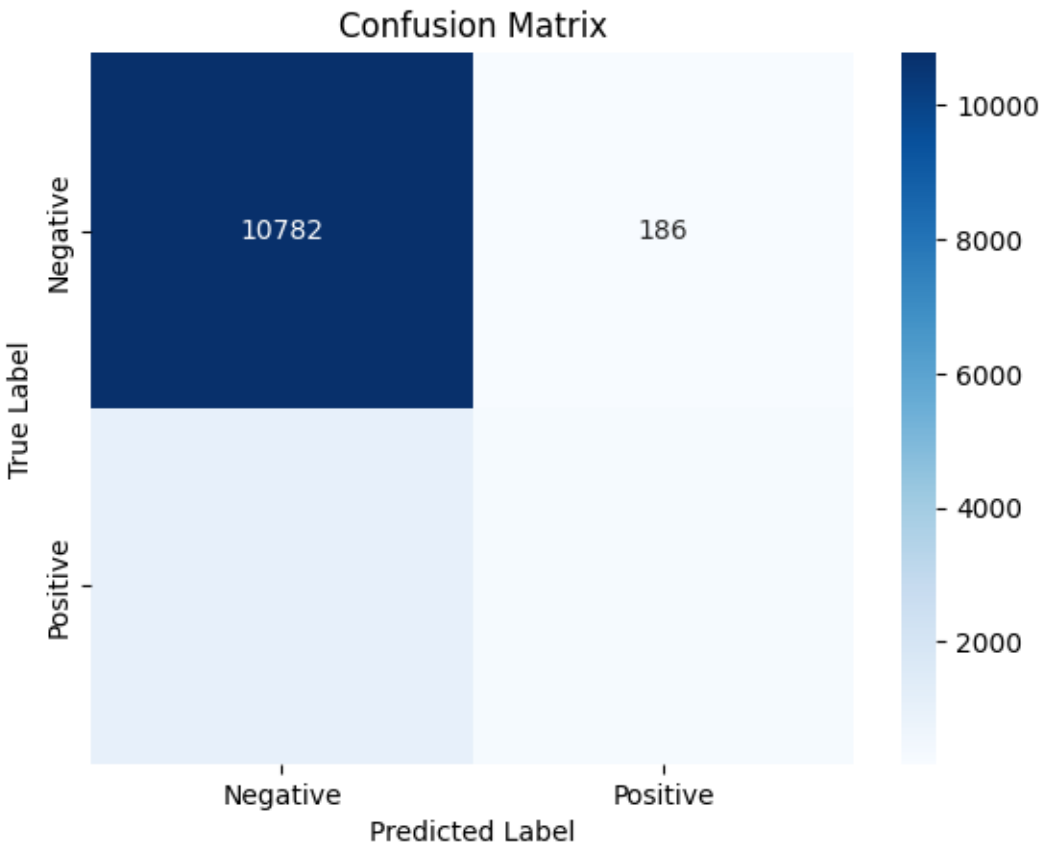
```
/Users/raghavirajumohan/anaconda3/lib/python3.11/site-
packages/sklearn/utils/validation.py:1229: DataConversionWarning: A column-
vector y was passed when a 1d array was expected. Please change the shape of y
to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

[33]:
```
LogisticRegression()
```

[34]:
```
# X test transfromed
X_test_learner = np.array(
    [
        (bagging.predict_proba(X_test)[:,1]),
        boost.predict_proba(X_test)[:,1],
        dtree_main.predict_proba(X_test)[:,1],
     ]).T
```

[35]:
```
y_pred_super_learner = supper_learner.predict(X_test_learner)
y_true = y_test
cm_raw = confusion_matrix(y_true, y_pred_super_learner)
```

[36]:
```
class_labels = ['Negative', 'Positive']

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



[37]:
```
supper_learner.coef_
```

[37]:
```
array([[ 6.82209283, 25.11383276, -0.38364038]])
```

Boosting stands out as the most influential among the base learners, significantly shaping the superlearner's predictions. Bagging follows closely behind, contributing notably to the ensemble's performance. Conversely, the Decision Tree model, while included, has a minor impact and even introduces a negative effect on the final output. Boosting's superior performance underscores its effectiveness in capturing intricate patterns and enhancing predictive accuracy. While Bagging plays a crucial role in diversifying predictions and improving overall robustness, the inclusion of the Decision Tree model does not lead to significant changes in the ensemble's predictive power, especially when compared to the substantial gains observed with Boosting. This emphasizes the importance of selecting and prioritizing influential base learners in ensemble methods to optimize overall performance.