

0.) Import and Clean data

```
In [ ]: import pandas as pd  
# from colab import drive  
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)
```

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

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

```
Out[ ]:   age      job marital education default housing loan contact month day_of_week ... campaign pdays previous poutcome emp.  
0  56  housemaid married basic.4y    no     no  no  telephone  may  mon ... 1  999  0 nonexistent  
1  57    services married high.school unknown    no  no  no  telephone  may  mon ... 1  999  0 nonexistent  
2  37    services married high.school    no  yes  no  telephone  may  mon ... 1  999  0 nonexistent  
3  40  admin. married basic.6y    no  no  no  telephone  may  mon ... 1  999  0 nonexistent  
4  56    services married high.school    no  no  yes  telephone  may  mon ... 1  999  0 nonexistent
```

5 rows × 21 columns

```
In [ ]: df = df.drop(['default', 'pdays', "previous", "poutcome", "emp.var.rate", "cons.price.idx", "cons.conf.idx"],  
df = pd.get_dummies(df, columns = ["loan", "job", "marital", "housing", "contact", "day_of_week", "campaign", "month", "education"],
```

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

```
Out[ ]:   age duration y loan_unknown loan_yes job_blue-collar job_entrepreneur job_housemaid job_management job_retired ... month_nov month_oc  
0  56     261  no    False  False  False  False  True  False  False  False ...  False  False  
1  57     149  no    False  False  False  False  False  False  False  False ...  False  False  
2  37     226  no    False  False  False  False  False  False  False  False ...  False  False  
3  40     151  no    False  False  False  False  False  False  False  False ...  False  False  
4  56     307  no    False  True  False  False  False  False  False  False ...  False  False
```

5 rows × 83 columns

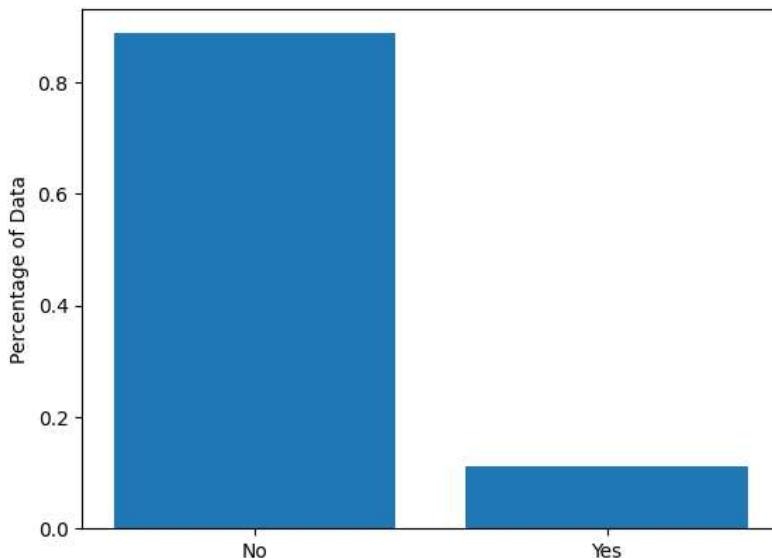
```
In [ ]: y = pd.get_dummies(df['y'], drop_first = True)  
X = df.drop(['y'], axis = 1)
```

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

```
Out[ ]:   age duration loan_unknown loan_yes job_blue-collar job_entrepreneur job_housemaid job_management job_retired job_self-employed ... month_nov  
39075  29       77    False  False  False  False  False  False  False  False  False ...  
34855  29       12    False  False  False  False  False  False  False  False  False ...  
7107   45      277    False  False  True  False  False  False  False  False  False ...  
31614  34       70    False  False  False  False  False  False  False  False  False ...  
34878  32     1181    False  False  False  False  False  False  False  False  False ...  
5 rows × 82 columns
```

```
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 [ ]: a = np.arange(2, 80)
a
```

```
Out[ ]: array([ 2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
       19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35,
       36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52,
       53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69,
       70, 71, 72, 73, 74, 75, 76, 77, 78, 79])
```

```
In [ ]: x_scaled
```

```
Out[ ]:
```

	age	duration	loan_unknown	loan_yes	job_blue-collar	job_entrepreneur	job_housemaid	job_management	job_retired	job_self-employed	...	month_no
0	29	77	False	False	False	False	False	False	False	False	...	False
1	29	12	False	False	False	False	False	False	False	False	...	False
2	45	277	False	False	True	False	False	False	False	False	...	False
3	34	70	False	False	False	False	False	False	False	False	...	False
4	32	1181	False	False	False	False	False	False	False	False	...	False
...
51155	38	748	False	False	False	False	False	False	False	False	...	False
51156	38	654	False	False	False	False	False	False	False	False	...	False
51157	25	1042	False	False	False	False	False	False	False	False	...	False
51158	24	450	False	False	False	False	False	False	False	False	...	False
51159	30	857	False	False	False	False	False	False	False	False	...	False

51160 rows × 82 columns

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

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

# Train Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Update the range of categorical_features
smote = SMOTENC(categorical_features = np.arange(2, 82))
#smote = SMOTE()
x_scaled, y_scaled = smote.fit_resample(X_train, y_train)

#scaler = StandardScaler().fit(X_train)

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

```
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:767: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.  
.. if not hasattr(array, "sparse") and array.dtypes.apply(is_sparse).any():  
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:767: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.  
.. if not hasattr(array, "sparse") and array.dtypes.apply(is_sparse).any():  
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:605: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.  
.. if is_sparse(pd_dtype):  
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:614: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.  
.. if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
```

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_scaled)
```

```
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:767: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.  
.. if not hasattr(array, "sparse") and array.dtypes.apply(is_sparse).any():  
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:605: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.  
.. if is_sparse(pd_dtype):  
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:614: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.  
.. if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):  
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:767: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.  
.. if not hasattr(array, "sparse") and array.dtypes.apply(is_sparse).any():
```

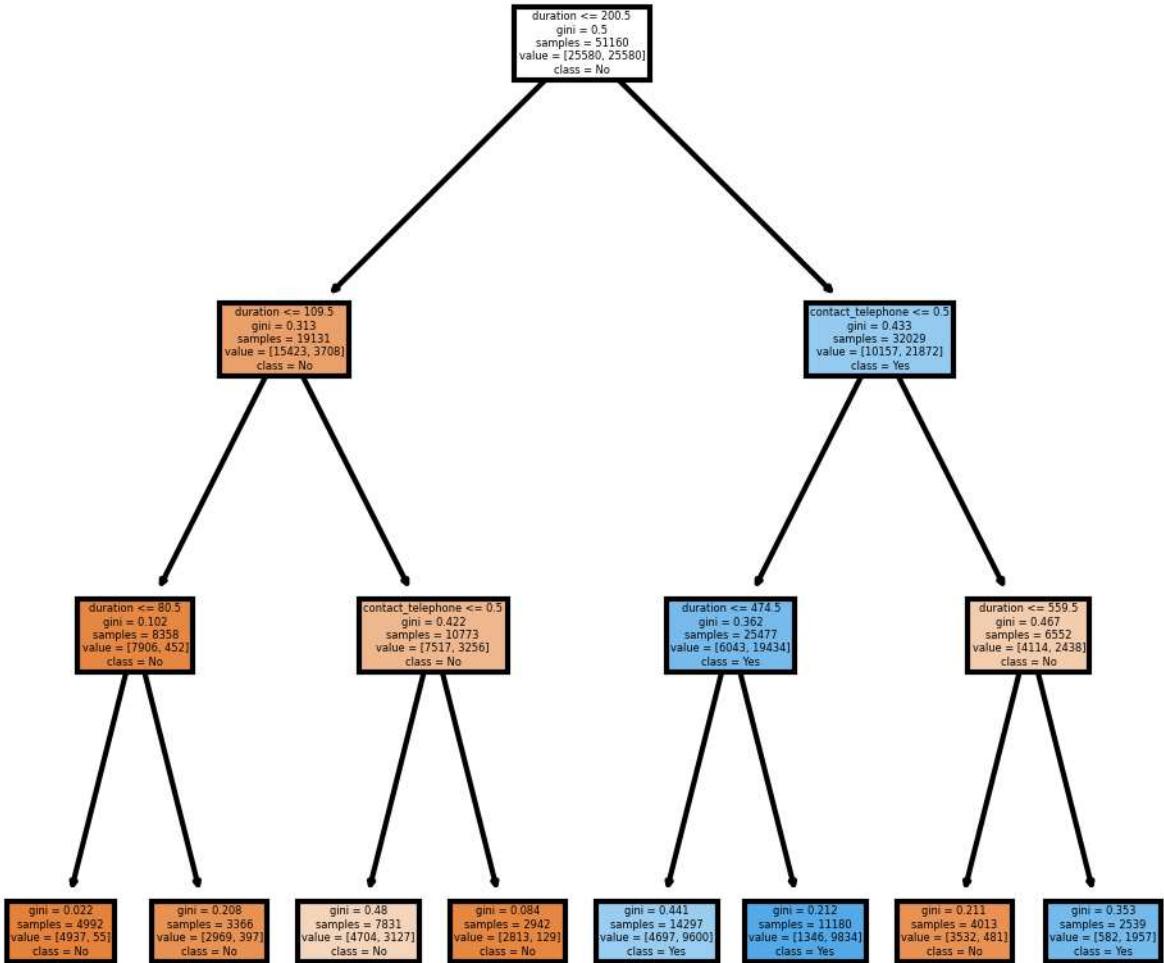
Out[]: ▾ DecisionTreeClassifier

```
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=["No", "Yes"])
```

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

```
Out[ ]: [Text(0.5, 0.875, 'duration <= 200.5\nngini = 0.5\nnsamples = 51160\nvalue = [25580, 25580]\nnclass = No'),  
Text(0.25, 0.625, 'duration <= 109.5\nngini = 0.313\nnsamples = 19131\nvalue = [15423, 3708]\nnclass = No'),  
Text(0.125, 0.375, 'duration <= 80.5\nngini = 0.102\nnsamples = 8358\nvalue = [7906, 452]\nnclass = No'),  
Text(0.0625, 0.125, 'gini = 0.022\nnsamples = 4992\nvalue = [4937, 55]\nnclass = No'),  
Text(0.1875, 0.125, 'gini = 0.208\nnsamples = 3366\nvalue = [2969, 397]\nnclass = No'),  
Text(0.375, 0.375, 'contact_telephone <= 0.5\nngini = 0.422\nnsamples = 10773\nvalue = [7517, 3256]\nnclass = No'),  
Text(0.3125, 0.125, 'gini = 0.48\nnsamples = 7831\nvalue = [4704, 3127]\nnclass = No'),  
Text(0.4375, 0.125, 'gini = 0.084\nnsamples = 2942\nvalue = [2813, 129]\nnclass = No'),  
Text(0.75, 0.625, 'contact_telephone <= 0.5\nngini = 0.433\nnsamples = 32029\nvalue = [10157, 21872]\nnclass = Yes'),  
Text(0.625, 0.375, 'duration <= 474.5\nngini = 0.362\nnsamples = 25477\nvalue = [6043, 19434]\nnclass = Yes'),  
Text(0.5625, 0.125, 'gini = 0.441\nnsamples = 14297\nvalue = [4697, 9600]\nnclass = Yes'),  
Text(0.6875, 0.125, 'gini = 0.212\nnsamples = 11180\nvalue = [1346, 9834]\nnclass = Yes'),  
Text(0.875, 0.375, 'duration <= 559.5\nngini = 0.467\nnsamples = 6552\nvalue = [4114, 2438]\nnclass = No'),  
Text(0.8125, 0.125, 'gini = 0.211\nnsamples = 4013\nvalue = [3532, 481]\nnclass = No'),  
Text(0.9375, 0.125, 'gini = 0.353\nnsamples = 2539\nvalue = [582, 1957]\nnclass = Yes')]
```



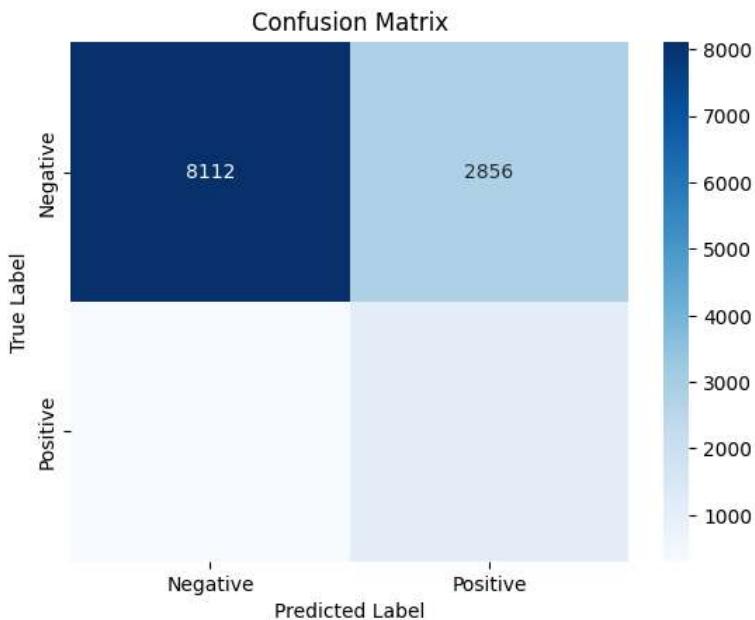
1b.) Confusion matrix on out of sample data. Visualize and store as variable

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

c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:767: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.
.. if not hasattr(array, "sparse") and array.dtypes.apply(is_sparse).any():
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:605: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.
.. if is_sparse(pd_dtype):
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:614: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.
.. if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:767: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.
.. if not hasattr(array, "sparse") and array.dtypes.apply(is_sparse).any():
```

```
In [ ]: 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()
```



3.) Use bagging on your descision tree

```
In [ ]: # optimize on max depth...
dtree_main = DecisionTreeClassifier(max_depth = 3)
```

```
In [ ]: bagging = BaggingClassifier(base_estimator=dtree_main,
n_estimators=100,
max_samples=.5,
max_features=1.)
```

```
bagging.fit(x_scaled, y_scaled)
y_pred = bagging.predict(X_test)
```

```
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:767: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.
.. if not hasattr(array, "sparse") and array.dtypes.apply(is_sparse).any():
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:605: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.
.. if is_sparse(pd_dtype):
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:614: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.
.. if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:767: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.
.. if not hasattr(array, "sparse") and array.dtypes.apply(is_sparse).any():
c:\ProgramData\anaconda3\lib\site-packages\sklearn\ensemble\_bagging.py:802: 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)
c:\ProgramData\anaconda3\lib\site-packages\sklearn\ensemble\_base.py:166: FutureWarning: `base_estimator` was renamed to `estimator` in version 1.2 and will be removed in 1.4.
.. warnings.warn(
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:767: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.
.. if not hasattr(array, "sparse") and array.dtypes.apply(is_sparse).any():
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:605: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.
.. if is_sparse(pd_dtype):
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:614: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.
.. if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
```

```
In [ ]: x_scaled
```

```
Out[ ]:      age duration loan_unknown loan_yes job_blue-collar job_entrepreneur job_housemaid job_management job_retired job_self-employed ... month_no
0    29        77   False   False   False   False   False   False   False   False   False   ...
1    29        12   False   False   False   False   False   False   False   False   False   ...
2    45       277   False   False    True   False   False   False   False   False   False   ...
3    34        70   False   False   False   False   False   False   False   False   False   ...
4    32      1181   False   False   False   False   False   False   False   False   False   ...
...   ...       ...   ...   ...   ...   ...   ...   ...   ...   ...   ...   ...
51155  38      1159   False   False   False   False   False   False   False   False   False   ...
51156  28       153   False   False   False   False   False   False   False   False   False   ...
51157  37      336   False   False   False   False   False   False   False   False   False   ...
51158  59       159   False   False   False   False   False   False   False   False   False   ...
51159  39       526   False   False    True   False   False   False   False   False   False   ...

51160 rows × 82 columns
```

In []: `y_scaled`

```
Out[ ]:      yes
0  False
1  False
2  False
3  False
4  False
...
51155  True
51156  True
51157  True
51158  True
51159  True
```

51160 rows × 1 columns

In []:

In []:

4.) Boost your tree

```
In [ ]: from sklearn.ensemble import AdaBoostClassifier
In [ ]: dtree_main = DecisionTreeClassifier(max_depth = 3)
```

5). Train a logistic regression on decision tredd, boosted tree, bagged tree

```
In [ ]: y_true = y_test
cm_raw = confusion_matrix(y_true, y_pred)

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()
```

```
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:767: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.
  if not hasattr(array, "sparse") and array.dtypes.apply(is_sparse).any():
```



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

Defaulting to user installation because normal site-packages is not writeable
Note: you may need to restart the kernel to use updated packages.

Requirement already satisfied: mlens in c:\users\thinkpad\appdata\roaming\python\python310\site-packages (0.2.3)
Requirement already satisfied: scipy>=0.17 in c:\programdata\anaconda3\lib\site-packages (from mlens) (1.10.0)
Requirement already satisfied: numpy>=1.11 in c:\programdata\anaconda3\lib\site-packages (from mlens) (1.23.5)

```
In [ ]: from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier

# from mlens.ensemble import SuperLearner
```

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

```
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:767: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.
  if not hasattr(array, "sparse") and array.dtypes.apply(is_sparse).any():
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:605: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.
  if is_sparse(pd_dtype):
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:614: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.
  if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
c:\ProgramData\anaconda3\lib\site-packages\sklearn\base.py:413: UserWarning: X has feature names, but BaggingClassifier was fitted without feature names
  warnings.warn(
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:767: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.
  if not hasattr(array, "sparse") and array.dtypes.apply(is_sparse).any():
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:605: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.
  if is_sparse(pd_dtype):
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:614: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.
  if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:767: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.
  if not hasattr(array, "sparse") and array.dtypes.apply(is_sparse).any():
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:605: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.
  if is_sparse(pd_dtype):
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:614: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.
  if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
```

```
In [ ]: super_learner = LogisticRegression()
```

```
In [ ]: # Assuming bagging, boost, and dtree_main are your base models
predictions = [bagging.predict(X_train), boost.predict(X_train), dtree_main.predict(X_train)]

# Stack predictions horizontally to create a feature matrix for the super learner
# Each model's predictions become a column in the matrix
X_base_learners = np.column_stack(predictions)

# Now fit the super learner
super_learner = LogisticRegression()
super_learner.fit(X_base_learners, y_train)
```

```
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:767: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.  
    if not hasattr(array, "sparse") and array.dtypes.apply(is_sparse).any():  
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:605: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.  
    if is_sparse(pd_dtype):  
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:614: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.  
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):  
c:\ProgramData\anaconda3\lib\site-packages\sklearn\base.py:413: UserWarning: X has feature names, but BaggingClassifier was fitted without feature names  
    warnings.warn(  
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:767: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.  
    if not hasattr(array, "sparse") and array.dtypes.apply(is_sparse).any():  
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:605: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.  
    if is_sparse(pd_dtype):  
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:614: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.  
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):  
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:767: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.  
    if not hasattr(array, "sparse") and array.dtypes.apply(is_sparse).any():  
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:605: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.  
    if is_sparse(pd_dtype):  
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:614: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.  
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):  
c:\ProgramData\anaconda3\lib\site-packages\sklearn\utils\validation.py:767: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.  
    if not hasattr(array, "sparse") and array.dtypes.apply(is_sparse).any():  
c:\ProgramData\anaconda3\lib\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[ ]: ▾ LogisticRegression  
        LogisticRegression()
```

```
In [ ]: super_learner.coef_
array([[1.30680608, 4.9456112 , 0.20863634]])
```

6.)

The boost model stands out as the best, as it accurately categorizes the highest number of labels. Specifically, the boost model exhibits the highest coefficient. This indicates that the boost model significantly enhances the meta model's ability to make accurate predictions.