CMSE 381, Fundamental Data Science Methods

November 9, 2025

Homework 7

Lowell Monis

Instructor: Dr. Mengsen Zhang

Question 1: ISLP § 8.4.4

This question relates to the plots in Figure 8.14 in ISLP. A copy of the figure is attached below.

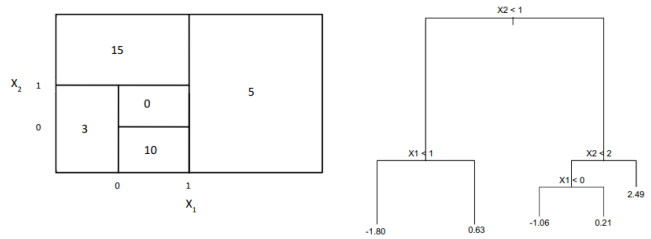
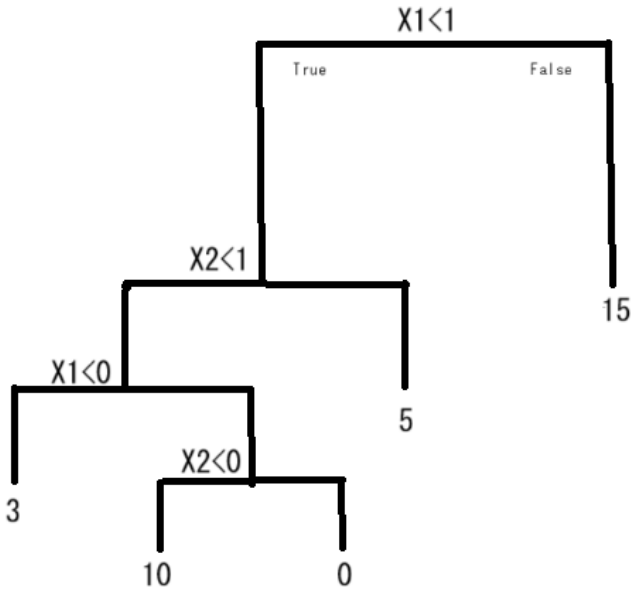


FIGURE 8.14. Left: *A partition of the predictor space corresponding to Exercise 4a.* Right: *A tree corresponding to Exercise 4b.*

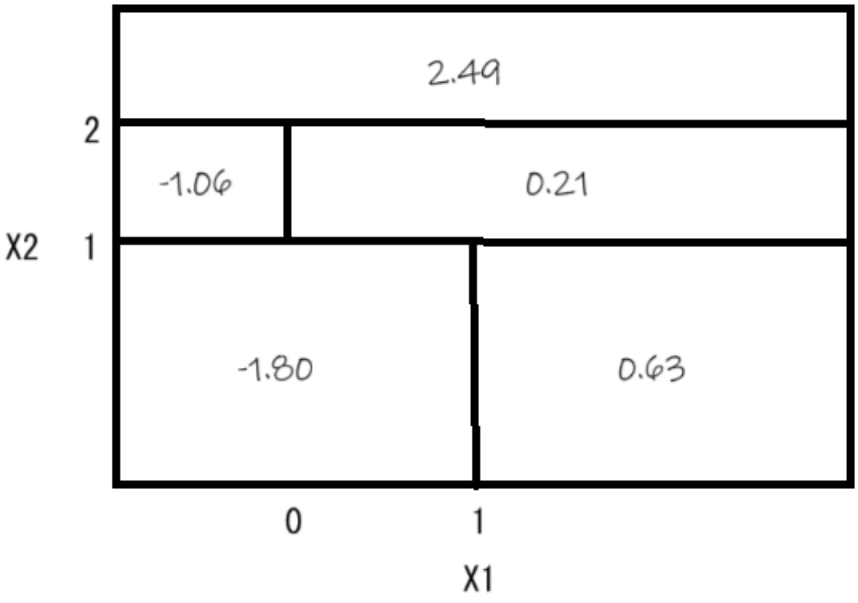
(a) Sketch the tree corresponding to the partition of the predictor space illustrated in the left-hand panel of Figure 8.14. The numbers inside the boxes indicate the mean of Y within each region. A text-based visualization of the corresponding tree for the given partitioned predictor space is as follows:

- If $X_1 \geq 1$, then $Y = 5$ (leaf node).
- If $X_1 < 1$, then:
  - If $X_2 \geq 1$, then $Y = 15$ (leaf node).
  - If $X_2 < 1$, then:
    - If $X_1 < 0$, then $Y = 3$ (leaf node).
    - If $X_1 \geq 0$, then:
      - If $X_2 \geq 0$, then $Y = 0$ (leaf node).
      - If $X_2 < 0$, then $Y = 10$ (leaf node).

The sketch I created is provided below:



(b) Create a diagram similar to the left-hand panel of Figure 8.14, using the tree illustrated in the right-hand panel of the same figure. You should divide up the predictor space into the correct regions, and indicate the mean for each region. The diagram is as follows.



Question 2: ISLP § 8.4.9

Only a subset of these problems are to be answered. An additional subproblem has been added to this question by the instructor.

This problem involves the `OJ` data set which is part of the `ISLP` package. First, I will be loading the data set as downloaded from the course website.

[2]:
```
oj=pd.read_csv('../data/OJ.csv').rename({'Unnamed: 0': ''}, axis=1).set_index('')
oj.head()
```

[2]:

	Purchase	WeekofPurchase	StoreID	PriceCH	PriceMM	DiscCH	DiscMM	\
1	CH	237	1	1.75	1.99	0.00	0.0	
2	CH	239	1	1.75	1.99	0.00	0.3	
3	CH	245	1	1.86	2.09	0.17	0.0	
4	MM	227	1	1.69	1.69	0.00	0.0	
5	CH	228	7	1.69	1.69	0.00	0.0	

	SpecialCH	SpecialMM	LoyalCH	SalePriceMM	SalePriceCH	PriceDiff	Store7	\
1	0	0	0.500000	1.99	1.75	0.24	No	
2	0	1	0.600000	1.69	1.75	-0.06	No	
3	0	0	0.680000	2.09	1.69	0.40	No	
4	0	0	0.400000	1.69	1.69	0.00	No	
5	0	0	0.956535	1.69	1.69	0.00	Yes	

	PctDiscMM	PctDiscCH	ListPriceDiff	STORE
1	0.000000	0.000000	0.24	1
2	0.150754	0.000000	0.24	1
3	0.000000	0.091398	0.23	1
4	0.000000	0.000000	0.00	1
5	0.000000	0.000000	0.00	0

(a) Create a training set containing a random sample of 800 observations, and a test set containing the remaining observations. I will use the `train_test_split()` method from `sklearn.model_selection`, with the random seed state set at `381`.

[3]:
```
oj_train, oj_test = train_test_split(oj, train_size=800, random_state=381)
```

(b) Fit a tree to the training data, with `Purchase` as the response and the other variables as predictors. What is the training error rate? First, I will split the training and testing data to predictor-response format. I will also clean the data and encode the qualitative variables that are in the form of a string.

[4]:
```
X_train = pd.get_dummies(oj_train.drop('Purchase', axis=1), drop_first=True)
X_test = pd.get_dummies(oj_test.drop('Purchase', axis=1), drop_first=True)
```

```
y_train = oj_train['Purchase']
y_test = oj_test['Purchase']
```

We can now train a classification decision tree to the training data.

[5]:
```
clf = DecisionTreeClassifier(random_state=381)
clf.fit(X_train,y_train)
```

[5]:
```
DecisionTreeClassifier(random_state=381)
```

We can now calculate the training error.

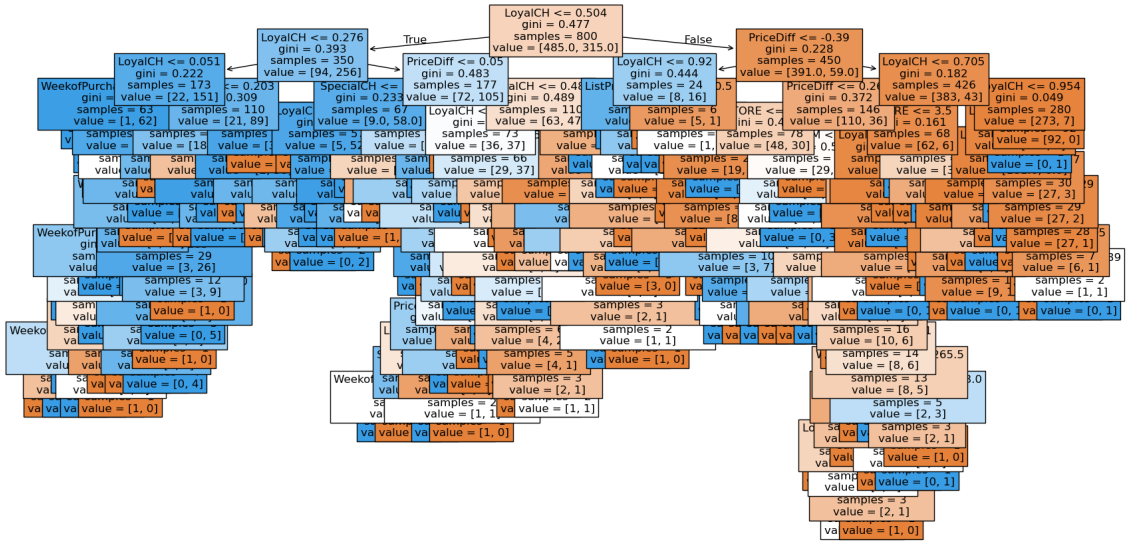
[6]:
```
(1-accuracy_score(y_train, clf.predict(X_train)))
```

[6]:
```
0.007499999999999951
```

The error rate is very low, at less than 0.75%. The tree is most likely overfit, so I expect the tree to be pretty bushy, and with significant depth, since I haven't constrained the maximum depth.

(c) Create a plot of the tree, and interpret the results. How many terminal nodes does the tree have? We can now use the `plot_tree()` method in `sklearn.tree` to to visualize this tree.

[7]:
```
fig = plt.figure(figsize = (20,10))
_=tree.plot_tree(clf, feature_names = X_train.columns, filled = True, fontsize =␣
↪12)
plt.show()
```



The plot agrees with my expectation that this will be a bushy tree. Since I cannot possibly count the number of terminal nodes, or leaves, I will query the tree object (`clf`) using `tree_.n_leaves` as demonstrated in ISLP § 8.3.

[8]:
```
clf.tree_.n_leaves
```

[8]: 151

There are 151 terminal nodes in the tree. This leads me to conclude that the tree generated is hardly interpretable.

(d) Use the `export_text()` function to produce a text summary of the fitted tree. Pick one of the terminal nodes, and interpret the information displayed.

[9]:
```
print(tree.export_text(clf, feature_names = list(X_train.columns)))
```

```
|--- LoyalCH <= 0.50
|   |--- LoyalCH <= 0.28
|   |   |--- LoyalCH <= 0.05
|   |   |   |--- WeekofPurchase <= 268.50
|   |   |   |   |--- class: MM
|   |   |   |--- WeekofPurchase >  268.50
|   |   |   |   |--- PriceDiff <= 0.26
|   |   |   |   |   |--- class: MM
|   |   |   |   |--- PriceDiff >  0.26
|   |   |   |   |   |--- LoyalCH <= 0.00
|   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |--- LoyalCH >  0.00
|   |   |   |   |   |   |--- class: CH
|   |   |--- LoyalCH >  0.05
|   |   |   |--- LoyalCH <= 0.20
|   |   |   |   |--- LoyalCH <= 0.18
|   |   |   |   |   |--- STORE <= 3.50
|   |   |   |   |   |   |--- WeekofPurchase <= 273.50
|   |   |   |   |   |   |   |--- WeekofPurchase <= 264.00
|   |   |   |   |   |   |   |   |--- STORE <= 1.50
|   |   |   |   |   |   |   |   |   |--- WeekofPurchase <= 243.00
|   |   |   |   |   |   |   |   |   |   |--- LoyalCH <= 0.07
|   |   |   |   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |   |   |   |   |   |--- LoyalCH >  0.07
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 5
|   |   |   |   |   |   |   |   |   |--- WeekofPurchase >  243.00
|   |   |   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |   |   |   |--- STORE >  1.50
|   |   |   |   |   |   |   |   |   |--- WeekofPurchase <= 236.50
|   |   |   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |   |   |   |   |--- WeekofPurchase >  236.50
|   |   |   |   |   |   |   |   |   |   |--- PctDiscCH <= 0.03
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 4
|   |   |   |   |   |   |   |   |   |   |--- PctDiscCH >  0.03
|   |   |   |   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |   |   |   |--- WeekofPurchase >  264.00
|   |   |   |   |   |   |   |   |   |--- class: CH
```

```
|   |   |   |   |   |   |   |--- WeekofPurchase >  273.50
|   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |   |--- STORE >  3.50
|   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |--- LoyalCH >  0.18
|   |   |   |   |   |   |--- PriceDiff <= 0.27
|   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |   |--- PriceDiff >  0.27
|   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |--- LoyalCH >  0.20
|   |   |   |   |   |--- PriceDiff <= 0.54
|   |   |   |   |   |   |--- LoyalCH <= 0.26
|   |   |   |   |   |   |   |--- WeekofPurchase <= 228.50
|   |   |   |   |   |   |   |   |--- STORE <= 1.00
|   |   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |   |   |   |--- STORE >  1.00
|   |   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |   |   |--- WeekofPurchase >  228.50
|   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |   |--- LoyalCH >  0.26
|   |   |   |   |   |   |   |--- LoyalCH <= 0.27
|   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |   |   |--- LoyalCH >  0.27
|   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |--- PriceDiff >  0.54
|   |   |   |   |   |   |--- class: CH
|   |--- LoyalCH >  0.28
|   |   |--- PriceDiff <= 0.05
|   |   |   |--- SpecialCH <= 0.50
|   |   |   |   |--- LoyalCH <= 0.28
|   |   |   |   |   |--- class: CH
|   |   |   |   |--- LoyalCH >  0.28
|   |   |   |   |   |--- STORE <= 0.50
|   |   |   |   |   |   |--- ListPriceDiff <= 0.07
|   |   |   |   |   |   |   |--- LoyalCH <= 0.45
|   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |   |   |--- LoyalCH >  0.45
|   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |   |--- ListPriceDiff >  0.07
|   |   |   |   |   |   |   |--- LoyalCH <= 0.33
|   |   |   |   |   |   |   |   |--- PriceDiff <= -0.06
|   |   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |   |   |   |--- PriceDiff >  -0.06
|   |   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |   |   |--- LoyalCH >  0.33
|   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |--- STORE >  0.50
|   |   |   |   |   |   |--- ListPriceDiff <= 0.31
```

```
|   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |   |   |--- ListPriceDiff >  0.31
|   |   |   |   |   |   |   |   |--- LoyalCH <= 0.47
|   |   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |   |   |   |--- LoyalCH >  0.47
|   |   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |--- SpecialCH >  0.50
|   |   |   |   |--- LoyalCH <= 0.45
|   |   |   |   |   |--- LoyalCH <= 0.38
|   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |--- LoyalCH >  0.38
|   |   |   |   |   |   |--- PriceCH <= 1.87
|   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |   |--- PriceCH >  1.87
|   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |--- LoyalCH >  0.45
|   |   |   |   |   |--- STORE <= 1.00
|   |   |   |   |   |   |--- WeekofPurchase <= 233.50
|   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |   |--- WeekofPurchase >  233.50
|   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |--- STORE >  1.00
|   |   |   |   |   |   |--- class: MM
|   |   |--- PriceDiff >  0.05
|   |   |   |--- LoyalCH <= 0.48
|   |   |   |   |--- LoyalCH <= 0.32
|   |   |   |   |   |--- class: CH
|   |   |   |   |--- LoyalCH >  0.32
|   |   |   |   |   |--- PriceDiff <= 0.33
|   |   |   |   |   |   |--- LoyalCH <= 0.46
|   |   |   |   |   |   |   |--- WeekofPurchase <= 234.00
|   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |   |   |--- WeekofPurchase >  234.00
|   |   |   |   |   |   |   |   |--- SalePriceCH <= 1.82
|   |   |   |   |   |   |   |   |   |--- LoyalCH <= 0.45
|   |   |   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |   |   |   |   |--- LoyalCH >  0.45
|   |   |   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |   |   |   |--- SalePriceCH >  1.82
|   |   |   |   |   |   |   |   |   |--- LoyalCH <= 0.45
|   |   |   |   |   |   |   |   |   |   |--- LoyalCH <= 0.42
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 6
|   |   |   |   |   |   |   |   |   |--- LoyalCH >  0.42
|   |   |   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |   |   |   |--- LoyalCH >  0.45
|   |   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |   |   |--- LoyalCH >  0.46
|   |   |   |   |   |   |   |   |--- class: MM
```

```
|   |   |   |   |   |--- PriceDiff >  0.33
|   |   |   |   |   |   |--- LoyalCH <= 0.33
|   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |   |--- LoyalCH >  0.33
|   |   |   |   |   |   |   |--- LoyalCH <= 0.47
|   |   |   |   |   |   |   |   |--- WeekofPurchase <= 245.00
|   |   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |   |   |   |--- WeekofPurchase >  245.00
|   |   |   |   |   |   |   |   |   |--- WeekofPurchase <= 252.00
|   |   |   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |   |   |   |   |--- WeekofPurchase >  252.00
|   |   |   |   |   |   |   |   |   |   |--- LoyalCH <= 0.41
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 5
|   |   |   |   |   |   |   |   |   |   |--- LoyalCH >  0.41
|   |   |   |   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |   |   |   |--- LoyalCH >  0.47
|   |   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |--- LoyalCH >  0.48
|   |   |   |   |   |--- ListPriceDiff <= 0.38
|   |   |   |   |   |   |--- PriceCH <= 1.88
|   |   |   |   |   |   |   |--- SalePriceMM <= 2.04
|   |   |   |   |   |   |   |   |--- ListPriceDiff <= 0.29
|   |   |   |   |   |   |   |   |   |--- PriceCH <= 1.81
|   |   |   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |   |   |   |   |--- PriceCH >  1.81
|   |   |   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |   |   |   |--- ListPriceDiff >  0.29
|   |   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |   |   |--- SalePriceMM >  2.04
|   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |   |--- PriceCH >  1.88
|   |   |   |   |   |   |   |--- PriceCH <= 1.94
|   |   |   |   |   |   |   |   |--- WeekofPurchase <= 245.50
|   |   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |   |   |   |--- WeekofPurchase >  245.50
|   |   |   |   |   |   |   |   |   |--- WeekofPurchase <= 251.00
|   |   |   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |   |   |   |   |--- WeekofPurchase >  251.00
|   |   |   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |   |   |--- PriceCH >  1.94
|   |   |   |   |   |   |   |   |--- WeekofPurchase <= 253.00
|   |   |   |   |   |   |   |   |   |--- WeekofPurchase <= 251.50
|   |   |   |   |   |   |   |   |   |   |--- StoreID <= 3.50
|   |   |   |   |   |   |   |   |   |   |   |--- WeekofPurchase <= 244.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |   |   |   |   |   |   |--- WeekofPurchase >  244.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |   |--- StoreID >  3.50
```

```
|   |   |   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |   |   |   |   |--- WeekofPurchase >  251.50
|   |   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |   |   |   |--- WeekofPurchase >  253.00
|   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |--- ListPriceDiff >  0.38
|   |   |   |   |   |--- class: MM
|--- LoyalCH >  0.50
|   |--- PriceDiff <= -0.39
|   |   |--- LoyalCH <= 0.92
|   |   |   |--- ListPriceDiff <= 0.16
|   |   |   |   |--- class: MM
|   |   |   |--- ListPriceDiff >  0.16
|   |   |   |   |--- LoyalCH <= 0.71
|   |   |   |   |   |--- LoyalCH <= 0.63
|   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |--- LoyalCH >  0.63
|   |   |   |   |   |   |--- class: CH
|   |   |   |   |--- LoyalCH >  0.71
|   |   |   |   |   |--- class: MM
|   |   |--- LoyalCH >  0.92
|   |   |   |--- SpecialCH <= 0.50
|   |   |   |   |--- class: CH
|   |   |   |--- SpecialCH >  0.50
|   |   |   |   |--- LoyalCH <= 1.00
|   |   |   |   |   |--- class: MM
|   |   |   |   |--- LoyalCH >  1.00
|   |   |   |   |   |--- class: CH
|   |--- PriceDiff >  -0.39
|   |   |--- LoyalCH <= 0.71
|   |   |   |--- PriceDiff <= 0.27
|   |   |   |   |--- STORE <= 0.50
|   |   |   |   |   |--- LoyalCH <= 0.69
|   |   |   |   |   |   |--- SpecialCH <= 0.50
|   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |   |--- SpecialCH >  0.50
|   |   |   |   |   |   |   |--- PriceDiff <= -0.13
|   |   |   |   |   |   |   |   |--- WeekofPurchase <= 233.50
|   |   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |   |   |   |--- WeekofPurchase >  233.50
|   |   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |   |   |--- PriceDiff >  -0.13
|   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |--- LoyalCH >  0.69
|   |   |   |   |   |   |--- class: MM
|   |   |   |   |--- STORE >  0.50
|   |   |   |   |   |--- SalePriceMM <= 2.15
|   |   |   |   |   |   |--- PctDiscMM <= 0.20
```

```
|   |   |   |   |   |   |   |   |--- LoyalCH <= 0.69
|   |   |   |   |   |   |   |   |   |--- WeekofPurchase <= 236.50
|   |   |   |   |   |   |   |   |   |   |--- WeekofPurchase <= 228.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |   |   |   |   |   |--- WeekofPurchase >  228.50
|   |   |   |   |   |   |   |   |   |   |   |--- STORE <= 3.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |   |   |   |   |   |   |--- STORE >  3.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |--- WeekofPurchase >  236.50
|   |   |   |   |   |   |   |   |   |   |--- PriceDiff <= -0.06
|   |   |   |   |   |   |   |   |   |   |   |--- SalePriceCH <= 1.94
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |   |--- SalePriceCH >  1.94
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |--- PriceDiff >  -0.06
|   |   |   |   |   |   |   |   |   |   |--- StoreID <= 1.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |   |   |   |   |   |--- StoreID >  1.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 10
|   |   |   |   |   |   |   |   |--- LoyalCH >  0.69
|   |   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |   |   |--- PctDiscMM >  0.20
|   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |   |--- SalePriceMM >  2.15
|   |   |   |   |   |   |   |--- class: CH
|   |   |   |--- PriceDiff >  0.27
|   |   |   |   |--- STORE <= 3.50
|   |   |   |   |   |--- LoyalCH <= 0.54
|   |   |   |   |   |   |--- StoreID <= 2.50
|   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |   |--- StoreID >  2.50
|   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |--- LoyalCH >  0.54
|   |   |   |   |   |   |--- WeekofPurchase <= 251.50
|   |   |   |   |   |   |   |--- WeekofPurchase <= 248.00
|   |   |   |   |   |   |   |   |--- LoyalCH <= 0.68
|   |   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |   |   |   |--- LoyalCH >  0.68
|   |   |   |   |   |   |   |   |   |--- WeekofPurchase <= 235.50
|   |   |   |   |   |   |   |   |   |   |--- WeekofPurchase <= 230.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |   |   |   |   |   |--- WeekofPurchase >  230.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |   |   |   |   |--- WeekofPurchase >  235.50
|   |   |   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |   |   |   |--- WeekofPurchase >  248.00
|   |   |   |   |   |   |   |   |   |--- class: MM
```

```
|   |   |   |   |   |   |   |--- WeekofPurchase >  251.50
|   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |--- STORE >  3.50
|   |   |   |   |   |   |--- LoyalCH <= 0.62
|   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |   |--- LoyalCH >  0.62
|   |   |   |   |   |   |   |--- LoyalCH <= 0.68
|   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |   |   |--- LoyalCH >  0.68
|   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |--- LoyalCH >  0.71
|   |   |   |   |--- LoyalCH <= 0.95
|   |   |   |   |   |--- LoyalCH <= 0.95
|   |   |   |   |   |   |--- LoyalCH <= 0.93
|   |   |   |   |   |   |   |--- SalePriceMM <= 2.26
|   |   |   |   |   |   |   |   |--- LoyalCH <= 0.75
|   |   |   |   |   |   |   |   |   |--- LoyalCH <= 0.74
|   |   |   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |   |   |   |   |--- LoyalCH >  0.74
|   |   |   |   |   |   |   |   |   |   |--- LoyalCH <= 0.74
|   |   |   |   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |   |   |   |   |   |--- LoyalCH >  0.74
|   |   |   |   |   |   |   |   |   |   |   |--- SpecialMM <= 0.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |   |   |   |   |   |   |--- SpecialMM >  0.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |   |   |   |--- LoyalCH >  0.75
|   |   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |   |   |--- SalePriceMM >  2.26
|   |   |   |   |   |   |   |   |--- LoyalCH <= 0.81
|   |   |   |   |   |   |   |   |   |--- WeekofPurchase <= 257.00
|   |   |   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |   |   |   |   |--- WeekofPurchase >  257.00
|   |   |   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |   |   |   |--- LoyalCH >  0.81
|   |   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |   |--- LoyalCH >  0.93
|   |   |   |   |   |   |   |--- PriceCH <= 1.77
|   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |   |   |--- PriceCH >  1.77
|   |   |   |   |   |   |   |   |--- LoyalCH <= 0.93
|   |   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |   |   |   |   |--- LoyalCH >  0.93
|   |   |   |   |   |   |   |   |   |--- STORE <= 3.50
|   |   |   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |   |   |   |   |--- STORE >  3.50
|   |   |   |   |   |   |   |   |   |   |--- DiscCH <= 0.05
|   |   |   |   |   |   |   |   |   |   |   |--- class: CH
```

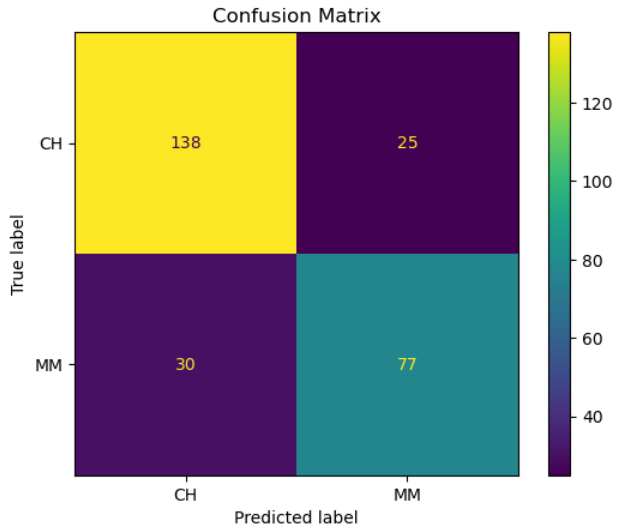
```
|   |   |   |   |   |   |   |   |   |--- DiscCH >  0.05
|   |   |   |   |   |   |   |   |   |   |--- LoyalCH <= 0.94
|   |   |   |   |   |   |   |   |   |   |   |--- class: CH
|   |   |   |   |   |   |   |   |   |   |--- LoyalCH >  0.94
|   |   |   |   |   |   |   |   |   |   |   |--- class: MM
|   |   |   |   |--- LoyalCH >  0.95
|   |   |   |   |   |--- class: MM
|   |   |   |--- LoyalCH >  0.95
|   |   |   |   |--- class: CH
```

I am choosing to interpret the first terminal node presented in the previous output.

The node was classified as Minute Maid (MM). The node can be interpreted as any customer who makes a purchase less than 268.5 weeks from the start of the data collection that has lesser than 5% of brand loyalty to Citrus Hill (with brand loyalty to Citrus Hill being split 3 distinct times at 50%, 28% and 5%) most likely purchases a Minute Maid orange juice bottle.

(e) Predict the response on the test data, and produce a confusion matrix comparing the test labels to the predicted test labels. What is the test error rate?

[10]:
```
ConfusionMatrixDisplay(confusion_matrix=confusion_matrix(clf.
 ↪predict(X_test),y_test),
                       display_labels=['CH', 'MM']).plot()
plt.title('Confusion Matrix')
plt.show()
```



[11]:
```
(1-accuracy_score(y_test, clf.predict(X_test)))
```

[11]: 0.20370370370370372

The test error rate is a bit more than 21%, which is a massive increase from the training error rate.

(f) Use cross-validation on the training set in order to determine the optimal tree size.
I first find the maximum tree depth of the current tree so I can create a range of candidate tree depth/size values.

[12]:
```
clf.tree_.max_depth
```

[12]: 20

I can now conduct a 10-fold cross-validation accordingly to estimate the optimal tree size. We will be scoring the accuracy of the different depths.

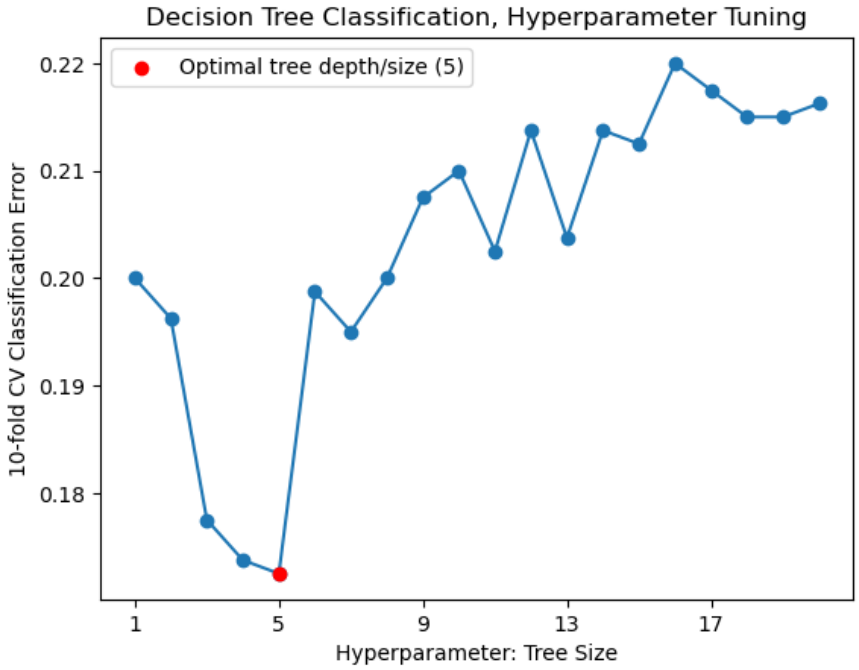
[13]:
```
kf=KFold(n_splits=10, shuffle=True, random_state=381)
M=np.zeros(clf.tree_.max_depth)
for depth in range(1,len(M)+1):
    clf_cv=DecisionTreeClassifier(max_depth=depth, random_state=381)
    scores = cross_val_score(clf_cv, X_train, y_train, cv=kf)
    M[depth-1]=np.mean(scores)
depth_opt=np.argmax(M)+1
acc_opt=max(M)
print(depth_opt, acc_opt, sep=', ')
```

```
5, 0.8275
```

It looks like a maximum depth of 5 provides the highest cross-validation accuracy at 82.75% (lowest error rate at 17.25%).

(g) Produce a plot with tree size on the x-axis and cross-validated classification error rate on the y-axis.

[14]:
```
plt.plot(range(1,clf.tree_.max_depth+1),1-M, '-o')
plt.title('Decision Tree Classification, Hyperparameter Tuning')
plt.scatter(depth_opt, 1-acc_opt, c='red', zorder=2, label=f'Optimal tree depth/
 ↪size ({depth_opt})')
plt.xlabel('Hyperparameter: Tree Size')
plt.ylabel('10-fold CV Classification Error')
plt.legend()
plt.xticks(list(range(1,clf.tree_.max_depth+1,4)))
plt.show()
```



(h) Which tree size corresponds to the lowest cross-validated classification error rate? As determined in (f), the optimal tree size is 5, since it corresponds to the lowest cross-validated error rate of 17.25%.

(i) [Modified] Build a random forest model for the same classification task. Using the techniques from class, what choice of `n_estimators` and `max_features` would you choose for your model? I will be conducting a 10-fold cross-validation in a limited parameter grid that I will be creating. The number of estimators will be selected from a range of values, while the candidate values for the maximum number of features in terms of the total number of predictors p will be $\sqrt{p}$, $\frac{p}{3}$, $\frac{p}{2}$, and p. Ideally, I would use quarter the number of predictors too, i.e., $\frac{p}{4}$, but the integer type conversions of $\frac{p}{4}$ and $\sqrt{p}$ are the same.

[15]:
```
p=oj.shape[1]
param_grid = {
    'n_estimators': np.arange(10,501,10),
    'max_features': [int(np.sqrt(p)), int(p/3), int(p/2), p]
    # I initially used 'sqrt', but that complicated labeling in the plot.
}
```

I will now conduct the cross-validation. I will be doing a grid search this time.

[16]:
```
rf_clf = RandomForestClassifier(random_state=381, oob_score = True, max_depth=p)
grid_search = GridSearchCV(
    estimator=rf_clf,
    param_grid=param_grid,
    scoring='accuracy',
    cv=10
)
```

With that done, I will now proceed to train the classifier via the grid search to identify the optimal parameters. Cross-validating takes an immense amount of time with the number of parameters we are testing out.

[17]:
```
grid_search.fit(X_train, y_train)
```

[17]:
```
GridSearchCV(cv=10,
             estimator=RandomForestClassifier(max_depth=18, oob_score=True,
                                              random_state=381),
             param_grid={'max_features': [4, 6, 9, 18],
                         'n_estimators': array([ 10,  20,  30,  40,  50,  60,
70,  80,  90, 100, 110, 120, 130,
       140, 150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250, 260,
       270, 280, 290, 300, 310, 320, 330, 340, 350, 360, 370, 380, 390,
       400, 410, 420, 430, 440, 450, 460, 470, 480, 490, 500])},
             scoring='accuracy')
```

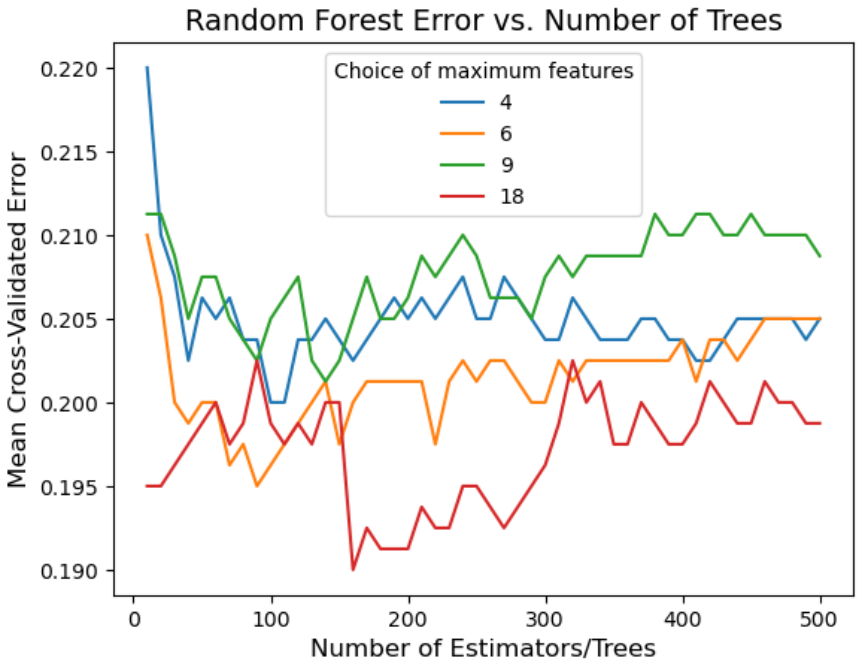
The optimal parameters are as follows:

[18]:
```
opt_n = grid_search.best_params_['n_estimators']
opt_m = grid_search.best_params_['max_features']
opt_acc = grid_search.best_score_
print(opt_n, opt_m, 1-opt_acc, sep=', ')
```

```
160, 18, 0.19000000000000006
```

We can conclude that the best parameters to conduct a Random Forest classification of this data would be 160 trees, using all 18 features, giving us an error of 19%. But since it is not advised to use all features, we can plot the error rates to find other optimal parameters.

[19]:
```
results = pd.DataFrame(grid_search.cv_results_).loc[:, ['param_n_estimators',␣
↪'param_max_features', 'mean_test_score']]
results.columns = ['n_estimators', 'max_features', 'mean_accuracy']
for m in results['max_features'].unique():
    subset = results[results['max_features'] == m]
    plt.plot(subset['n_estimators'], 1-subset['mean_accuracy'],
             label=m)
plt.title('Random Forest Error vs. Number of Trees', fontsize=14)
plt.xlabel('Number of Estimators/Trees', fontsize=12)
plt.ylabel('Mean Cross-Validated Error', fontsize=12)
plt.legend(title='Choice of maximum features')
plt.show()
```



Disregarding the values for the full data set, it looks like an alternate value would be 40 trees when using half the predictors.

If I had a large amount of computing power, I would set `n_estimators=160` and `max_features=18`. If I had limited computing power, I would set `n_estimators=40` and `max_features=0.5` or `max_features=9`.

Acknowledgements

I adopted the `GridSearchCV()` method from `sklearn.model_selection` to conduct a grid search of the candidate optimal hyperparameters value instead of a regular nested loop structure.