Question 1

Part A:

DecisionTreeClassifier

---------------------------------------------------------------------------------------------------------

Dataset | 5% | 10% | 15% | 20% | 25% | 30% | 35% | 40% | 45% | 50% |

---------------------------------------------------------------------------------------------------------

australian | 72.61% | 74.63% | 75.52% | 77.53% | 77.97% | 79.86% | 83.05% | 81.29% | 80.14% | 82.91% |

balance-scale | 70.10% | 72.47% | 71.20% | 75.69% | 73.77% | 75.67% | 77.74% | 75.99% | 78.09% | 76.98% |

hypothyroid | 94.94% | 96.31% | 97.77% | 99.18% | 99.21% | 99.42% | 99.42% | 99.52% | 99.34% | 99.20% |

BernoulliNB with priors

---------------------------------------------------------------------------------------------------------

Dataset | 5% | 10% | 15% | 20% | 25% | 30% | 35% | 40% | 45% | 50% |

---------------------------------------------------------------------------------------------------------

australian | 73.47% | 79.85% | 81.72% | 80.43% | 79.69% | 79.84% | 80.12% | 81.14% | 82.16% | 81.28% |

balance-scale | 46.08% | 46.08% | 46.08% | 46.08% | 46.08% | 46.08% | 46.08% | 46.08% | 46.08% | 46.08% |

hypothyroid | 91.38% | 91.81% | 92.23% | 92.23% | 92.23% | 92.26% | 92.23% | 92.23% | 92.23% | 92.23% |

Part B:

True statements:

--- (3) most of the 6 models show a learning curve

--- (4) All 3 Decision Tree models are generally better than Bernoulli Naive Bayes models

Part C:

BernoulliNB with priors

---------------------------------------------------------------------------------------------------------

Dataset | 5% | 10% | 15% | 20% | 25% | 30% | 35% | 40% | 45% | 50% |

---------------------------------------------------------------------------------------------------------

australian | 73.47% | 79.85% | 81.72% | 80.43% | 79.69% | 79.84% | 80.12% | 81.14% | 82.16% | 81.28% |

balance-scale | 46.08% | 46.08% | 46.08% | 46.08% | 46.08% | 46.08% | 46.08% | 46.08% | 46.08% | 46.08% |

hypothyroid | 91.38% | 91.81% | 92.23% | 92.23% | 92.23% | 92.26% | 92.23% | 92.23% | 92.23% | 92.23% |

BernoulliNB with uniform priors

---------------------------------------------------------------------------------------------------------

Dataset | 5% | 10% | 15% | 20% | 25% | 30% | 35% | 40% | 45% | 50% |

---------------------------------------------------------------------------------------------------------

australian | 73.62% | 79.27% | 81.44% | 78.98% | 78.40% | 79.69% | 78.52% | 79.83% | 80.41% | 80.41% |

balance-scale | 46.08% | 46.08% | 46.08% | 46.08% | 46.08% | 46.08% | 46.08% | 46.08% | 46.08% | 46.08% |

hypothyroid | 83.88% | 79.59% | 77.44% | 74.79% | 73.12% | 65.05% | 53.60% | 51.30% | 51.09% | 50.26% |

According to the sheet, BNB preforms better with priors.

Question 2

Part A:

accuracy score for training dataset: 0.8564516129032258

accuracy score for test dataset: 0.8277153558052435

Part B:

The optimal number of min\_samples\_leaf is: 5

The AUC score is: 0.8877923976608187

Part C:

![手机屏幕截图

描述已自动生成]()

![手机屏幕截图

描述已自动生成]()

Part D:

P(S=true | G=female, C=1): 0.36885245901639346

The Code

1. **import** pandas as pd
2. **import** numpy as np
3. **from** sklearn **import** tree
4. **from** sklearn.metrics **import** roc\_auc\_score, roc\_curve, auc
5. **import** matplotlib.pyplot as plt

8. **def** normalize(data):
9. **return** (data - data.min()) / (data.max() - data.min())

12. **def** AUC\_score(X, Y, model):
13. \_prob = model.predict\_proba(X)[:,1]
14. **return** roc\_auc\_score(Y, \_prob)

17. **def** main():
18. # Read data and creating test and training sets
19. data = pd.read\_csv("titanic.csv")
20. data\_normalized = normalize(data.iloc[:, :])
21. training\_set = data\_normalized.iloc[:620, :]
22. test\_set = data\_normalized.iloc[620:, :]
23. training\_set\_x = training\_set.iloc[:, :-1].values
24. training\_set\_y = training\_set.iloc[:, -1:].values
25. test\_set\_x = test\_set.iloc[:, :-1].values
26. test\_set\_y = test\_set.iloc[:, -1:].values
28. # Part A
29. clf = tree.DecisionTreeClassifier()
30. clf = clf.fit(training\_set\_x, training\_set\_y)
31. **print**('Part A(accuracy score for training dataset):', clf.score(training\_set\_x, training\_set\_y))
32. **print**('Part A(accuracy score for test dataset):', clf.score(test\_set\_x, test\_set\_y))
34. # Part B
35. training\_set\_auc\_score = []
36. test\_set\_auc\_score = []
37. **for** i **in** range(2, 21):
38. clf = tree.DecisionTreeClassifier(min\_samples\_leaf=i, random\_state=1)
39. clf.fit(training\_set\_x, training\_set\_y)
41. training\_set\_auc\_score.append(AUC\_score(training\_set\_x, training\_set\_y, clf))
42. test\_set\_auc\_score.append(AUC\_score(test\_set\_x, test\_set\_y, clf))
43. **print**("\nPart B,The optimal number of min\_samples\_leaf is: ", test\_set\_auc\_score.index(max(test\_set\_auc\_score)) + 2)
44. **print**("Part B,The AUC score is: ", max(test\_set\_auc\_score))
46. # Part C
47. plt.bar(range(2, 21), training\_set\_auc\_score, 0.4, color="blue")
48. plt.ylim(0.8, 0.95)
49. plt.xticks(range(2, 21))
50. plt.xlabel("number of min\_samples\_leaf")
51. plt.ylabel("AUC score")
52. plt.title("AUC score for number of min\_samples\_leaf in training sets")
53. plt.show()
54. plt.bar(range(2, 21), test\_set\_auc\_score, 0.4, color="blue")
55. plt.ylim(0.8, 0.95)
56. plt.xticks(range(2, 21))
57. plt.xlabel("number of min\_samples\_leaf")
58. plt.ylabel("AUC score")
59. plt.title("AUC score for number of min\_samples\_leaf in test sets")
60. plt.show()
62. # Part D
63. survived, total = 0, 0
64. **for** index, row **in** data.iterrows():
65. **if** row['Pclass'] == 1 & row['Sex'] == 1:
66. total += 1
67. **if** row['Survived'] == 1:
68. survived += 1
69. **print**("\nPart D, P(S=true | G=female, C=1): ", survived / total)

72. **if** \_\_name\_\_ == '\_\_main\_\_':
73. main()