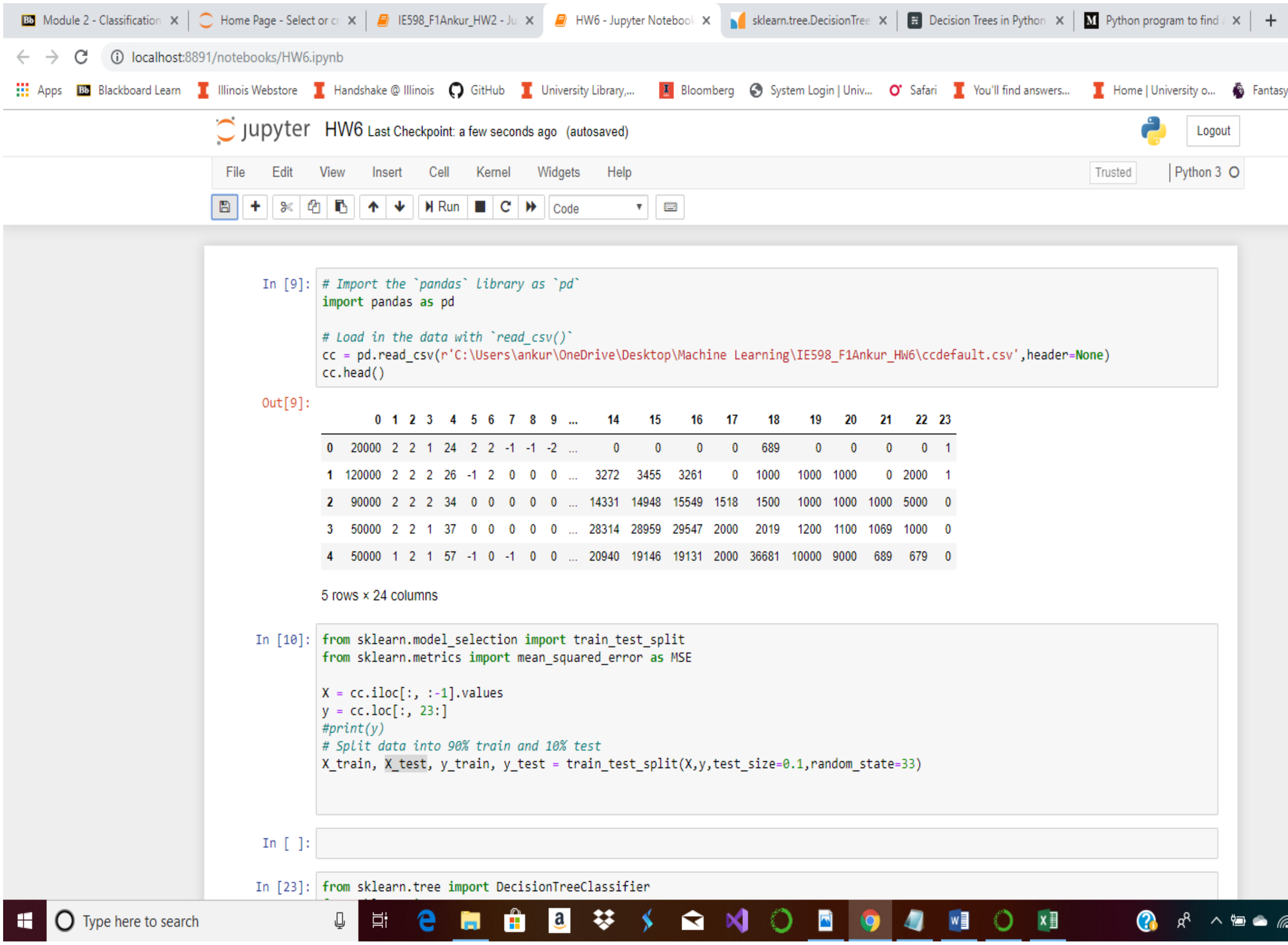
My Name (myNetID)

IE598 MLF F19

Module 6 Homework (Cross validation)

Part 1: Random test train splits



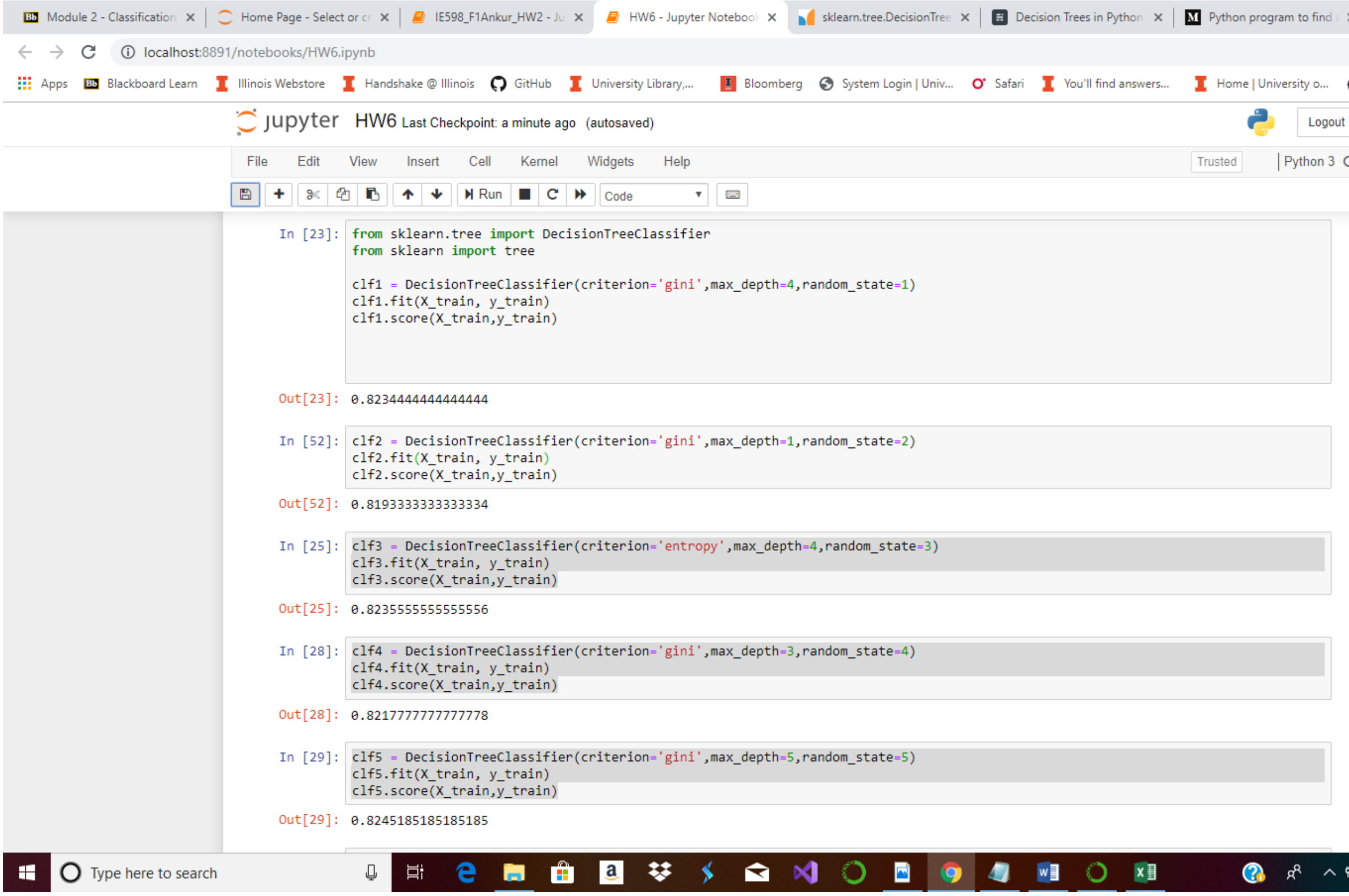
```
In [9]: # Import the `pandas` library as `pd`
        import pandas as pd

        # Load in the data with `read_csv()`
        cc = pd.read_csv(r'C:\Users\ankur\OneDrive\Desktop\Machine Learning\IE598_F1Ankur_HW6\ccdefault.csv',header=None)
        cc.head()
```

Out[9]:

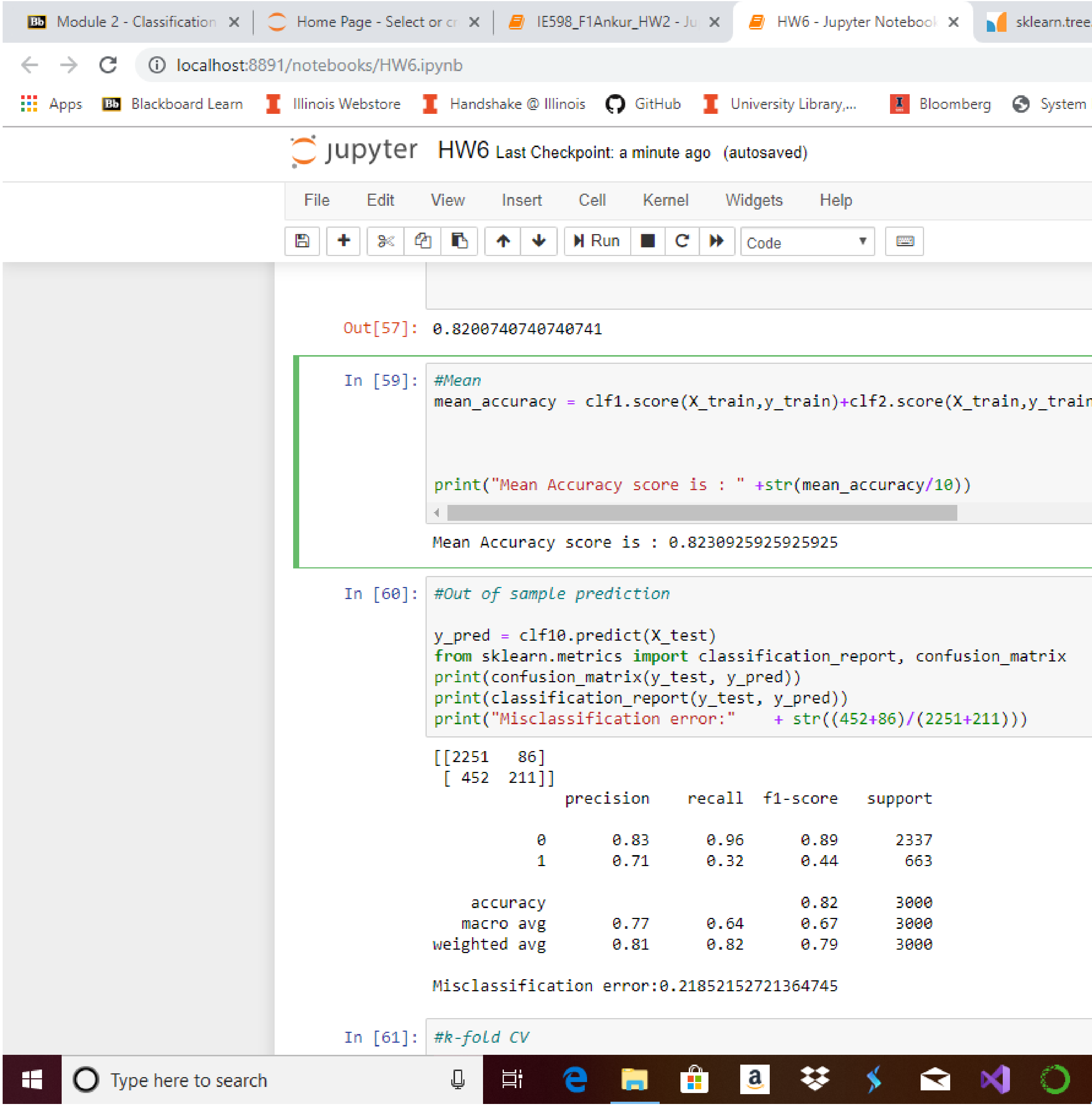
	0	1	2	3	4	5	6	7	8	9	...	14	15	16	17	18	19	20	21	22	23
0	20000	2	2	1	24	2	2	-1	-1	-2	...	0	0	0	0	689	0	0	0	0	1
1	120000	2	2	2	26	-1	2	0	0	0	...	3272	3455	3261	0	1000	1000	1000	0	2000	1
2	90000	2	2	2	34	0	0	0	0	0	...	14331	14948	15549	1518	1500	1000	1000	1000	5000	0
3	50000	2	2	1	37	0	0	0	0	0	...	28314	28959	29547	2000	2019	1200	1100	1069	1000	0
4	50000	1	2	1	57	-1	0	-1	0	0	...	20940	19146	19131	2000	36681	10000	9000	689	679	0

5 rows × 24 columns

```
In [10]: from sklearn.model_selection import train_test_split
         from sklearn.metrics import mean_squared_error as MSE

         X = cc.iloc[:, :-1].values
         y = cc.loc[:, 23:]
         #print(y)
         # Split data into 90% train and 10% test
         X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.1,random_state=33)

In [ ]:

In [23]: from sklearn.tree import DecisionTreeClassifier
```



```
In [23]: from sklearn.tree import DecisionTreeClassifier
         from sklearn import tree

         clf1 = DecisionTreeClassifier(criterion='gini',max_depth=4,random_state=1)
         clf1.fit(X_train, y_train)
         clf1.score(X_train,y_train)

Out[23]: 0.8234444444444444

In [52]: clf2 = DecisionTreeClassifier(criterion='gini',max_depth=1,random_state=2)
         clf2.fit(X_train, y_train)
         clf2.score(X_train,y_train)

Out[52]: 0.8193333333333334

In [25]: clf3 = DecisionTreeClassifier(criterion='entropy',max_depth=4,random_state=3)
         clf3.fit(X_train, y_train)
         clf3.score(X_train,y_train)

Out[25]: 0.8235555555555556

In [28]: clf4 = DecisionTreeClassifier(criterion='gini',max_depth=3,random_state=4)
         clf4.fit(X_train, y_train)
         clf4.score(X_train,y_train)

Out[28]: 0.8217777777777778

In [29]: clf5 = DecisionTreeClassifier(criterion='gini',max_depth=5,random_state=5)
         clf5.fit(X_train, y_train)
         clf5.score(X_train,y_train)

Out[29]: 0.8245185185185185
```



```
Out[57]: 0.8200740740740741
```

```
In [59]: #Mean
         mean_accuracy = clf1.score(X_train,y_train)+clf2.score(X_train,y_trair



         print("Mean Accuracy score is : " +str(mean_accuracy/10))
```

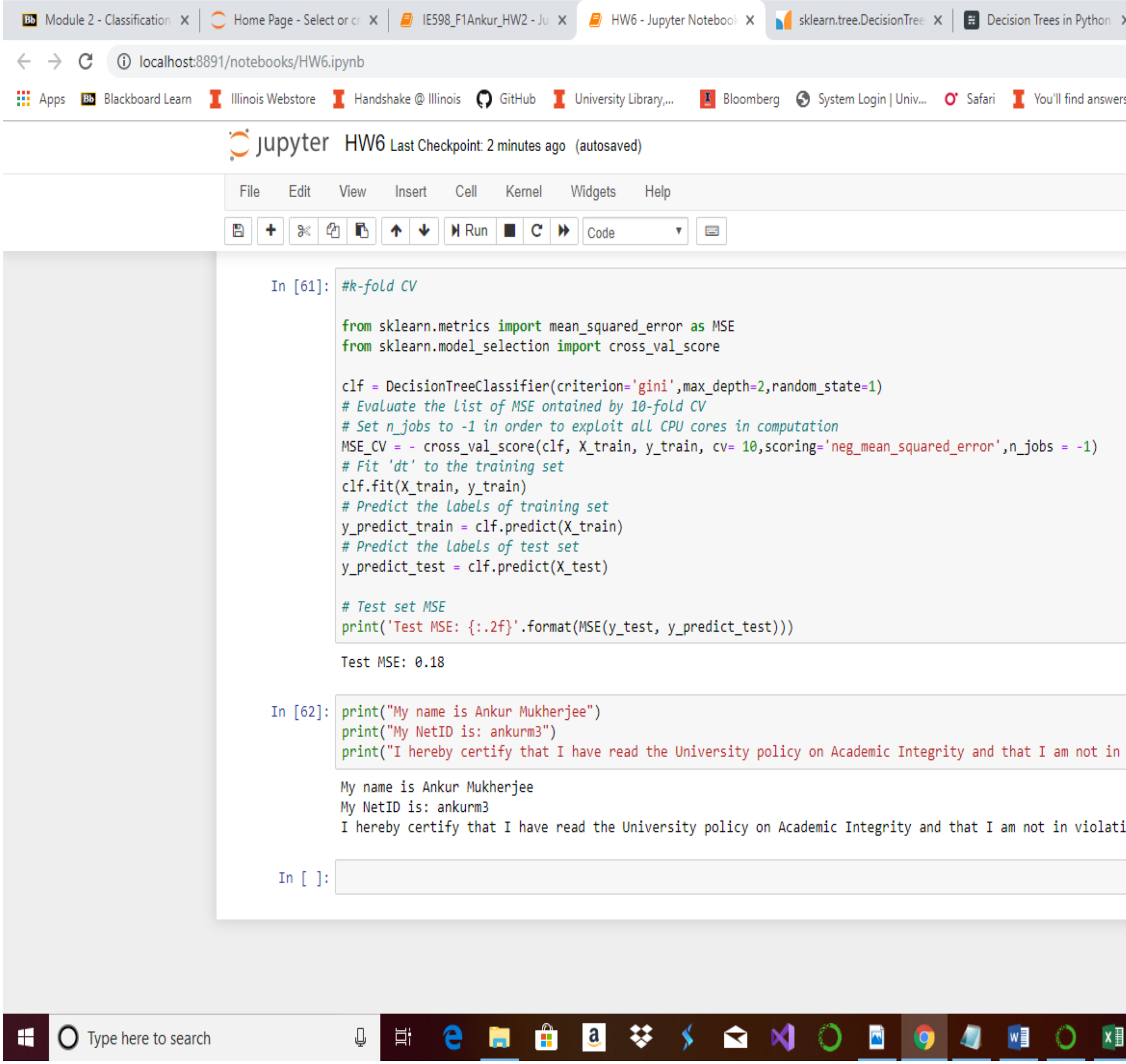
```
Mean Accuracy score is : 0.8230925925925925
```

```
In [60]: #Out of sample prediction

         y_pred = clf10.predict(X_test)
         from sklearn.metrics import classification_report, confusion_matrix
         print(confusion_matrix(y_test, y_pred))
         print(classification_report(y_test, y_pred))
         print("Misclassification error:"    + str((452+86)/(2251+211)))
```

```
[[2251   86]
 [ 452  211]]
              precision    recall  f1-score   support

           0       0.83      0.96      0.89      2337
           1       0.71      0.32      0.44       663

    accuracy                           0.82      3000
   macro avg       0.77      0.64      0.67      3000
weighted avg       0.81      0.82      0.79      3000

Misclassification error:0.21852152721364745
```

```
In [61]: #k-fold CV
```

Part 2: Cross validation



In [61]:

```
#k-fold CV

from sklearn.metrics import mean_squared_error as MSE
from sklearn.model_selection import cross_val_score

clf = DecisionTreeClassifier(criterion='gini',max_depth=2,random_state=1)
# Evaluate the list of MSE ontained by 10-fold CV
# Set n_jobs to -1 in order to exploit all CPU cores in computation
MSE_CV = - cross_val_score(clf, X_train, y_train, cv= 10,scoring='neg_mean_squared_error',n_jobs = -1)
# Fit 'dt' to the training set
clf.fit(X_train, y_train)
# Predict the labels of training set
y_predict_train = clf.predict(X_train)
# Predict the labels of test set
y_predict_test = clf.predict(X_test)

# Test set MSE
print('Test MSE: {:.2f}'.format(MSE(y_test, y_predict_test)))
```

```
Test MSE: 0.18
```

In [62]:

```
print("My name is Ankur Mukherjee")
print("My NetID is: ankurm3")
print("I hereby certify that I have read the University policy on Academic Integrity and that I am not in
```

```
My name is Ankur Mukherjee
My NetID is: ankurm3
I hereby certify that I have read the University policy on Academic Integrity and that I am not in violati
```

In [ ]:

Part 3: Conclusions

	Decision tree	Cross Val
Mean Squared Error	0.21	0.18

1) **The Cross Validation method produces better estimates by reducing the error as shown above.**
2) **This is because in the k-fold CV method, 10 errors are generated- E1, E2,E3…E10 by splitting the dataset in 10 partitions and then training the model. So, in successive steps error is reduced**

Part 4: Appendix

https://github.com/ankurmukherjeeuiuc/IE598_F1x_HW6