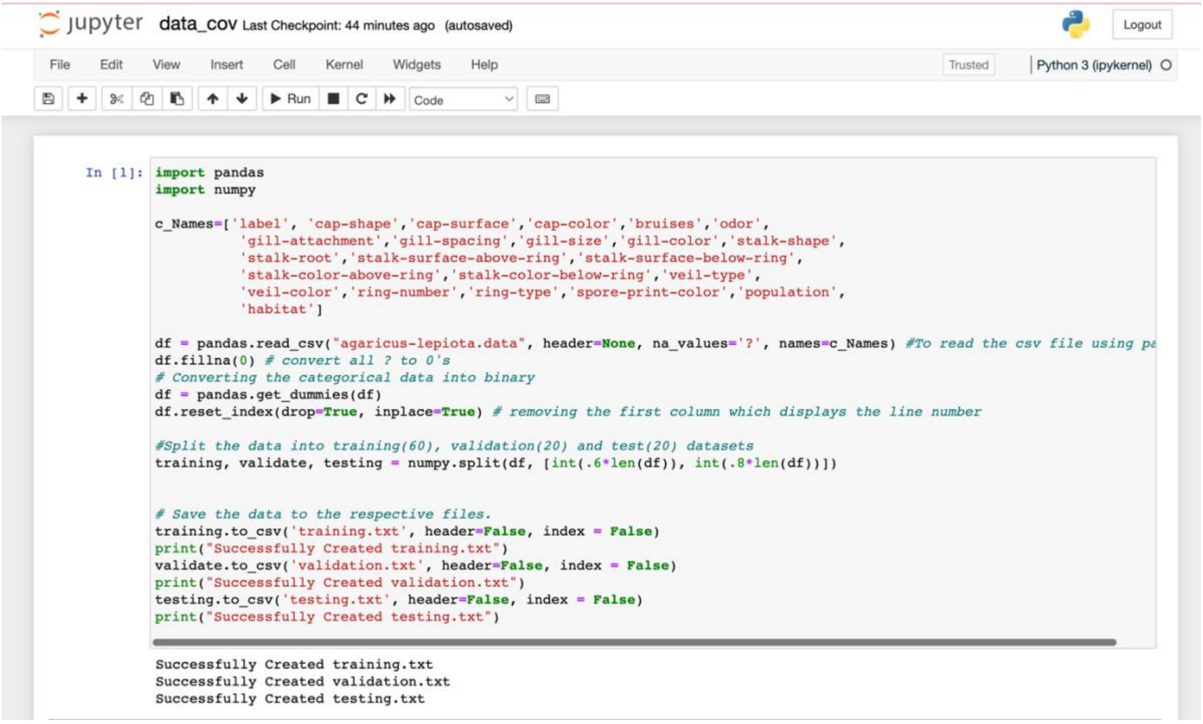
Final Project Report

Data Mining
-Tingjian Ge

Submitted by
-Shaili Trivedi(Shaili_Trivedi@student.uml.edu)
Student Id : 02008856

Abstract : We have been asked to create a simple neural network model to predict whether a particular mushroom that we have is poisonous or edible using agaricus-lepiota.data. The data given has in total 23 feature through which we can identify if the mushroom is edible or not.

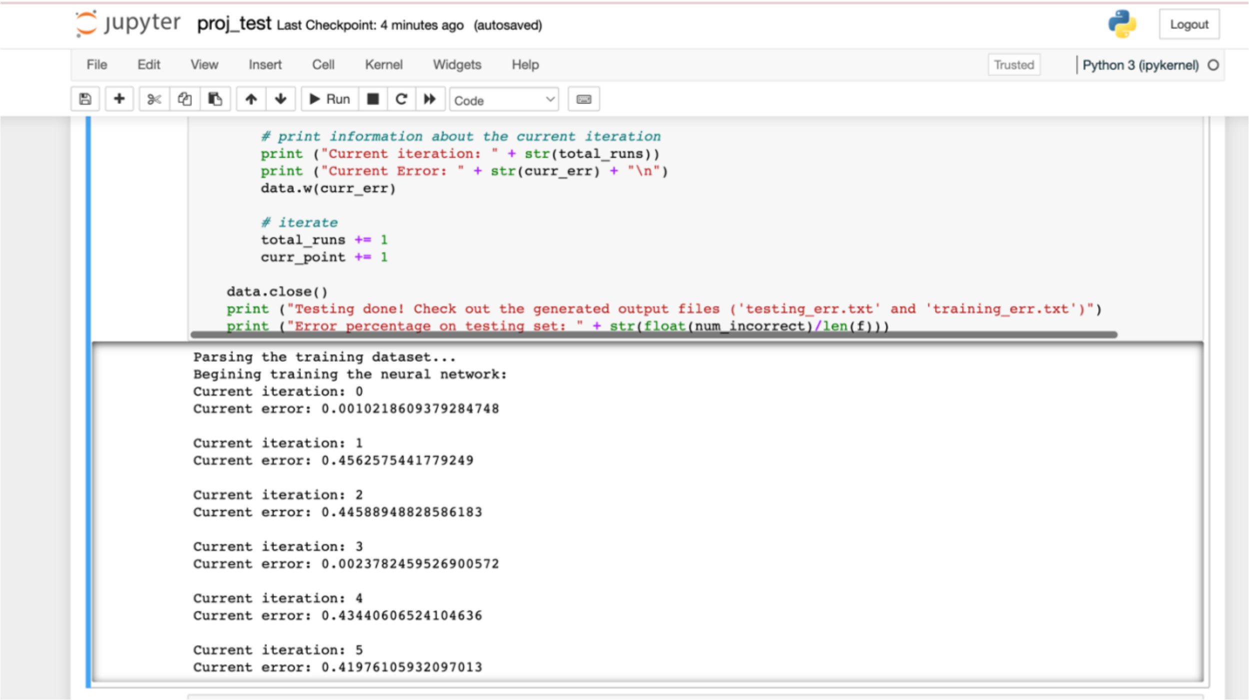
Screenshot of the report with some of the partial Running results



```
import pandas
import numpy

c_Names=['label', 'cap-shape','cap-surface','cap-color','bruises','odor',
          'gill-attachment','gill-spacing','gill-size','gill-color','stalk-shape',
          'stalk-root','stalk-surface-above-ring','stalk-surface-below-ring',
          'stalk-color-above-ring','stalk-color-below-ring','veil-type',
          'veil-color','ring-number','ring-type','spore-print-color','population',
          'habitat']

df = pandas.read_csv("agaricus-lepiota.data", header=None, na_values='?', names=c_Names) #To read the csv file using pa
df.fillna(0) # convert all ? to 0's
# Converting the categorical data into binary
df = pandas.get_dummies(df)
df.reset_index(drop=True, inplace=True) # removing the first column which displays the line number

#Split the data into training(60), validation(20) and test(20) datasets
training, validate, testing = numpy.split(df, [int(.6*len(df)), int(.8*len(df))])


# Save the data to the respective files.
training.to_csv('training.txt', header=False, index = False)
print("Successfully Created training.txt")
validate.to_csv('validation.txt', header=False, index = False)
print("Successfully Created validation.txt")
testing.to_csv('testing.txt', header=False, index = False)
print("Successfully Created testing.txt")
```

```
Successfully Created training.txt
Successfully Created validation.txt
Successfully Created testing.txt
```

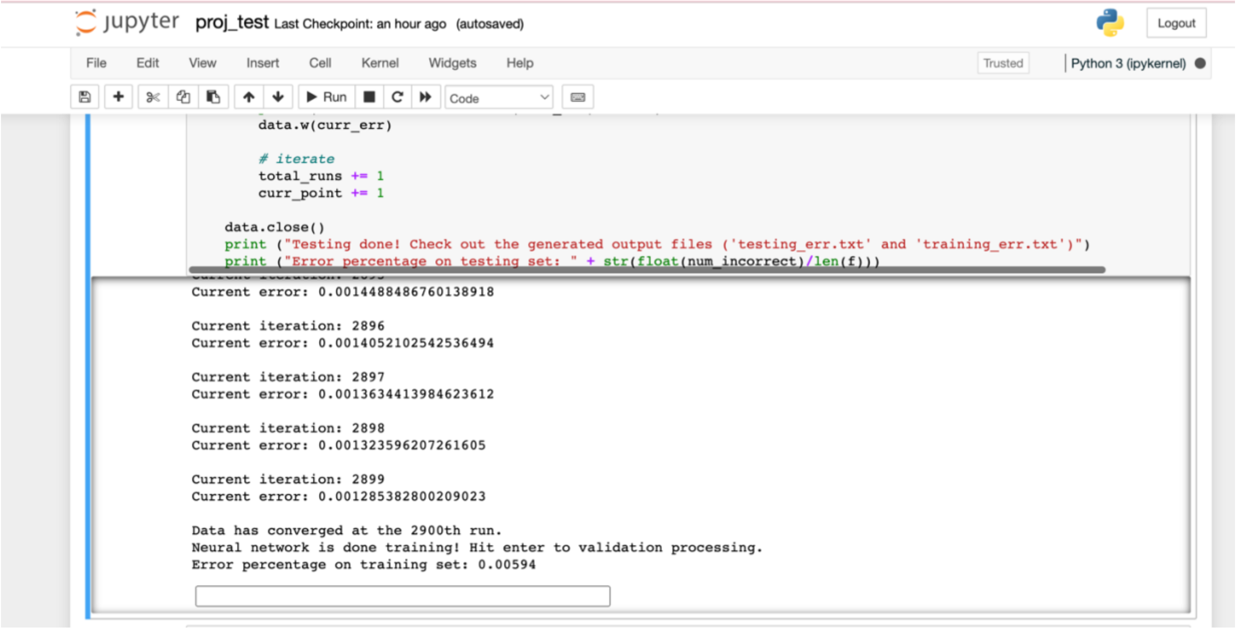
Screenshot : 1 (data_con.py)



```
        # print information about the current iteration
        print ("Current iteration: " + str(total_runs))
        print ("Current Error: " + str(curr_err) + "\n")
        data.w(curr_err)

        # iterate
        total_runs += 1
        curr_point += 1

    data.close()
    print ("Testing done! Check out the generated output files ('testing_err.txt' and 'training_err.txt')")
    print ("Error percentage on testing set: " + str(float(num_incorrect)/len(f)))
```

```
Parsing the training dataset...
Begining training the neural network:
Current iteration: 0
Current error: 0.0010218609379284748

Current iteration: 1
Current error: 0.4562575441779249

Current iteration: 2
Current error: 0.44588948828586183

Current iteration: 3
Current error: 0.0023782459526900572

Current iteration: 4
Current error: 0.43440606524104636

Current iteration: 5
Current error: 0.41976105932097013
```

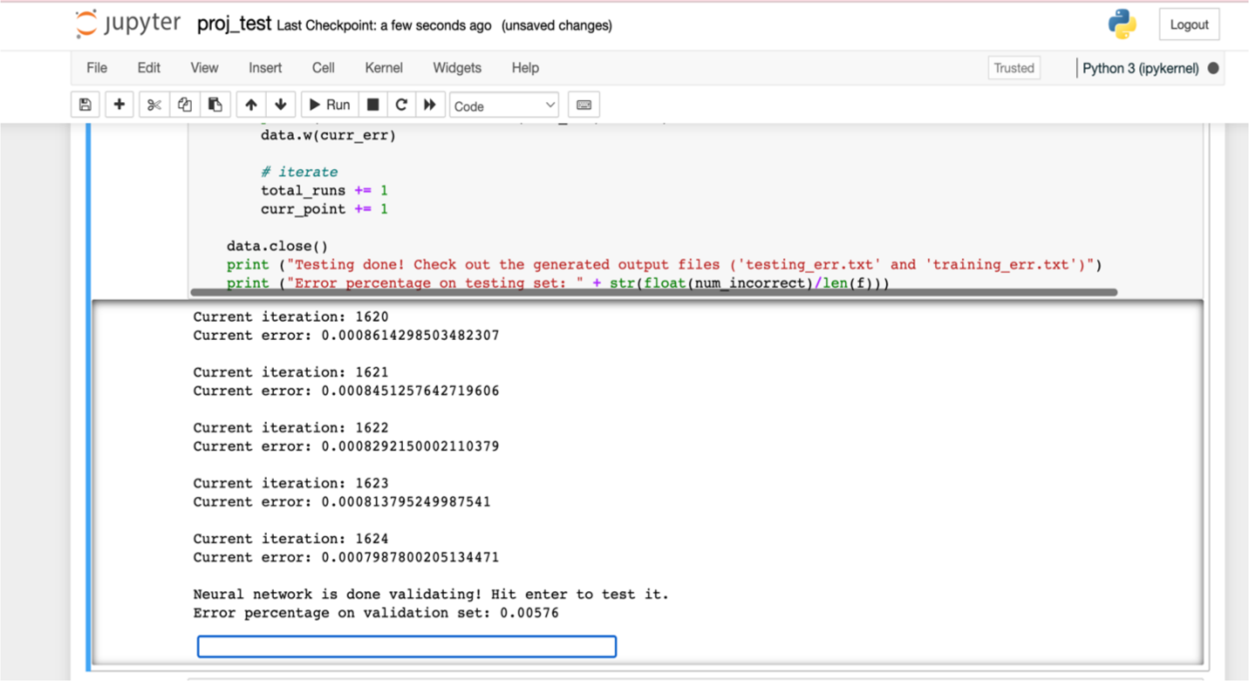
Screenshot 2 : (proj_test.py)



```
        data.w(curr_err)

        # iterate
        total_runs += 1
        curr_point += 1

    data.close()
    print ("Testing done! Check out the generated output files ('testing_err.txt' and 'training_err.txt')")
    print ("Error percentage on testing set: " + str(float(num_incorrect)/len(f)))
```

```
Current error: 0.0014488486760138918

Current iteration: 2896
Current error: 0.0014052102542536494

Current iteration: 2897
Current error: 0.0013634413984623612

Current iteration: 2898
Current error: 0.001323596207261605

Current iteration: 2899
Current error: 0.001285382800209023

Data has converged at the 2900th run.
Neural network is done training! Hit enter to validation processing.
Error percentage on training set: 0.00594
```

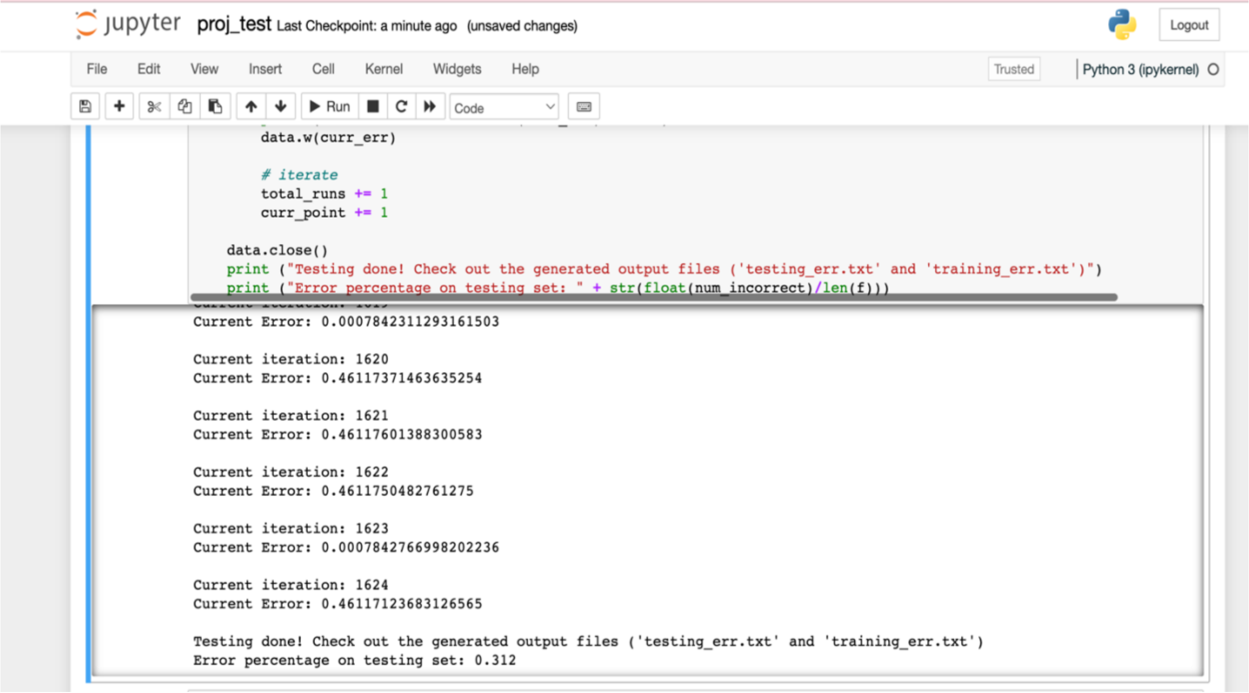
Screenshot 3 : (proj_test.py)



```
        data.w(curr_err)

        # iterate
        total_runs += 1
        curr_point += 1

    data.close()
    print ("Testing done! Check out the generated output files ('testing_err.txt' and 'training_err.txt')")
    print ("Error percentage on testing set: " + str(float(num_incorrect)/len(f)))
```

```
Current iteration: 1620
Current error: 0.0008614298503482307

Current iteration: 1621
Current error: 0.0008451257642719606

Current iteration: 1622
Current error: 0.0008292150002110379

Current iteration: 1623
Current error: 0.000813795249987541

Current iteration: 1624
Current error: 0.0007987800205134471

Neural network is done validating! Hit enter to test it.
Error percentage on validation set: 0.00576
```

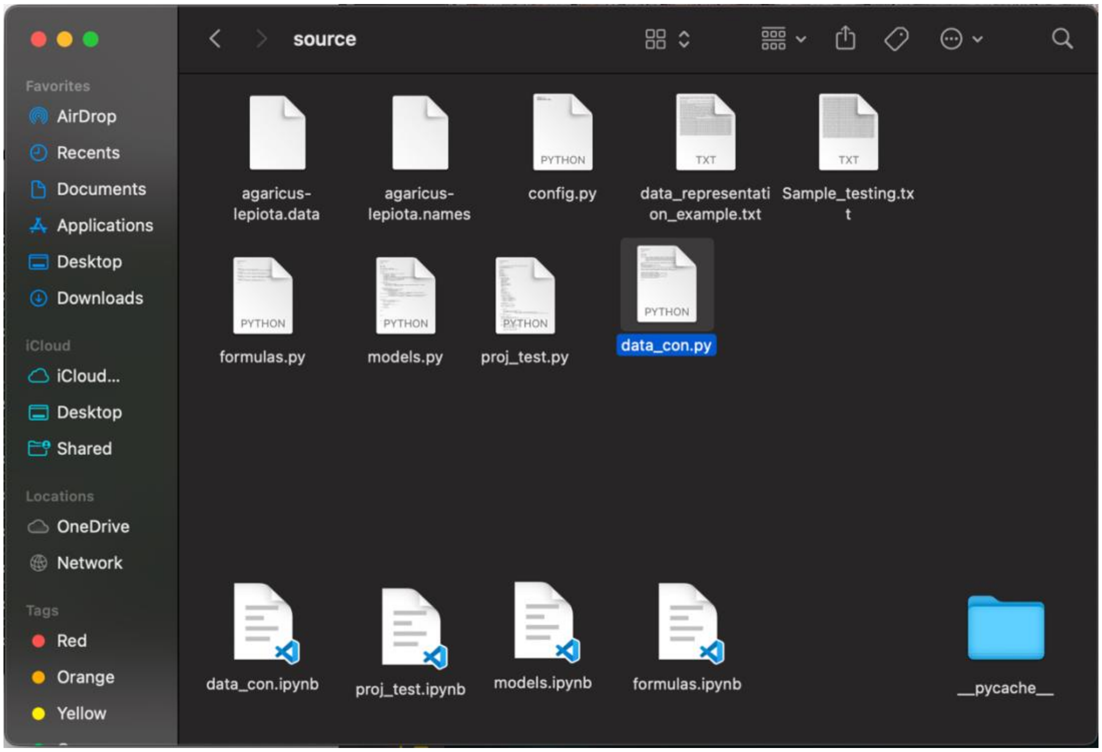
Screenshot 4 : (proj_test.py)



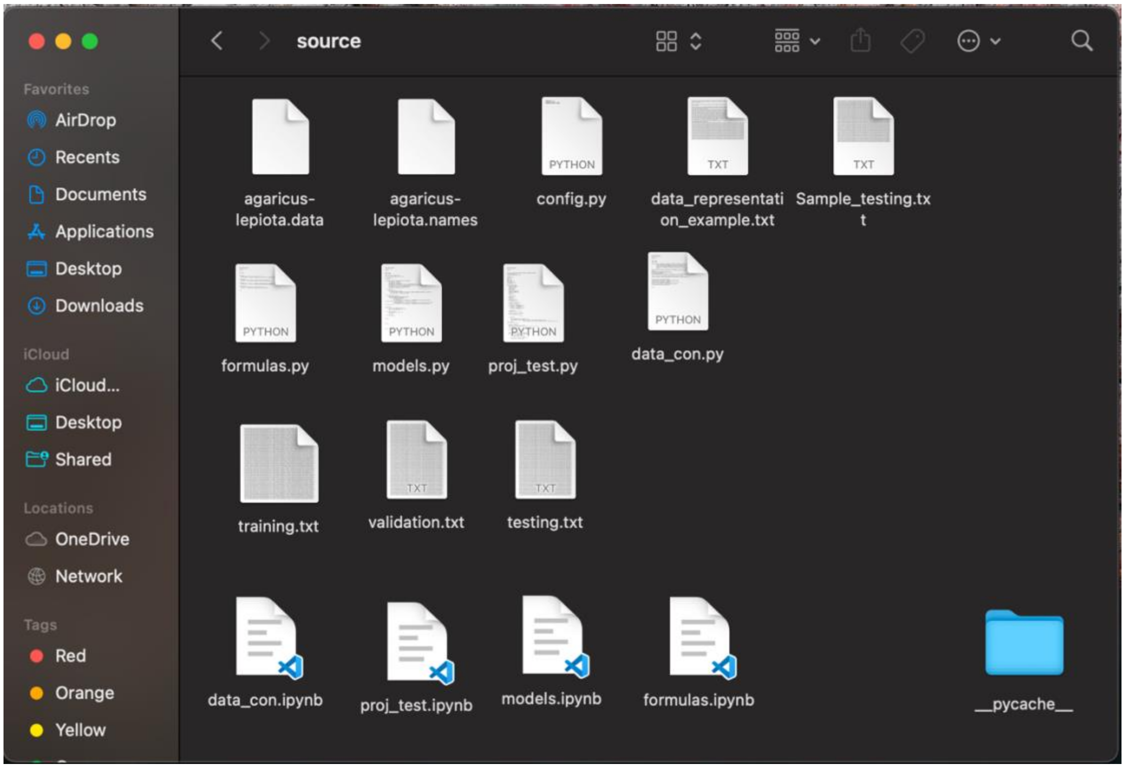
```
        data.w(curr_err)

        # iterate
        total_runs += 1
        curr_point += 1

    data.close()
    print ("Testing done! Check out the generated output files ('testing_err.txt' and 'training_err.txt')")
    print ("Error percentage on testing set: " + str(float(num_incorrect)/len(f)))
```

```
Current Error: 0.0007842311293161503

Current iteration: 1620
Current Error: 0.46117371463635254

Current iteration: 1621
Current Error: 0.46117601388300583

Current iteration: 1622
Current Error: 0.4611750482761275

Current iteration: 1623
Current Error: 0.0007842766998202236

Current iteration: 1624
Current Error: 0.46117123683126565

Testing done! Check out the generated output files ('testing_err.txt' and 'training_err.txt')
Error percentage on testing set: 0.312
```

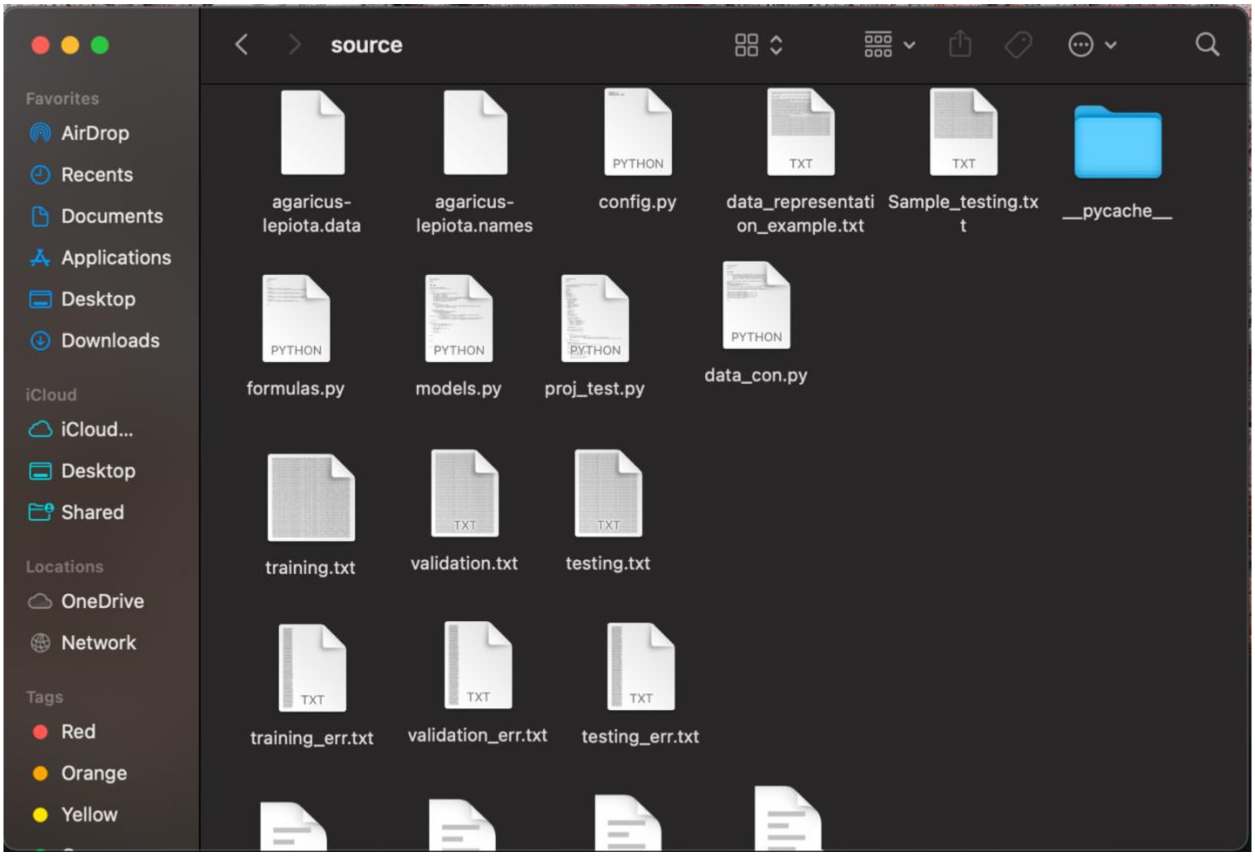
Screenshot 5 : (proj_test.py)



Screenshot 5 : Basic Files



Screenshot 6 : Folder with training.txt,validation.txt and testing.txt



Screenshot 7 : Folder with training_err.txt, validation.txt and testing_err.txt

Conclusion: Successfully created model that can predict weather the mushroom is edible or poisonous.