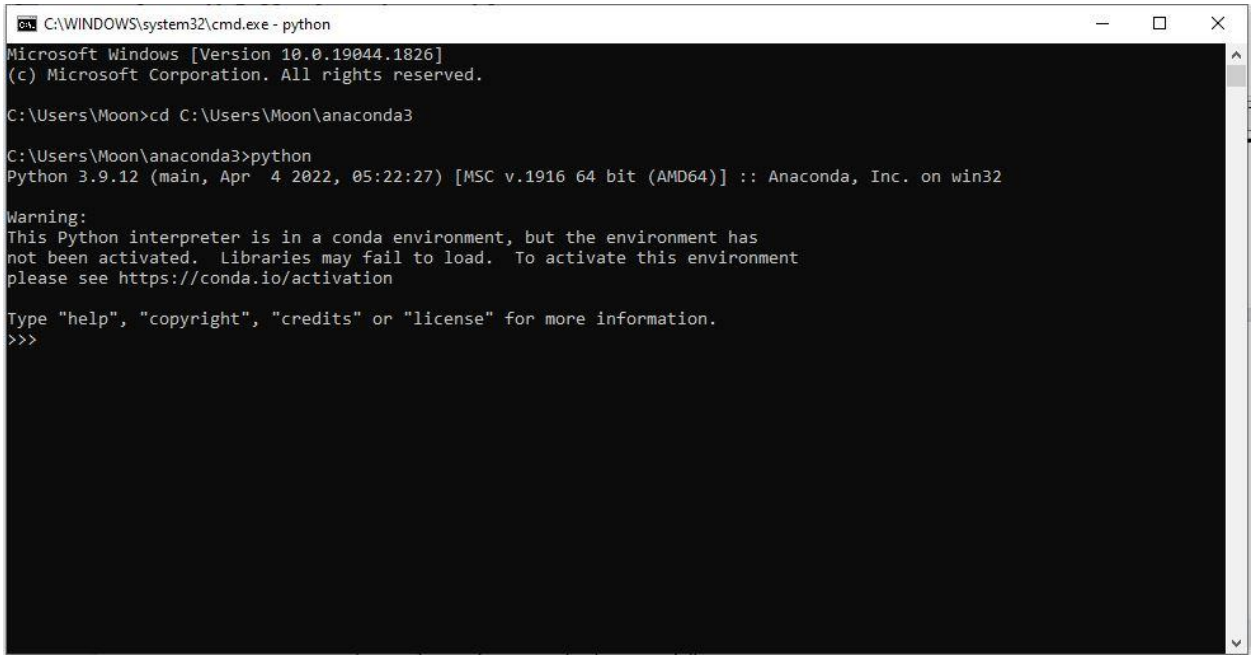
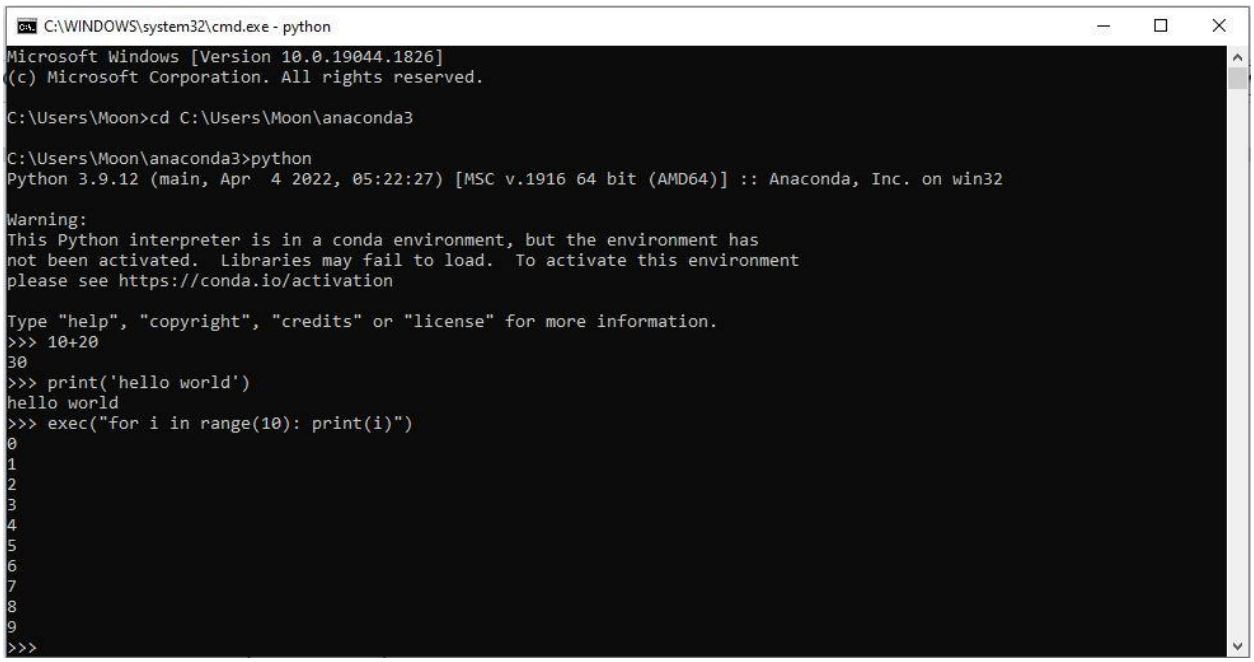
Name: Najmun Nahar

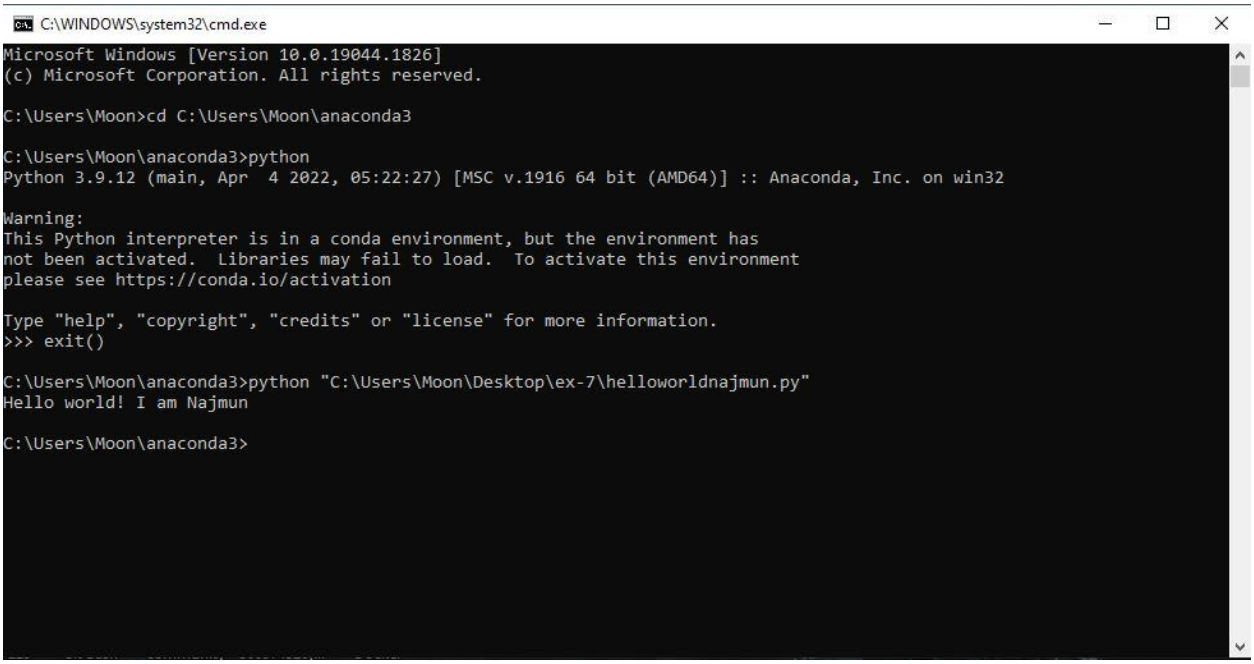
ID:301160081

Exercise #1

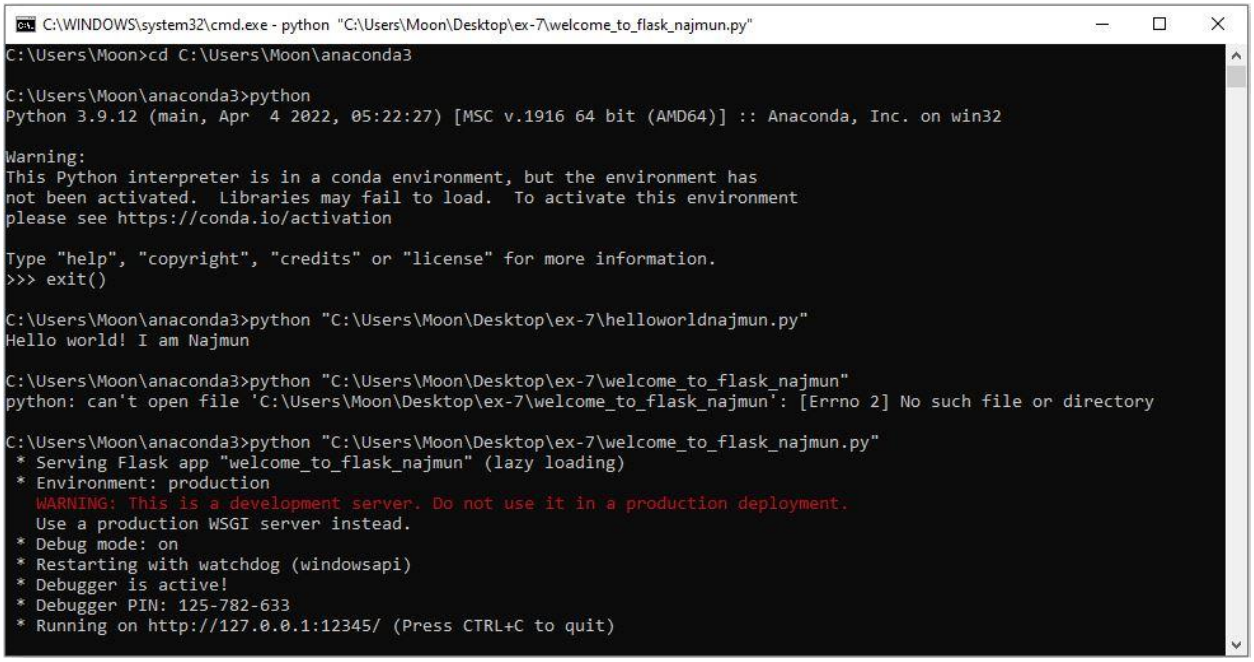




Exercise # 2

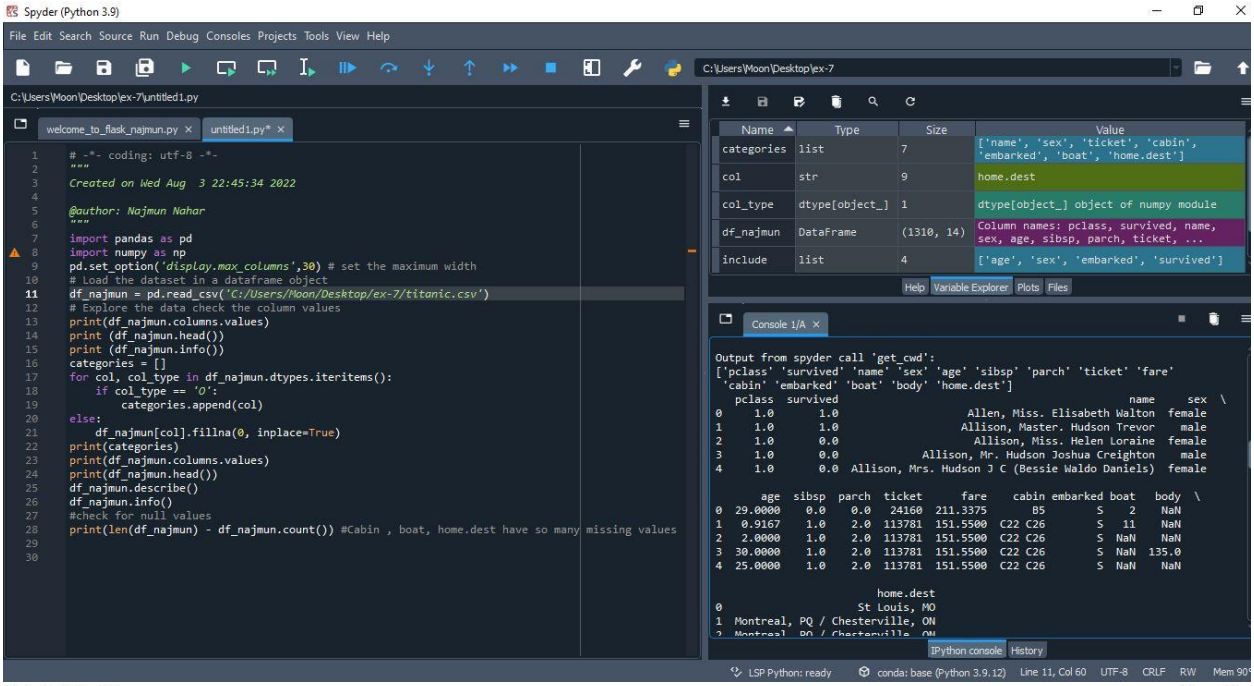


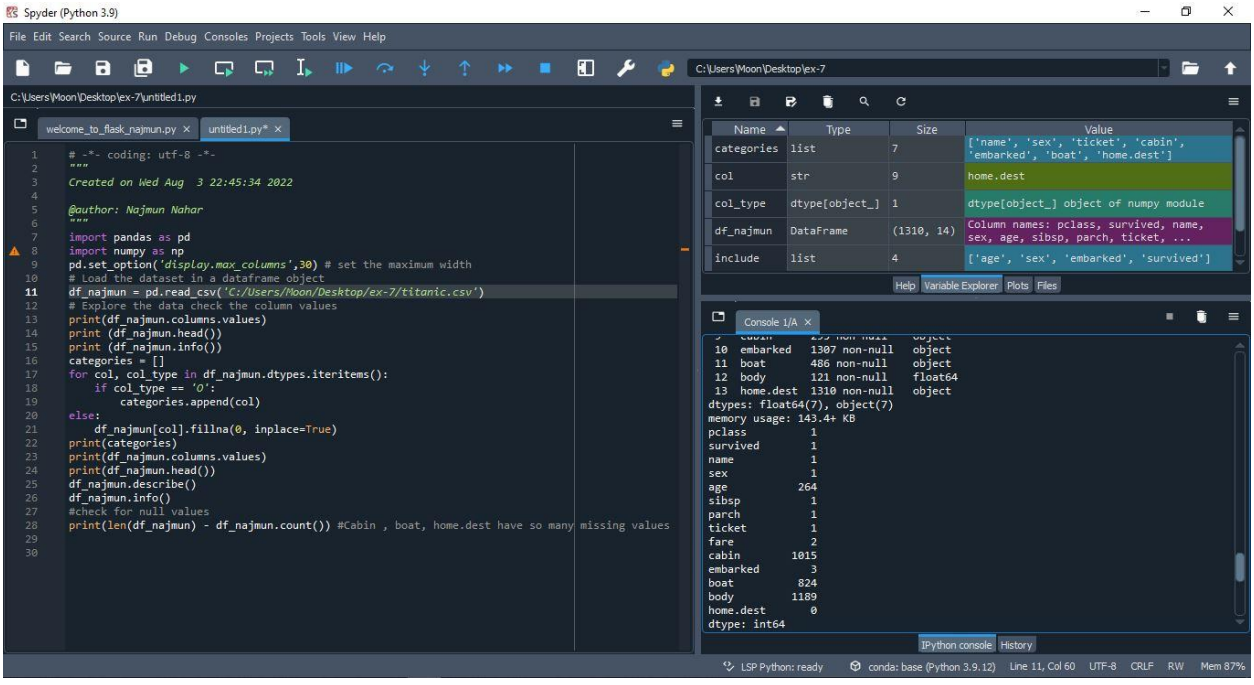
Exercise #3

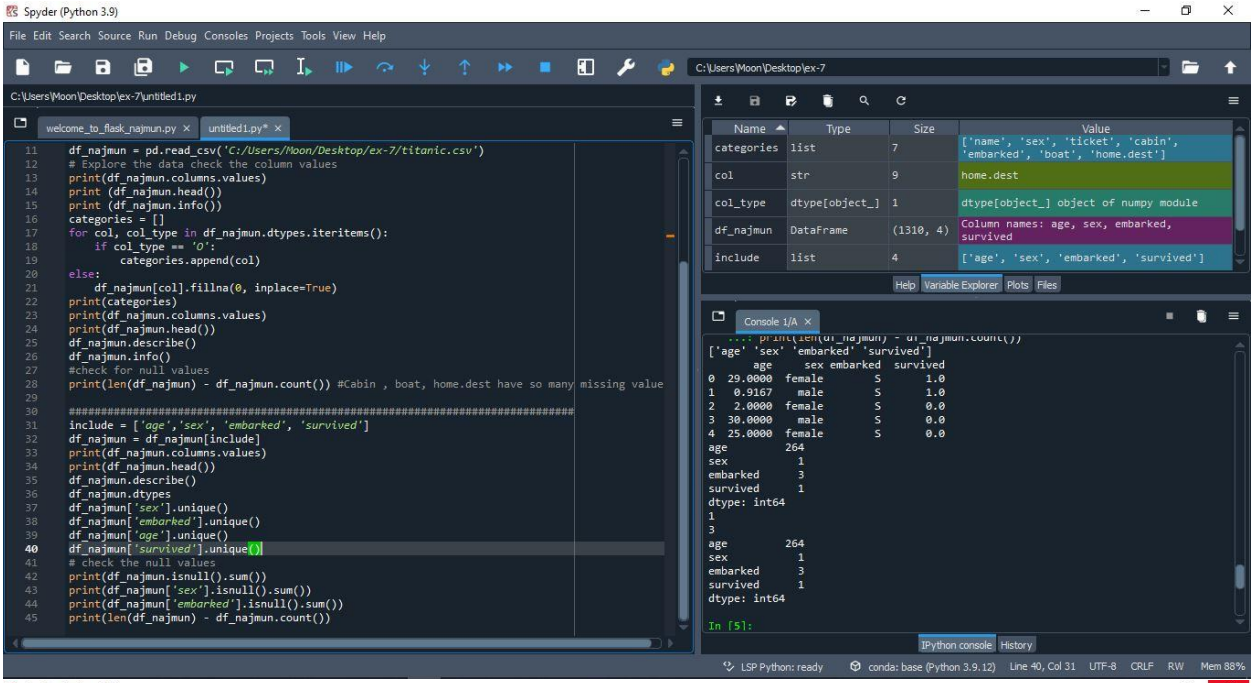


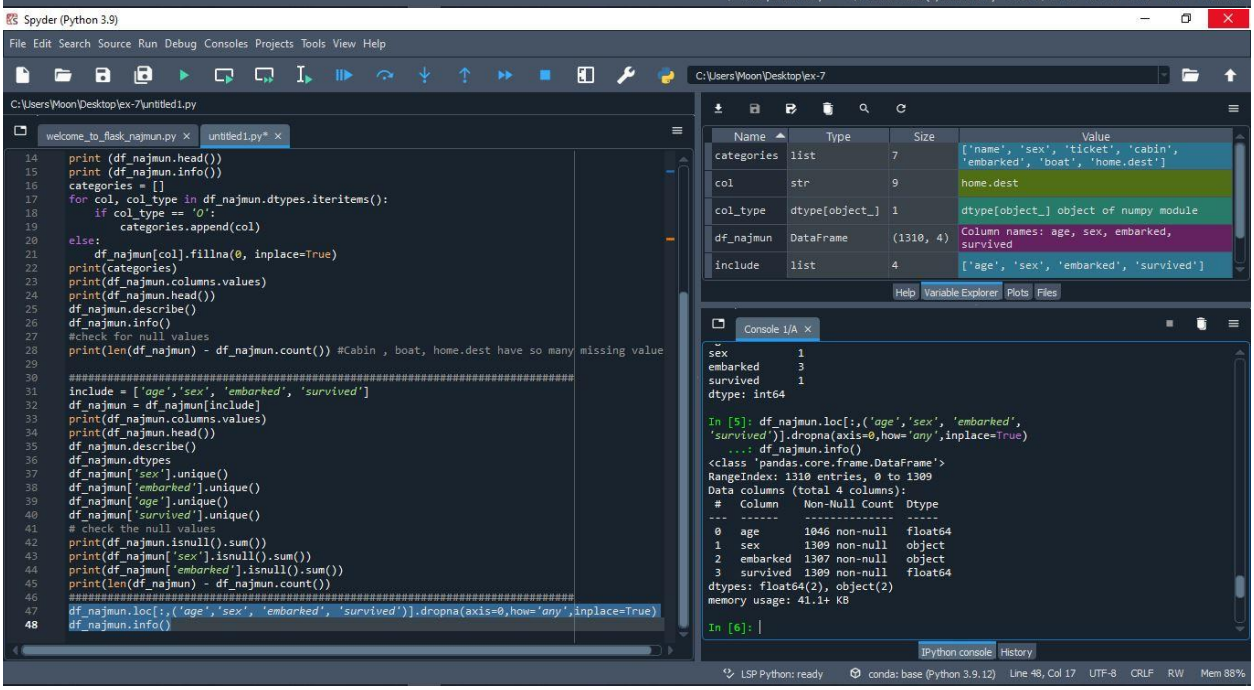

Exercise#10: Instructions - 22M
localhost:12345
localhost:12345
Welcome to machine learning model APIs!, Najmun

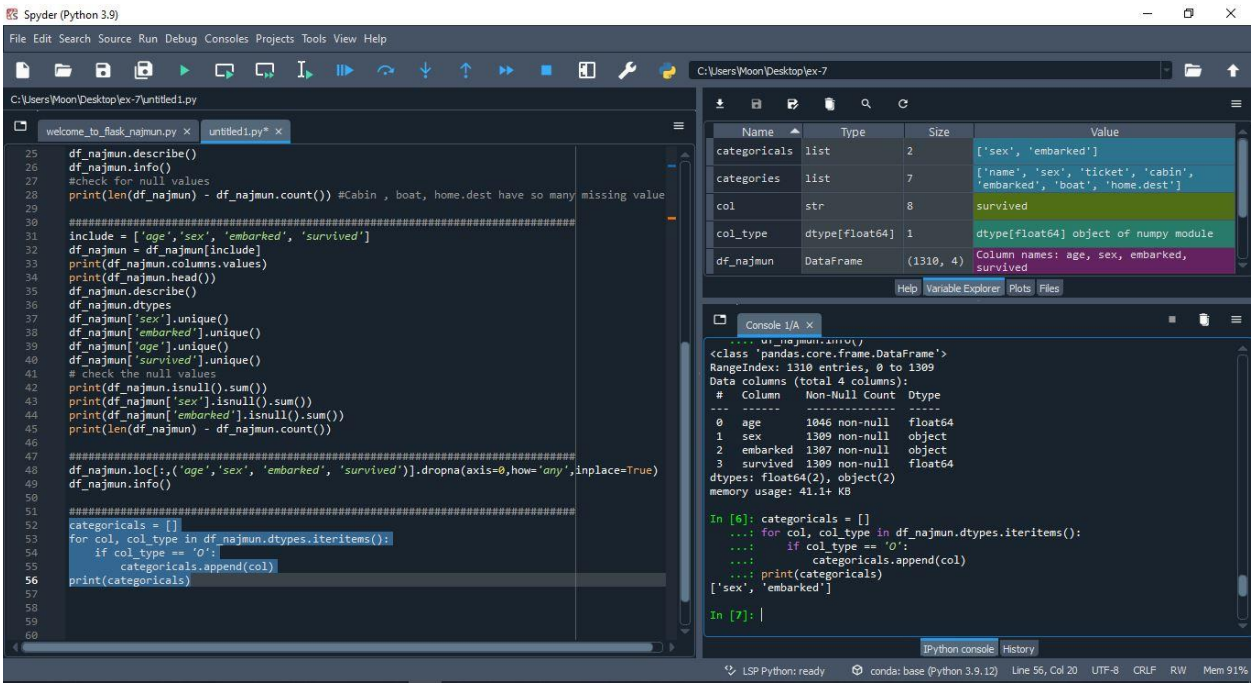
Exercise #4

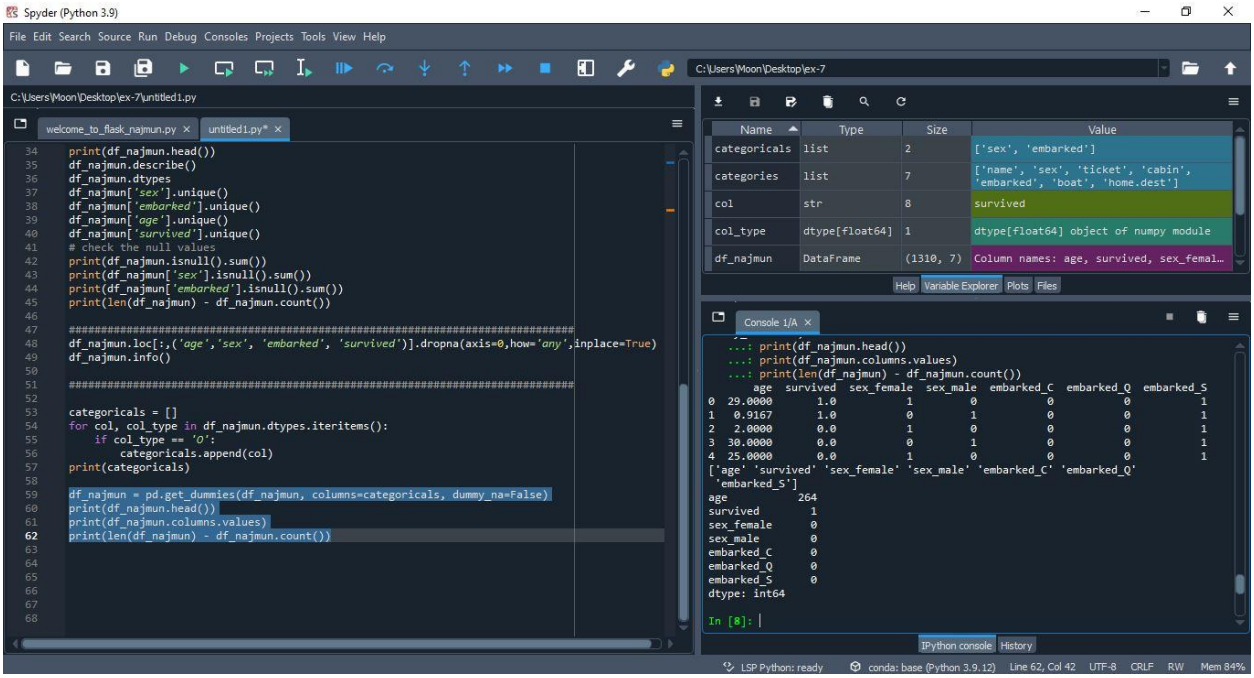

Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Moon\Desktop\ex-7
C:\Users\Moon\Desktop\ex-7\untitled1.py
welcome_to_flask_najmun.py
untitled1.py*
-*- coding: utf-8 -*-
"""
Created on Wed Aug 3 22:45:34 2022
@author: Najmun Nahar
"""
import pandas as pd
import numpy as np
pd.set_option('display.max_columns',30) # set the maximum width
# Load the dataset in a dataframe object
df_najmun = pd.read_csv('C:/Users/Moon/Desktop/ex-7/titanic.csv')
# Explore the data check the column values
print(df_najmun.columns.values)
print (df_najmun.head())
print (df_najmun.info())
categories = []
for col, col_type in df_najmun.dtypes.iteritems():
if col_type == 'O':
categories.append(col)
else:
df_najmun[col].fillna(0, inplace=True)
print(categories)
print(df_najmun.columns.values)
print(df_najmun.head())
df_najmun.describe()
df_najmun.info()
#check for null values
print(len(df_najmun) - df_najmun.count()) #Cabin , boat, home.dest have so many missing values
Name Type Size Value
categories list 7 ['name', 'sex', 'ticket', 'cabin', 'embarked', 'boat', 'home.dest']
col str 9 home.dest
col_type dtype[object_] 1 dtype[object_] object of numpy module
df_najmun DataFrame (1310, 14) Column names: pclass, survived, name, sex, age, sibsp, parch, ticket, ...
include list 4 ['age', 'sex', 'embarked', 'survived']
Help Variable Explorer Plots Files
Console 1/A
Output from spyder call 'get_cwd':
['pclass' 'survived' 'name' 'sex' 'age' 'sibsp' 'parch' 'ticket' 'fare' 'cabin' 'embarked' 'boat' 'body' 'home.dest']
pclass survived name sex
0 1.0 1.0 Allen, Miss. Elisabeth Walton female
1 1.0 1.0 Allison, Master. Hudson Trevor male
2 1.0 0.0 Allison, Miss. Helen Loraine female
3 1.0 0.0 Allison, Mr. Hudson Joshua Creighton male
4 1.0 0.0 Allison, Mrs. Hudson J C (Bessie Waldo Daniels) female
age sibsp parch ticket fare cabin embarked boat body
0 29.0000 0.0 0.0 24160 211.3375 B5 S 2 NaN
1 0.9167 1.0 2.0 113781 151.5500 C22 C26 S 11 NaN
2 2.0000 1.0 2.0 113781 151.5500 C22 C26 S NaN NaN
3 30.0000 1.0 2.0 113781 151.5500 C22 C26 S NaN 135.0
4 25.0000 1.0 2.0 113781 151.5500 C22 C26 S NaN NaN
home.dest
0 St Louis, MO
1 Montreal, PQ / Chesterville, ON
IPython console History
LSP Python: ready
conda: base (Python 3.9.12)
Line 11, Col 60
UTF-8
CRLF
RW

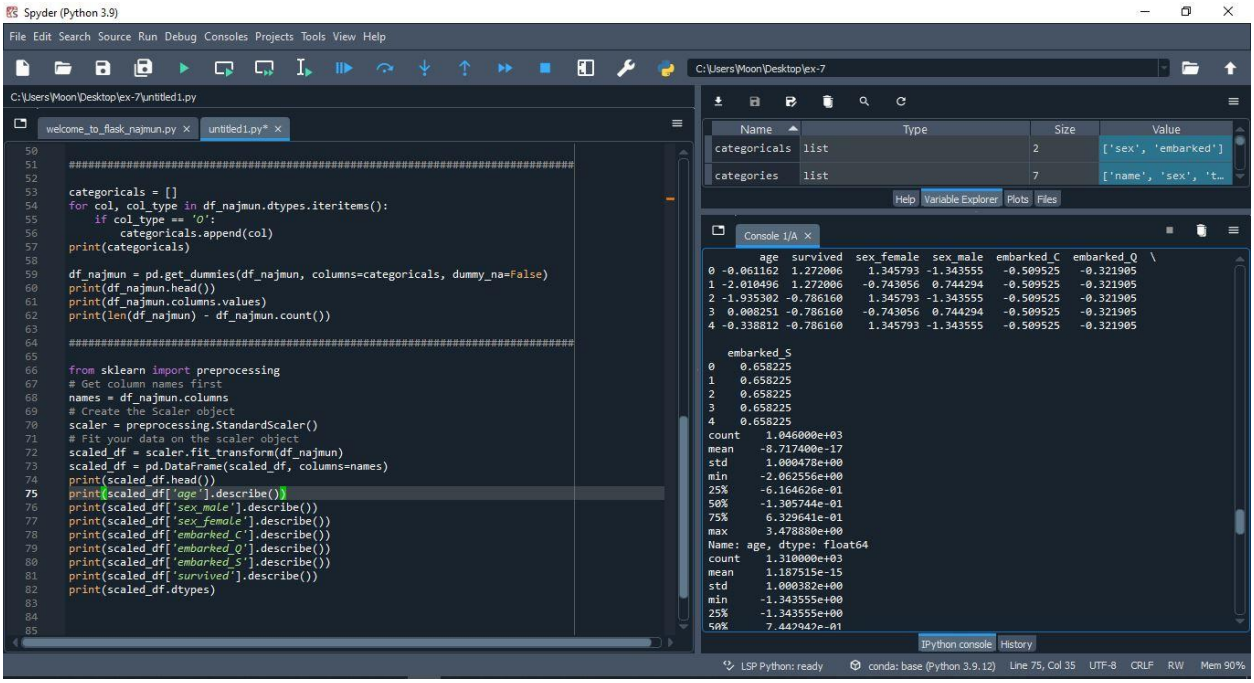

Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Moon\Desktop\ex-7
C:\Users\Moon\Desktop\ex-7\untitled1.py
welcome_to_flask_najmun.py
untitled1.py*
-*- coding: utf-8 -*-
"""
Created on Wed Aug 3 22:45:34 2022
@author: Najmun Nahar
"""
import pandas as pd
import numpy as np
pd.set_option('display.max_columns',30) # set the maximum width
# Load the dataset in a dataframe object
df_najmun = pd.read_csv('C:/Users/Moon/Desktop/ex-7/titanic.csv')
# Explore the data check the column values
print(df_najmun.columns.values)
print (df_najmun.head())
print (df_najmun.info())
categories = []
for col, col_type in df_najmun.dtypes.iteritems():
if col_type == 'O':
categories.append(col)
else:
df_najmun[col].fillna(0, inplace=True)
print(categories)
print(df_najmun.columns.values)
print(df_najmun.head())
df_najmun.describe()
df_najmun.info()
#check for null values
print(len(df_najmun) - df_najmun.count()) #Cabin , boat, home.dest have so many missing values
Name Type Size Value
categories list 7 ['name', 'sex', 'ticket', 'cabin', 'embarked', 'boat', 'home.dest']
col str 9 home.dest
col_type dtype[object_] 1 dtype[object_] object of numpy module
df_najmun DataFrame (1310, 14) Column names: pclass, survived, name, sex, age, sibsp, parch, ticket, ...
include list 4 ['age', 'sex', 'embarked', 'survived']
Help Variable Explorer Plots Files
Console 1/A
10 embarked 1307 non-null object
11 boat 486 non-null object
12 body 121 non-null float64
13 home.dest 1310 non-null object
dtypes: float64(7), object(7)
memory usage: 143.4+ KB
pclass 1
survived 1
name 1
sex 1
age 264
sibsp 1
parch 1
ticket 1
fare 2
cabin 1015
embarked 3
boat 824
body 1189
home.dest 0
dtype: int64
IPython console History
LSP Python: ready
conda: base (Python 3.9.12)
Line 11, Col 60
UTF-8
CRLF
RW

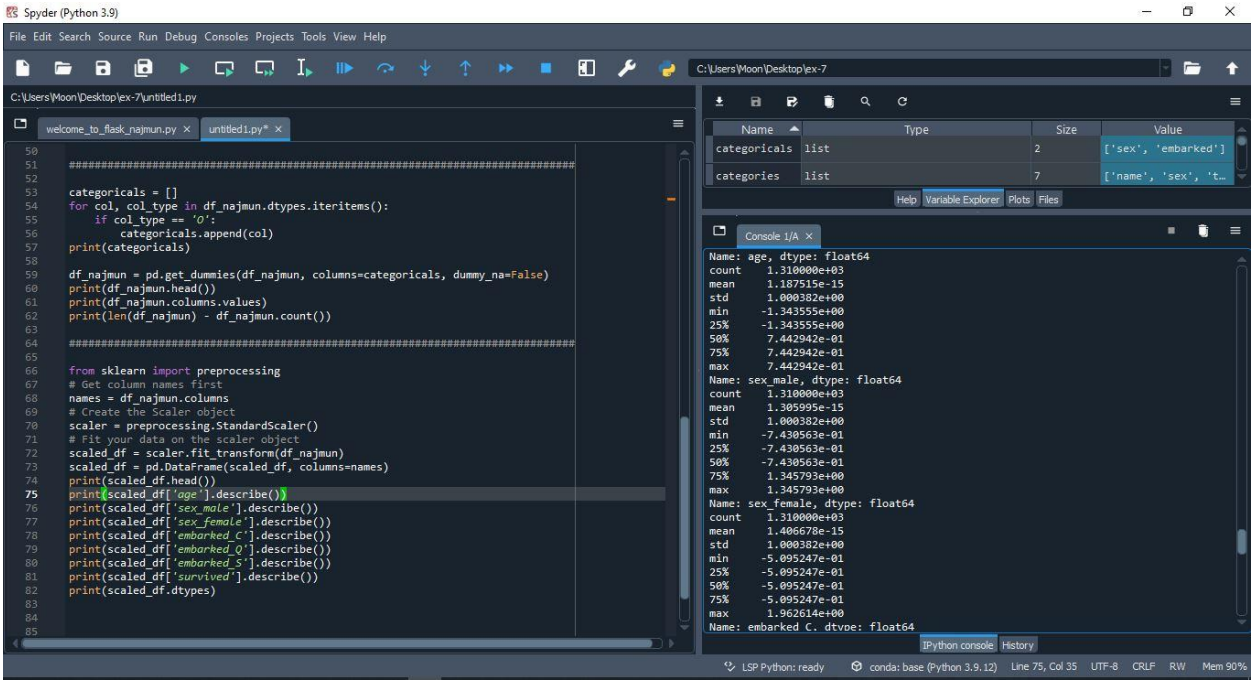

Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Moon\Desktop\ex-7
C:\Users\Moon\Desktop\ex-7\untitled1.py
welcome_to_flask_najmun.py
untitled1.py*
df_najmun = pd.read_csv('C:/Users/Moon/Desktop/ex-7/titanic.csv')
# Explore the data check the column values
print(df_najmun.columns.values)
print (df_najmun.head())
print (df_najmun.info())
categories = []
for col, col_type in df_najmun.dtypes.iteritems():
if col_type == 'O':
categories.append(col)
else:
df_najmun[col].fillna(0, inplace=True)
print(categories)
print(df_najmun.columns.values)
print(df_najmun.head())
df_najmun.describe()
df_najmun.info()
#check for null values
print(len(df_najmun) - df_najmun.count()) #Cabin , boat, home.dest have so many missing value
include = ['age','sex', 'embarked', 'survived']
df_najmun = df_najmun[include]
print(df_najmun.columns.values)
print(df_najmun.head())
df_najmun.describe()
df_najmun.dtypes
df_najmun['sex'].unique()
df_najmun['embarked'].unique()
df_najmun['age'].unique()
df_najmun['survived'].unique()
# check the null values
print(df_najmun.isnull().sum())
print(df_najmun['sex'].isnull().sum())
print(df_najmun['embarked'].isnull().sum())
print(len(df_najmun) - df_najmun.count())
Name Type Size Value
categories list 7 ['name', 'sex', 'ticket', 'cabin', 'embarked', 'boat', 'home.dest']
col str 9 home.dest
col_type dtype[object_] 1 dtype[object_] object of numpy module
df_najmun DataFrame (1310, 4) Column names: age, sex, embarked, survived
include list 4 ['age', 'sex', 'embarked', 'survived']
Help Variable Explorer Plots Files
Console 1/A
['age' 'sex' 'embarked' 'survived']
age sex embarked survived
0 29.0000 female S 1.0
1 0.9167 male S 1.0
2 2.0000 female S 0.0
3 30.0000 male S 0.0
4 25.0000 female S 0.0
age 264
sex 1
embarked 3
survived 1
dtype: int64
1
3
age 264
sex 1
embarked 3
survived 1
dtype: int64
In [5]:
IPython console History
LSP Python: ready
conda: base (Python 3.9.12)
Line 40, Col 31
UTF-8
CRLF
RW
Mem 88%

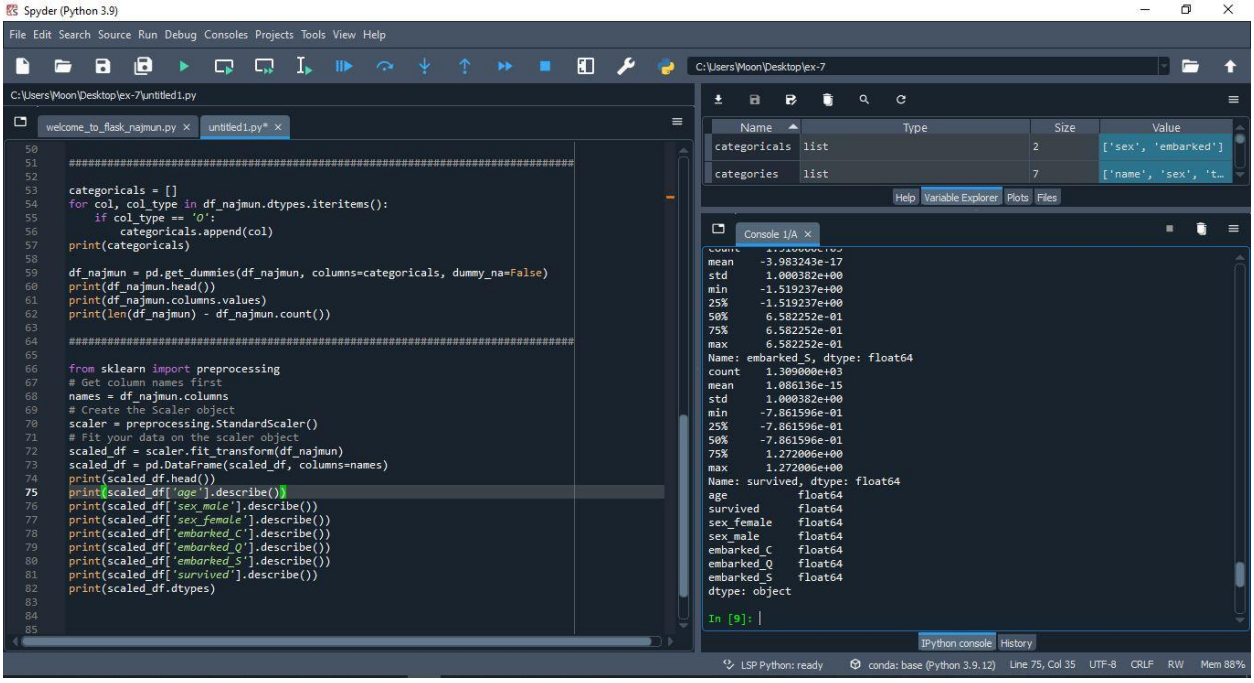

Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Moon\Desktop\ex-7
C:\Users\Moon\Desktop\ex-7\untitled1.py
welcome_to_flask_najmun.py
untitled1.py*
print (df_najmun.head())
print (df_najmun.info())
categories = []
for col, col_type in df_najmun.dtypes.iteritems():
if col_type == 'O':
categories.append(col)
else:
df_najmun[col].fillna(0, inplace=True)
print(categories)
print(df_najmun.columns.values)
print(df_najmun.head())
df_najmun.describe()
df_najmun.info()
#check for null values
print(len(df_najmun) - df_najmun.count()) #Cabin , boat, home.dest have so many missing value
include = ['age','sex', 'embarked', 'survived']
df_najmun = df_najmun[include]
print(df_najmun.columns.values)
print(df_najmun.head())
df_najmun.describe()
df_najmun.dtypes
df_najmun['sex'].unique()
df_najmun['embarked'].unique()
df_najmun['age'].unique()
df_najmun['survived'].unique()
# check the null values
print(df_najmun.isnull().sum())
print(df_najmun['sex'].isnull().sum())
print(df_najmun['embarked'].isnull().sum())
print(len(df_najmun) - df_najmun.count())
df_najmun.loc[:,('age','sex', 'embarked', 'survived')].dropna(axis=0,how='any',inplace=True)
df_najmun.info()
Name Type Size Value
categories list 7 ['name', 'sex', 'ticket', 'cabin', 'embarked', 'boat', 'home.dest']
col str 9 home.dest
col_type dtype[object_] 1 dtype[object_] object of numpy module
df_najmun DataFrame (1310, 4) Column names: age, sex, embarked, survived
include list 4 ['age', 'sex', 'embarked', 'survived']
Help Variable Explorer Plots Files
Console 1/A
sex 1
embarked 3
survived 1
dtype: int64
In [5]: df_najmun.loc[:,('age','sex', 'embarked', 'survived')].dropna(axis=0,how='any',inplace=True)
...: df_najmun.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1310 entries, 0 to 1309
Data columns (total 4 columns):
Column Non-Null Count Dtype
0 age 1046 non-null float64
1 sex 1309 non-null object
2 embarked 1307 non-null object
3 survived 1309 non-null float64
dtypes: float64(2), object(2)
memory usage: 41.1+ KB
In [6]:
IPython console History
LSP Python: ready
conda: base (Python 3.9.12)
Line 48, Col 17
UTF-8
CRLF
RW
Mem 88%

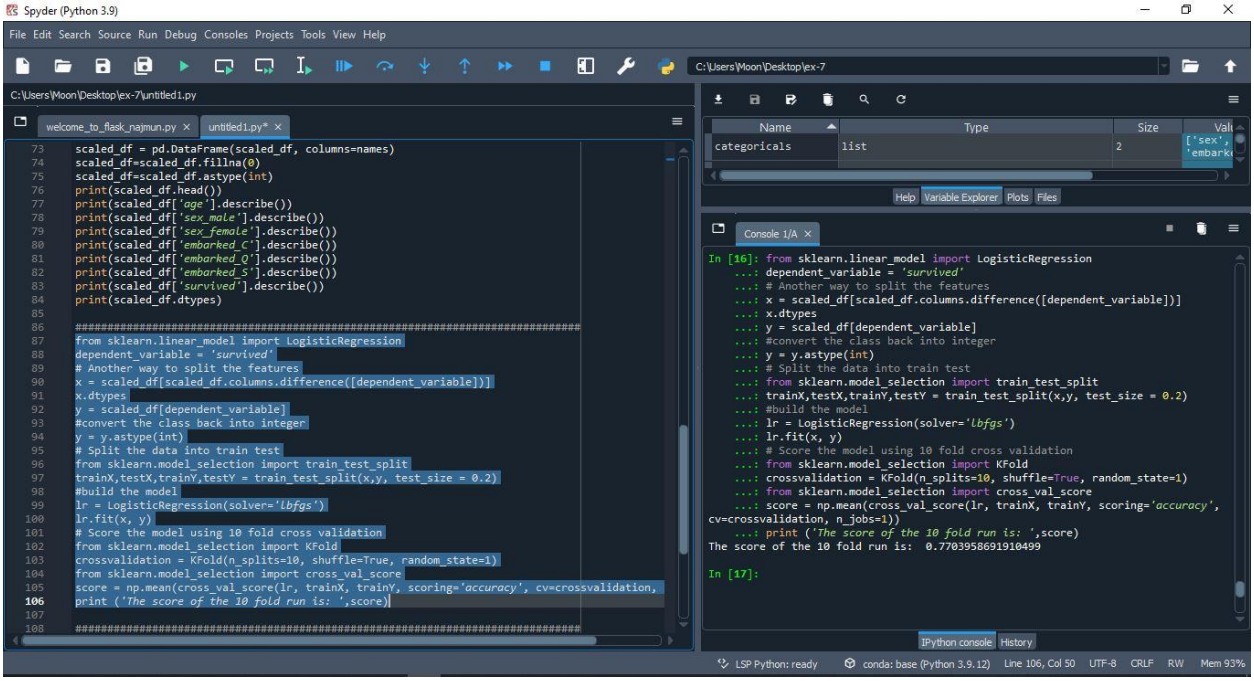

Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Moon\Desktop\ex-7
C:\Users\Moon\Desktop\ex-7\untitled1.py
welcome_to_flask_najmun.py
untitled1.py*
df_najmun.describe()
df_najmun.info()
#check for null values
print(len(df_najmun) - df_najmun.count()) #Cabin , boat, home.dest have so many missing value
include = ['age','sex', 'embarked', 'survived']
df_najmun = df_najmun[include]
print(df_najmun.columns.values)
print(df_najmun.head())
df_najmun.describe()
df_najmun.dtypes
df_najmun['sex'].unique()
df_najmun['embarked'].unique()
df_najmun['age'].unique()
df_najmun['survived'].unique()
# check the null values
print(df_najmun.isnull().sum())
print(df_najmun['sex'].isnull().sum())
print(df_najmun['embarked'].isnull().sum())
print(len(df_najmun) - df_najmun.count())
df_najmun.loc[:,('age','sex', 'embarked', 'survived')].dropna(axis=0,how='any',inplace=True)
df_najmun.info()
categoricals = []
for col, col_type in df_najmun.dtypes.iteritems():
if col_type == 'O':
categoricals.append(col)
print(categoricals)
Name Type Size Value
categoricals list 2 ['sex', 'embarked']
categories list 7 ['name', 'sex', 'ticket', 'cabin', 'embarked', 'boat', 'home.dest']
col str 8 survived
col_type dtype[float64] 1 dtype[float64] object of numpy module
df_najmun DataFrame (1310, 4) Column names: age, sex, embarked, survived
Help Variable Explorer Plots Files
Console 1/A
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1310 entries, 0 to 1309
Data columns (total 4 columns):
Column Non-Null Count Dtype
0 age 1046 non-null float64
1 sex 1309 non-null object
2 embarked 1307 non-null object
3 survived 1309 non-null float64
dtypes: float64(2), object(2)
memory usage: 41.1+ KB
In [6]: categoricals = []
...: for col, col_type in df_najmun.dtypes.iteritems():
...: if col_type == 'O':
...: categoricals.append(col)
...: print(categoricals)
['sex', 'embarked']
In [7]:
IPython console History
LSP Python: ready
conda: base (Python 3.9.12)
Line 56, Col 20
UTF-8
CRLF
RW
Mem 91%


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Moon\Desktop\ex-7
C:\Users\Moon\Desktop\ex-7\untitled1.py
welcome_to_flask_najmun.py
untitled1.py*
print(df_najmun.head())
df_najmun.describe()
df_najmun.dtypes
df_najmun['sex'].unique()
df_najmun['embarked'].unique()
df_najmun['age'].unique()
df_najmun['survived'].unique()
# check the null values
print(df_najmun.isnull().sum())
print(df_najmun['sex'].isnull().sum())
print(df_najmun['embarked'].isnull().sum())
print(len(df_najmun) - df_najmun.count())
df_najmun.loc[:,('age','sex', 'embarked', 'survived')].dropna(axis=0,how='any',inplace=True)
df_najmun.info()
categoricals = []
for col, col_type in df_najmun.dtypes.iteritems():
if col_type == 'O':
categoricals.append(col)
print(categoricals)
df_najmun = pd.get_dummies(df_najmun, columns=categoricals, dummy_na=False)
print(df_najmun.head())
print(df_najmun.columns.values)
print(len(df_najmun) - df_najmun.count())
Name Type Size Value
categoricals list 2 ['sex', 'embarked']
categories list 7 ['name', 'sex', 'ticket', 'cabin', 'embarked', 'boat', 'home.dest']
col str 8 survived
col_type dtype[float64] 1 dtype[float64] object of numpy module
df_najmun DataFrame (1310, 7) Column names: age, survived, sex_femal…
Help Variable Explorer Plots Files
Console 1/A
...: print(df_najmun.head())
...: print(df_najmun.columns.values)
...: print(len(df_najmun) - df_najmun.count())
age survived sex_female sex_male embarked_C embarked_Q embarked_S
0 29.0000 1.0 1 0 0 0 1
1 0.9167 1.0 0 1 0 0 1
2 2.0000 0.0 1 0 0 0 1
3 30.0000 0.0 0 1 0 0 1
4 25.0000 0.0 1 0 0 0 1
['age' 'survived' 'sex_female' 'sex_male' 'embarked_C' 'embarked_Q'
'embarked_S']
age 264
survived 1
sex_female 0
sex_male 0
embarked_C 0
embarked_Q 0
embarked_S 0
dtype: int64
In [8]:
IPython console History
LSP Python: ready
conda: base (Python 3.9.12)
Line 62, Col 42
UTF-8
CRLF
RW
Mem 84%

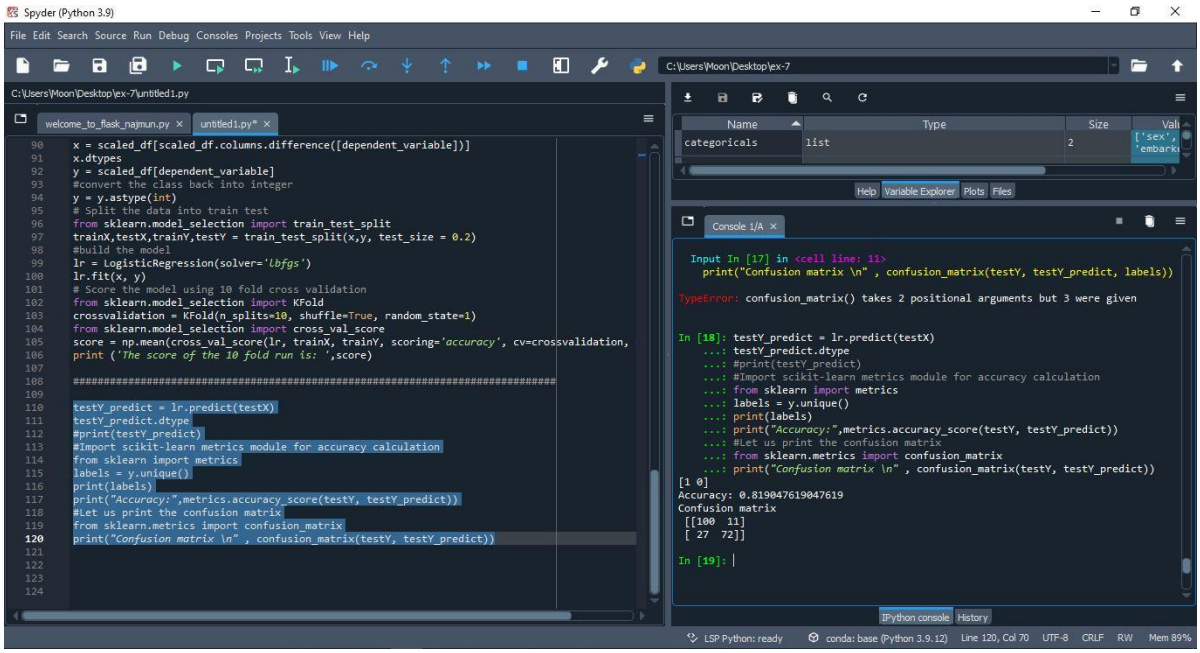

Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Moon\Desktop\ex-7
C:\Users\Moon\Desktop\ex-7\untitled1.py
welcome_to_flask_najmun.py
untitled1.py*
categoricals = []
for col, col_type in df_najmun.dtypes.iteritems():
if col_type == 'O':
categoricals.append(col)
print(categoricals)
df_najmun = pd.get_dummies(df_najmun, columns=categoricals, dummy_na=False)
print(df_najmun.head())
print(df_najmun.columns.values)
print(len(df_najmun) - df_najmun.count())
from sklearn import preprocessing
# Get column names first
names = df_najmun.columns
# Create the Scaler object
scaler = preprocessing.StandardScaler()
# Fit your data on the scaler object
scaled_df = scaler.fit_transform(df_najmun)
scaled_df = pd.DataFrame(scaled_df, columns=names)
print(scaled_df.head())
print(scaled_df['age'].describe())
print(scaled_df['sex_male'].describe())
print(scaled_df['sex_female'].describe())
print(scaled_df['embarked_C'].describe())
print(scaled_df['embarked_Q'].describe())
print(scaled_df['embarked_S'].describe())
print(scaled_df['survived'].describe())
print(scaled_df.dtypes)
Name Type Size Value
categoricals list 2 ['sex', 'embarked']
categories list 7 ['name', 'sex', 't…
Help Variable Explorer Plots Files
Console 1/A
age survived sex_female sex_male embarked_C embarked_Q \
0 -0.061162 1.272006 1.345793 -1.343555 -0.509525 -0.321905
1 -2.010496 1.272006 -0.743056 0.744294 -0.509525 -0.321905
2 -1.935302 -0.786160 1.345793 -1.343555 -0.509525 -0.321905
3 0.008251 -0.786160 -0.743056 0.744294 -0.509525 -0.321905
4 -0.338812 -0.786160 1.345793 -1.343555 -0.509525 -0.321905
embarked_S
0 0.658225
1 0.658225
2 0.658225
3 0.658225
4 0.658225
count 1.046000e+03
mean -8.717400e-17
std 1.000478e+00
min -2.062556e+00
25% -6.164626e-01
50% -1.305744e-01
75% 6.329641e-01
max 3.478880e+00
Name: age, dtype: float64
count 1.310000e+03
mean 1.187515e-15
std 1.000382e+00
min -1.343555e+00
25% -1.343555e+00
IPython console History
LSP Python: ready
conda: base (Python 3.9.12)
Line 75, Col 35
UTF-8
CRLF
RW
Mem 90%

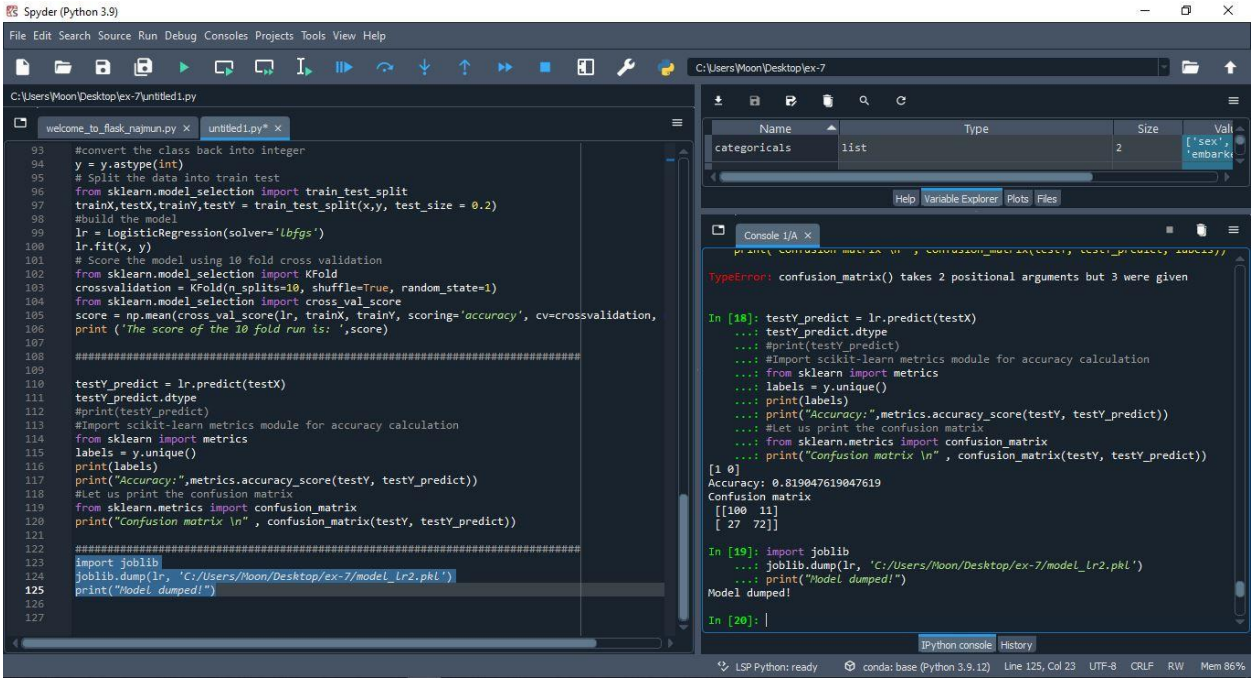

Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Moon\Desktop\ex-7
C:\Users\Moon\Desktop\ex-7\untitled1.py
welcome_to_flask_najmun.py
untitled1.py*
categoricals = []
for col, col_type in df_najmun.dtypes.iteritems():
if col_type == 'O':
categoricals.append(col)
print(categoricals)
df_najmun = pd.get_dummies(df_najmun, columns=categoricals, dummy_na=False)
print(df_najmun.head())
print(df_najmun.columns.values)
print(len(df_najmun) - df_najmun.count())
from sklearn import preprocessing
# Get column names first
names = df_najmun.columns
# Create the Scaler object
scaler = preprocessing.StandardScaler()
# Fit your data on the scaler object
scaled_df = scaler.fit_transform(df_najmun)
scaled_df = pd.DataFrame(scaled_df, columns=names)
print(scaled_df.head())
print(scaled_df['age'].describe())
print(scaled_df['sex_male'].describe())
print(scaled_df['sex_female'].describe())
print(scaled_df['embarked_C'].describe())
print(scaled_df['embarked_Q'].describe())
print(scaled_df['embarked_S'].describe())
print(scaled_df['survived'].describe())
print(scaled_df.dtypes)
Name Type Size Value
categoricals list 2 ['sex', 'embarked']
categories list 7 ['name', 'sex', 't…
Help Variable Explorer Plots Files
Console 1/A
Name: age, dtype: float64
count 1.310000e+03
mean 1.187515e-15
std 1.000382e+00
min -1.343555e+00
25% -1.343555e+00
50% 7.442942e-01
75% 7.442942e-01
max 7.442942e-01
Name: sex_male, dtype: float64
count 1.310000e+03
mean 1.305995e-15
std 1.000382e+00
min -7.430563e-01
25% -7.430563e-01
50% -7.430563e-01
75% 1.345793e+00
max 1.345793e+00
Name: sex_female, dtype: float64
count 1.310000e+03
mean 1.406678e-15
std 1.000382e+00
min -5.095247e-01
25% -5.095247e-01
50% -5.095247e-01
75% -5.095247e-01
max 1.962614e+00
Name: embarked_C, dtype: float64
IPython console History
LSP Python: ready
conda: base (Python 3.9.12)
Line 75, Col 35
UTF-8
CRLF
RW
Mem 90%

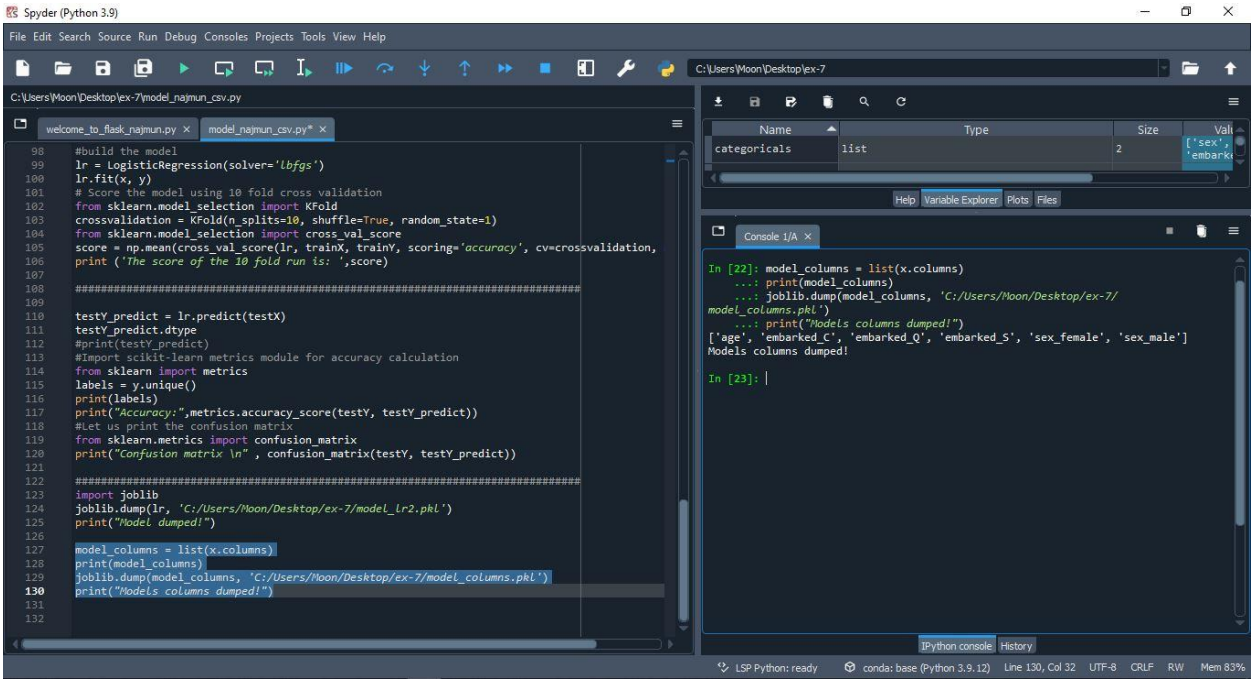

Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Moon\Desktop\ex-7
C:\Users\Moon\Desktop\ex-7\untitled1.py
welcome_to_flask_najmun.py
untitled1.py*
categoricals = []
for col, col_type in df_najmun.dtypes.iteritems():
if col_type == 'O':
categoricals.append(col)
print(categoricals)
df_najmun = pd.get_dummies(df_najmun, columns=categoricals, dummy_na=False)
print(df_najmun.head())
print(df_najmun.columns.values)
print(len(df_najmun) - df_najmun.count())
from sklearn import preprocessing
# Get column names first
names = df_najmun.columns
# Create the Scaler object
scaler = preprocessing.StandardScaler()
# Fit your data on the scaler object
scaled_df = scaler.fit_transform(df_najmun)
scaled_df = pd.DataFrame(scaled_df, columns=names)
print(scaled_df.head())
print(scaled_df['age'].describe())
print(scaled_df['sex_male'].describe())
print(scaled_df['sex_female'].describe())
print(scaled_df['embarked_C'].describe())
print(scaled_df['embarked_Q'].describe())
print(scaled_df['embarked_S'].describe())
print(scaled_df['survived'].describe())
print(scaled_df.dtypes)
Name Type Size Value
categoricals list 2 ['sex', 'embarked']
categories list 7 ['name', 'sex', 't…
Help Variable Explorer Plots Files
Console 1/A
mean -3.983243e-17
std 1.000382e+00
min -1.519237e+00
25% -1.519237e+00
50% 6.582252e-01
75% 6.582252e-01
max 6.582252e-01
Name: embarked_S, dtype: float64
count 1.309000e+03
mean 1.086136e-15
std 1.000382e+00
min -7.861596e-01
25% -7.861596e-01
50% -7.861596e-01
75% 1.272006e+00
max 1.272006e+00
Name: survived, dtype: float64
age float64
survived float64
sex_female float64
sex_male float64
embarked_C float64
embarked_Q float64
embarked_S float64
dtype: object
In [9]:
IPython console History
LSP Python: ready
conda: base (Python 3.9.12)
Line 75, Col 35
UTF-8
CRLF
RW
Mem 88%

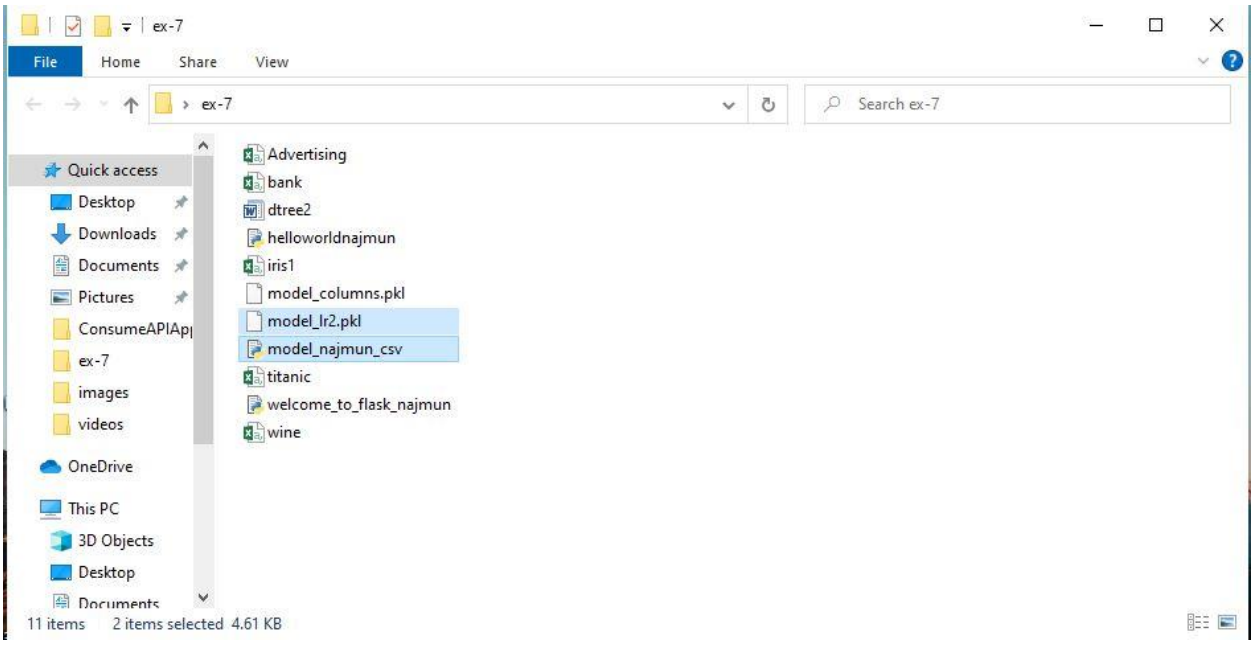

Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Moon\Desktop\ex-7
C:\Users\Moon\Desktop\ex-7\untitled1.py
welcome_to_flask_najmun.py
untitled1.py*
scaled_df = pd.DataFrame(scaled_df, columns=names)
scaled_df=scaled_df.fillna(0)
scaled_df=scaled_df.astype(int)
print(scaled_df.head())
print(scaled_df['age'].describe())
print(scaled_df['sex_male'].describe())
print(scaled_df['sex_female'].describe())
print(scaled_df['embarked_C'].describe())
print(scaled_df['embarked_Q'].describe())
print(scaled_df['embarked_S'].describe())
print(scaled_df['survived'].describe())
print(scaled_df.dtypes)
from sklearn.linear_model import LogisticRegression
dependent_variable = 'survived'
# Another way to split the features
x = scaled_df[scaled_df.columns.difference([dependent_variable])]
x.dtypes
y = scaled_df[dependent_variable]
#convert the class back into integer
y = y.astype(int)
# Split the data into train test
from sklearn.model_selection import train_test_split
trainX,testX,trainY,testY = train_test_split(x,y, test_size = 0.2)
#build the model
lr = LogisticRegression(solver='lbfgs')
lr.fit(x, y)
# Score the model using 10 fold cross validation
from sklearn.model_selection import KFold
crossvalidation = KFold(n_splits=10, shuffle=True, random_state=1)
from sklearn.model_selection import cross_val_score
score = np.mean(cross_val_score(lr, trainX, trainY, scoring='accuracy', cv=crossvalidation,
print ('The score of the 10 fold run is: ',score)
Name
Type
Size
categoricals
list
2
Help
Variable Explorer
Plots
Files
Console 1/A
In [16]: from sklearn.linear_model import LogisticRegression
...: dependent_variable = 'survived'
...: # Another way to split the features
...: x = scaled_df[scaled_df.columns.difference([dependent_variable])]
...: x.dtypes
...: y = scaled_df[dependent_variable]
...: #convert the class back into integer
...: y = y.astype(int)
...: # Split the data into train test
...: from sklearn.model_selection import train_test_split
...: trainX,testX,trainY,testY = train_test_split(x,y, test_size = 0.2)
...: #build the model
...: lr = LogisticRegression(solver='lbfgs')
...: lr.fit(x, y)
...: # Score the model using 10 fold cross validation
...: from sklearn.model_selection import KFold
...: crossvalidation = KFold(n_splits=10, shuffle=True, random_state=1)
...: from sklearn.model_selection import cross_val_score
...: score = np.mean(cross_val_score(lr, trainX, trainY, scoring='accuracy',
cv=crossvalidation, n_jobs=1))
...: print ('The score of the 10 fold run is: ',score)
The score of the 10 fold run is: 0.7703958691910499
In [17]:
IPython console
History
LSP Python: ready
conda: base (Python 3.9.12)
Line 106, Col 50
UTF-8
CRLF
RW
Mem 93%

.

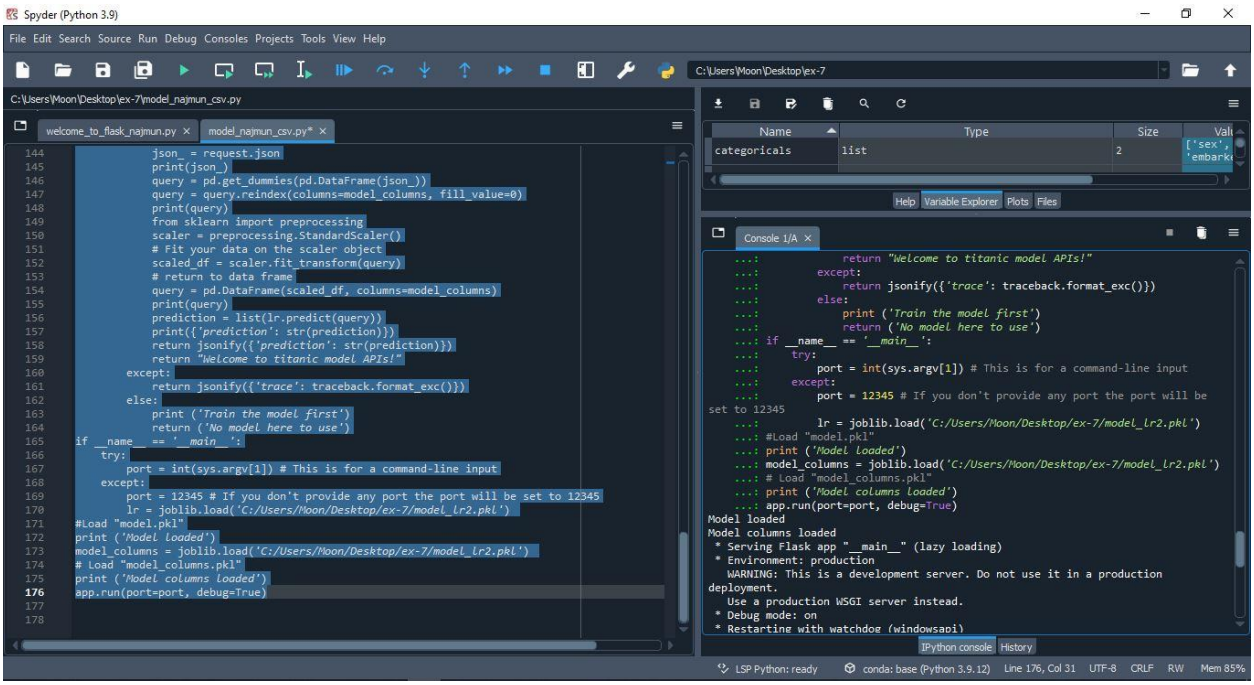

Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Moon\Desktop\ex-7
C:\Users\Moon\Desktop\ex-7\untitled1.py
welcome_to_flask_najmun.py
untitled1.py*
x = scaled_df[scaled_df.columns.difference([dependent_variable])]
x.dtypes
y = scaled_df[dependent_variable]
#convert the class back into integer
y = y.astype(int)
# Split the data into train test
from sklearn.model_selection import train_test_split
trainX,testX,trainY,testY = train_test_split(x,y, test_size = 0.2)
#build the model
lr = LogisticRegression(solver='lbfgs')
lr.fit(x, y)
# Score the model using 10 fold cross validation
from sklearn.model_selection import KFold
crossvalidation = KFold(n_splits=10, shuffle=True, random_state=1)
from sklearn.model_selection import cross_val_score
score = np.mean(cross_val_score(lr, trainX, trainY, scoring='accuracy', cv=crossvalidation,
print ('The score of the 10 fold run is: ',score)
testY_predict = lr.predict(testX)
testY_predict.dtype
#print(testY_predict)
#Import scikit-learn metrics module for accuracy calculation
from sklearn import metrics
labels = y.unique()
print(labels)
print("Accuracy:",metrics.accuracy_score(testY, testY_predict))
#Let us print the confusion matrix
from sklearn.metrics import confusion_matrix
print("Confusion matrix \n" , confusion_matrix(testY, testY_predict))
Name
Type
Size
categoricals
list
2
Help
Variable Explorer
Plots
Files
Console 1/A
Input In [17] in <cell line: 11>
print("Confusion matrix \n" , confusion_matrix(testY, testY_predict, labels))
TypeError: confusion_matrix() takes 2 positional arguments but 3 were given
In [18]: testY_predict = lr.predict(testX)
...: testY_predict.dtype
...: #print(testY_predict)
...: #Import scikit-learn metrics module for accuracy calculation
...: from sklearn import metrics
...: labels = y.unique()
...: print(labels)
...: print("Accuracy:",metrics.accuracy_score(testY, testY_predict))
...: #Let us print the confusion matrix
...: from sklearn.metrics import confusion_matrix
...: print("Confusion matrix \n" , confusion_matrix(testY, testY_predict))
[1 0]
Accuracy: 0.819047619047619
Confusion matrix
[[100 11]
[27 72]]
In [19]:
IPython console
History
LSP Python: ready
conda: base (Python 3.9.12)
Line 120, Col 70
UTF-8
CRLF
RW
Mem 89%


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Moon\Desktop\ex-7
C:\Users\Moon\Desktop\ex-7\untitled1.py
welcome_to_flask_najmun.py
untitled1.py*
#convert the class back into integer
y = y.astype(int)
# Split the data into train test
from sklearn.model_selection import train_test_split
trainX,testX,trainY,testY = train_test_split(x,y, test_size = 0.2)
#build the model
lr = LogisticRegression(solver='lbfgs')
lr.fit(x, y)
# Score the model using 10 fold cross validation
from sklearn.model_selection import KFold
crossvalidation = KFold(n_splits=10, shuffle=True, random_state=1)
from sklearn.model_selection import cross_val_score
score = np.mean(cross_val_score(lr, trainX, trainY, scoring='accuracy', cv=crossvalidation,
print ('The score of the 10 fold run is: ',score)
testY_predict = lr.predict(testX)
testY_predict.dtype
#print(testY_predict)
#Import scikit-learn metrics module for accuracy calculation
from sklearn import metrics
labels = y.unique()
print(labels)
print("Accuracy:",metrics.accuracy_score(testY, testY_predict))
#Let us print the confusion matrix
from sklearn.metrics import confusion_matrix
print("Confusion matrix \n" , confusion_matrix(testY, testY_predict))
import joblib
joblib.dump(lr, 'C:/Users/Moon/Desktop/ex-7/model_lr2.pkl')
print("Model dumped!")
Name
Type
Size
categoricals
list
2
Help
Variable Explorer
Plots
Files
Console 1/A
TypeError: confusion_matrix() takes 2 positional arguments but 3 were given
In [18]: testY_predict = lr.predict(testX)
...: testY_predict.dtype
...: #print(testY_predict)
...: #Import scikit-learn metrics module for accuracy calculation
...: from sklearn import metrics
...: labels = y.unique()
...: print(labels)
...: print("Accuracy:",metrics.accuracy_score(testY, testY_predict))
...: #Let us print the confusion matrix
...: from sklearn.metrics import confusion_matrix
...: print("Confusion matrix \n" , confusion_matrix(testY, testY_predict))
[1 0]
Accuracy: 0.819047619047619
Confusion matrix
[[100 11]
[27 72]]
In [19]: import joblib
...: joblib.dump(lr, 'C:/Users/Moon/Desktop/ex-7/model_lr2.pkl')
...: print("Model dumped!")
Model dumped!
In [20]:
IPython console
History
LSP Python: ready
conda: base (Python 3.9.12)
Line 125, Col 23
UTF-8
CRLF
RW
Mem 86%

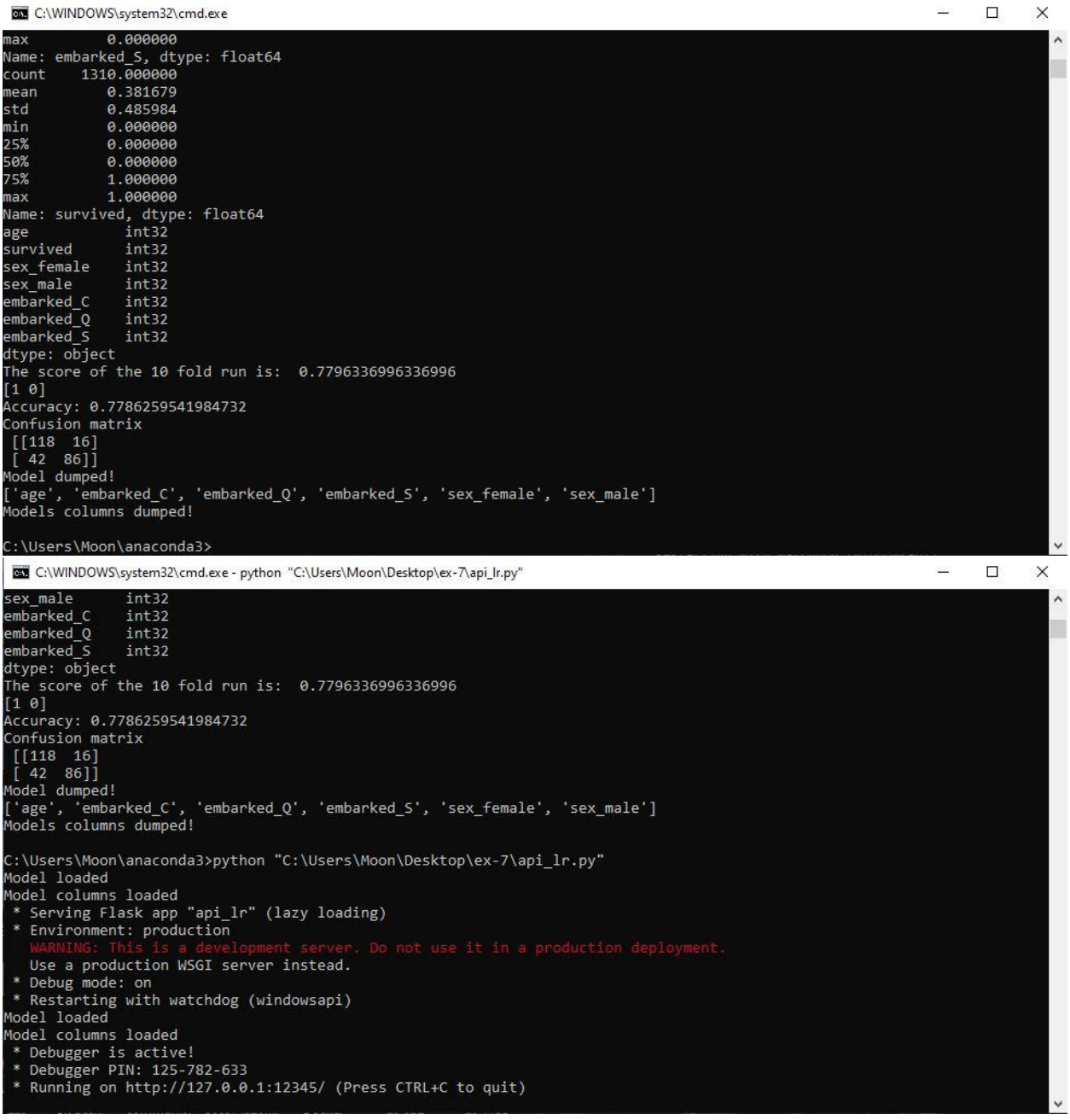

Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Moon\Desktop\ex-7
C:\Users\Moon\Desktop\ex-7\model_najmun_csv.py
welcome_to_flask_najmun.py
model_najmun_csv.py*
#build the model
lr = LogisticRegression(solver='lbfgs')
lr.fit(x, y)
# Score the model using 10 fold cross validation
from sklearn.model_selection import KFold
crossvalidation = KFold(n_splits=10, shuffle=True, random_state=1)
from sklearn.model_selection import cross_val_score
score = np.mean(cross_val_score(lr, trainX, trainY, scoring='accuracy', cv=crossvalidation,
print ('The score of the 10 fold run is: ',score)
testY_predict = lr.predict(testX)
testY_predict.dtype
#print(testY_predict)
#Import scikit-learn metrics module for accuracy calculation
from sklearn import metrics
labels = y.unique()
print(labels)
print("Accuracy:",metrics.accuracy_score(testY, testY_predict))
#Let us print the confusion matrix
from sklearn.metrics import confusion_matrix
print("Confusion matrix \n" , confusion_matrix(testY, testY_predict))
import joblib
joblib.dump(lr, 'C:/Users/Moon/Desktop/ex-7/model_lr2.pkl')
print("Model dumped!")
model_columns = list(x.columns)
print(model_columns)
joblib.dump(model_columns, 'C:/Users/Moon/Desktop/ex-7/model_columns.pkl')
print("Models columns dumped!")
Name
Type
Size
categoricals
list
2
Help
Variable Explorer
Plots
Files
Console 1/A
In [22]: model_columns = list(x.columns)
...: print(model_columns)
...: joblib.dump(model_columns, 'C:/Users/Moon/Desktop/ex-7/
model_columns.pkl')
...: print("Models columns dumped!")
['age', 'embarked_C', 'embarked_Q', 'embarked_S', 'sex_female', 'sex_male']
Models columns dumped!
In [23]:
IPython console
History
LSP Python: ready
conda: base (Python 3.9.12)
Line 130, Col 32
UTF-8
CRLF
RW
Mem 83%


ex-7
File
Home
Share
View
ex-7
Search ex-7
Quick access
Desktop
Downloads
Documents
Pictures
ConsumeAPIAp
ex-7
images
videos
OneDrive
This PC
3D Objects
Desktop
Documents
Advertising
bank
dtree2
helloworldnajmun
iris1
model_columns.pkl
model_lr2.pkl
model_najmun_csv
titanic
welcome_to_flask_najmun
wine
11 items
2 items selected 4.61 KB

Exercise #5


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Moon\Desktop\ex-7
C:\Users\Moon\Desktop\ex-7\model_najmun_csv.py
welcome_to_flask_najmun.py
model_najmun_csv.py*
json_ = request.json
print(json_)
query = pd.get_dummies(pd.DataFrame(json_))
query = query.reindex(columns=model_columns, fill_value=0)
print(query)
from sklearn import preprocessing
scaler = preprocessing.StandardScaler()
# Fit your data on the scaler object
scaled_df = scaler.fit_transform(query)
# return to data frame
query = pd.DataFrame(scaled_df, columns=model_columns)
print(query)
prediction = list(lr.predict(query))
print({'prediction': str(prediction)})
return jsonify({'prediction': str(prediction)})
return "Welcome to titanic model APIs!"
except:
return jsonify({'trace': traceback.format_exc()})
else:
print ('Train the model first')
return ('No model here to use')
if __name__ == '__main__':
try:
port = int(sys.argv[1]) # This is for a command-line input
except:
port = 12345 # If you don't provide any port the port will be set to 12345
lr = joblib.load('C:/Users/Moon/Desktop/ex-7/model_lr2.pkl')
#Load "model.pkl"
print ('Model loaded')
model_columns = joblib.load('C:/Users/Moon/Desktop/ex-7/model_lr2.pkl')
# Load "model_columns.pkl"
print ('Model columns loaded')
app.run(port=port, debug=True)
Name
Type
Size
categoricals
list
2
Help
Variable Explorer
Plots
Files
Console 1/A
Model loaded
Model columns loaded
* Serving Flask app "__main__" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: on
* Restarting with watchdog (windowsapi)
IPython console
History
LSP Python: ready
conda: base (Python 3.9.12)
Line 176, Col 31
UTF-8
CRLF
RW
Mem 85%

Exercise #6


C:\WINDOWS\system32\cmd.exe
max 0.000000
Name: embarked_S, dtype: float64
count 1310.000000
mean 0.381679
std 0.485984
min 0.000000
25% 0.000000
50% 0.000000
75% 1.000000
max 1.000000
Name: survived, dtype: float64
age int32
survived int32
sex_female int32
sex_male int32
embarked_C int32
embarked_Q int32
embarked_S int32
dtype: object
The score of the 10 fold run is: 0.7796336996336996
[1 0]
Accuracy: 0.7786259541984732
Confusion matrix
[[118 16]
[42 86]]
Model dumped!
['age', 'embarked_C', 'embarked_Q', 'embarked_S', 'sex_female', 'sex_male']
Models columns dumped!
C:\Users\Moon\anaconda3>
C:\WINDOWS\system32\cmd.exe - python "C:\Users\Moon\Desktop\ex-7\api_lr.py"
sex_male int32
embarked_C int32
embarked_Q int32
embarked_S int32
dtype: object
The score of the 10 fold run is: 0.7796336996336996
[1 0]
Accuracy: 0.7786259541984732
Confusion matrix
[[118 16]
[42 86]]
Model dumped!
['age', 'embarked_C', 'embarked_Q', 'embarked_S', 'sex_female', 'sex_male']
Models columns dumped!
C:\Users\Moon\anaconda3>python "C:\Users\Moon\Desktop\ex-7\api_lr.py"
Model loaded
Model columns loaded
* Serving Flask app "api_lr" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: on
* Restarting with watchdog (windowsapi)
Model loaded
Model columns loaded
* Debugger is active!
* Debugger PIN: 125-782-633
* Running on http://127.0.0.1:12345/ (Press CTRL+C to quit)