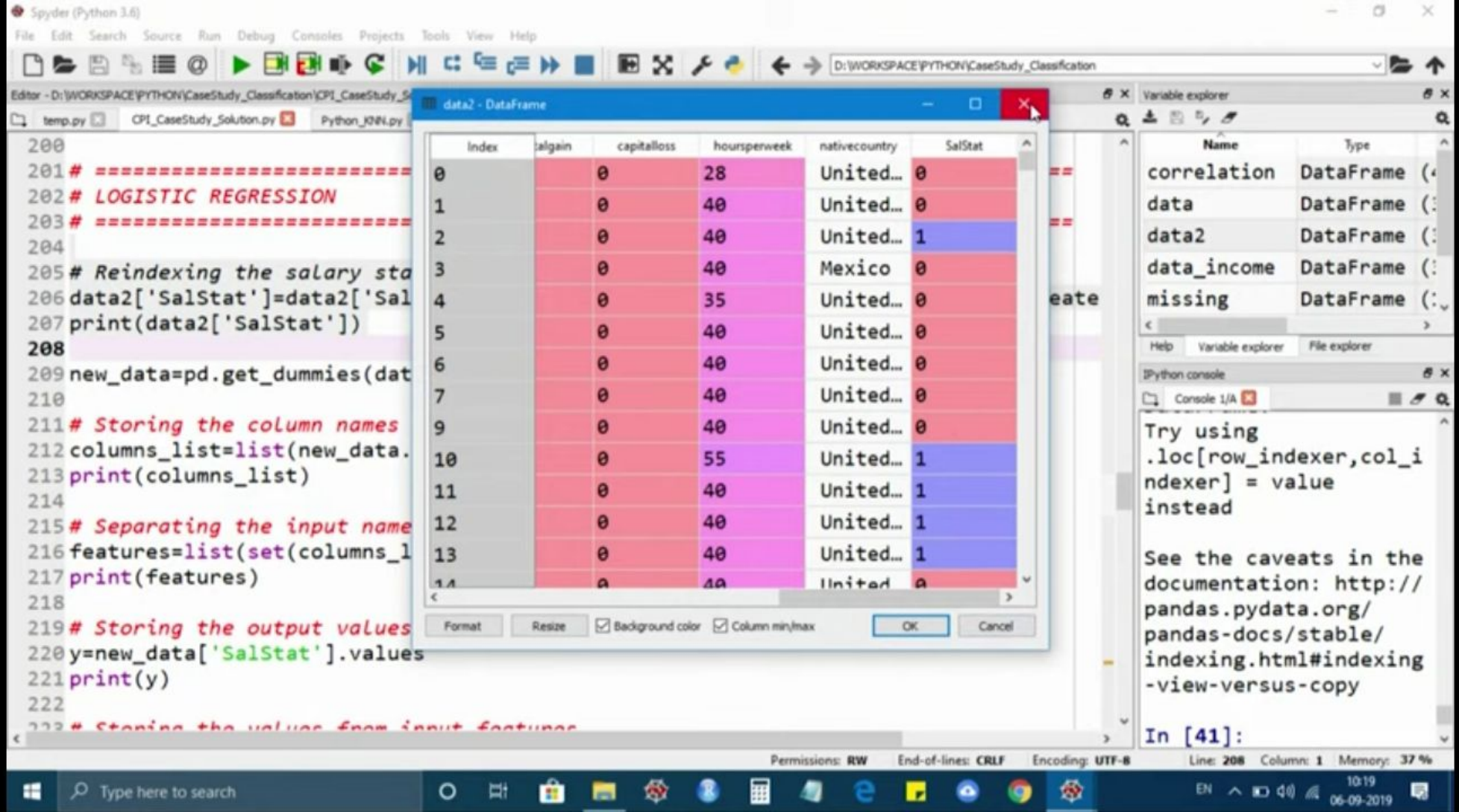
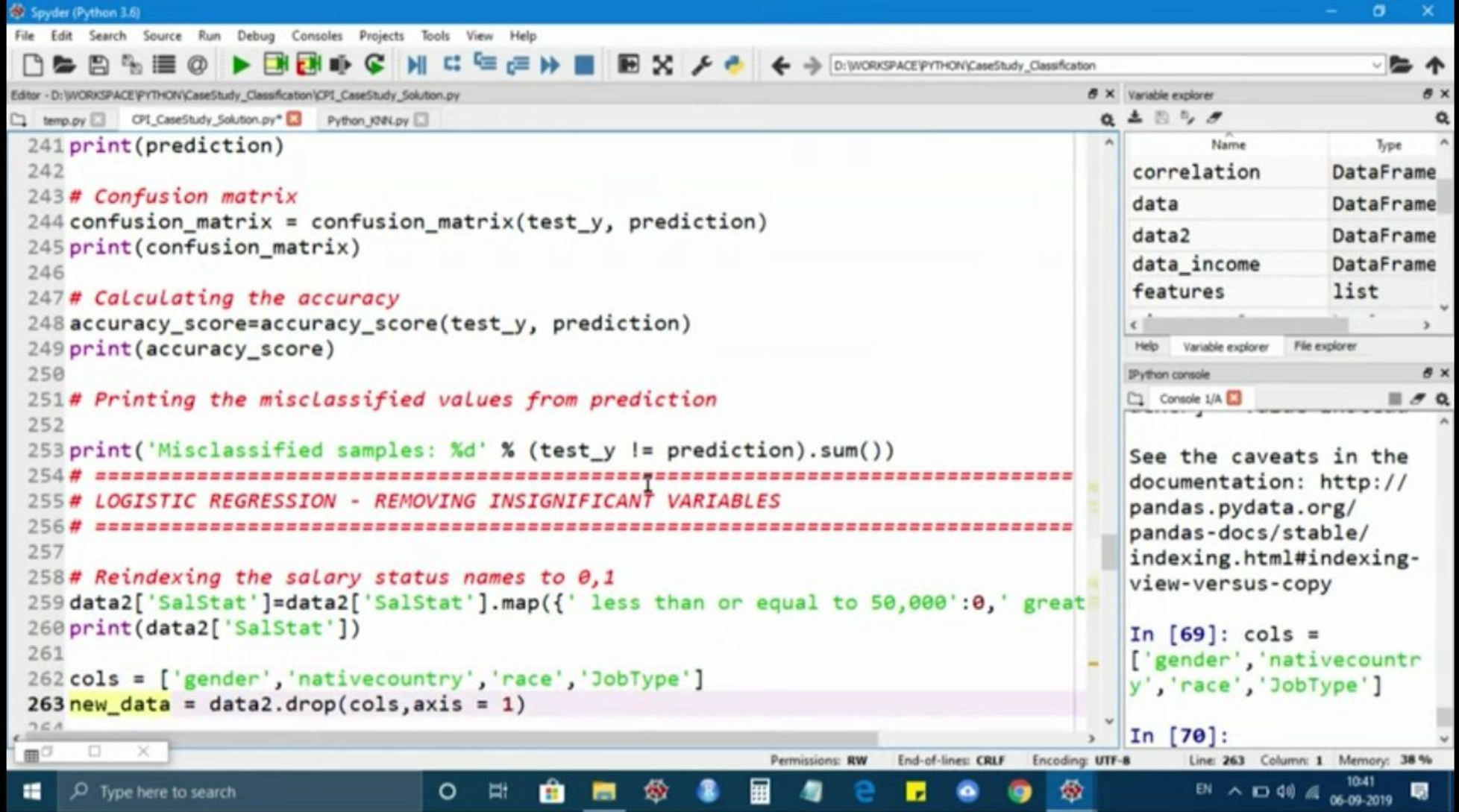
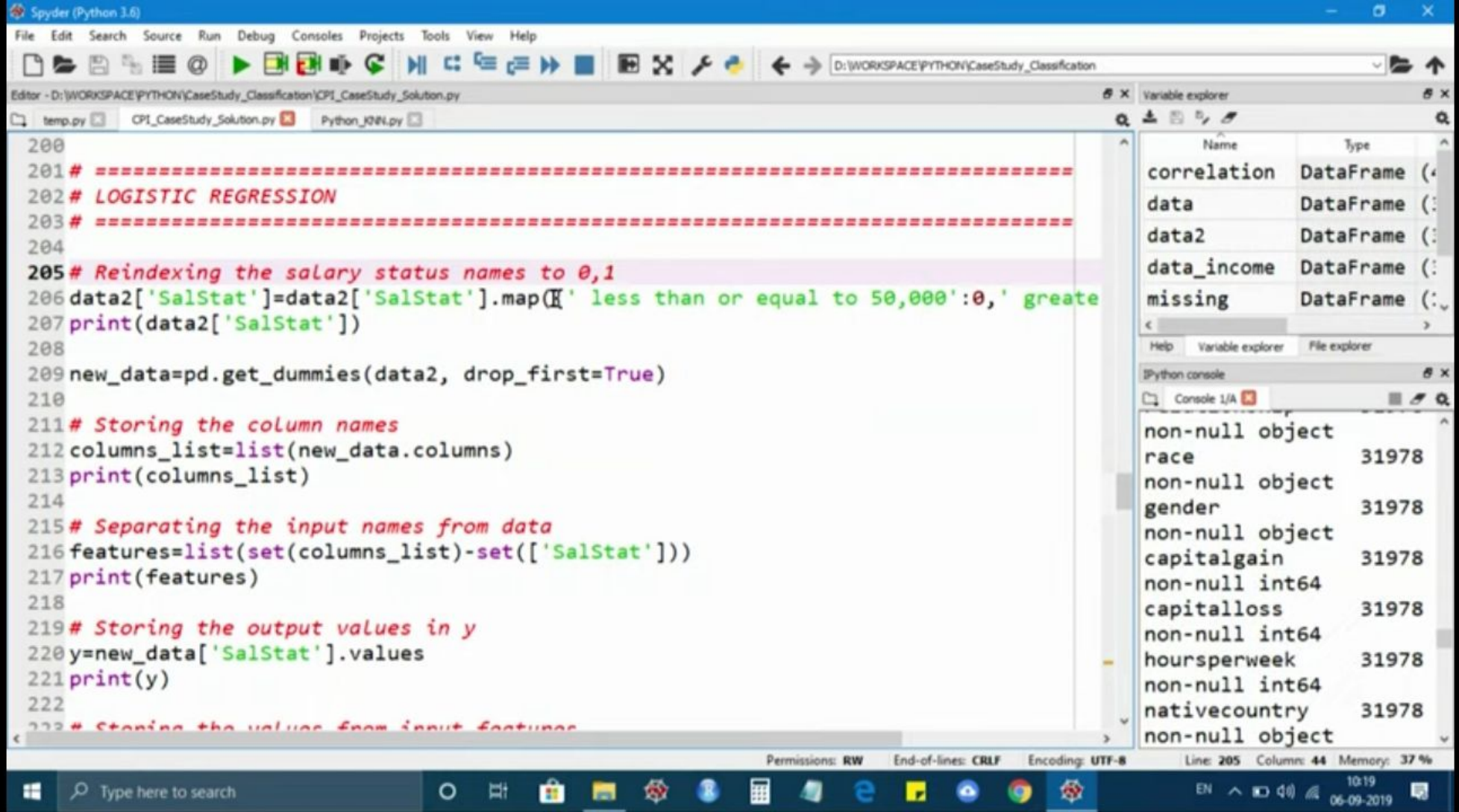
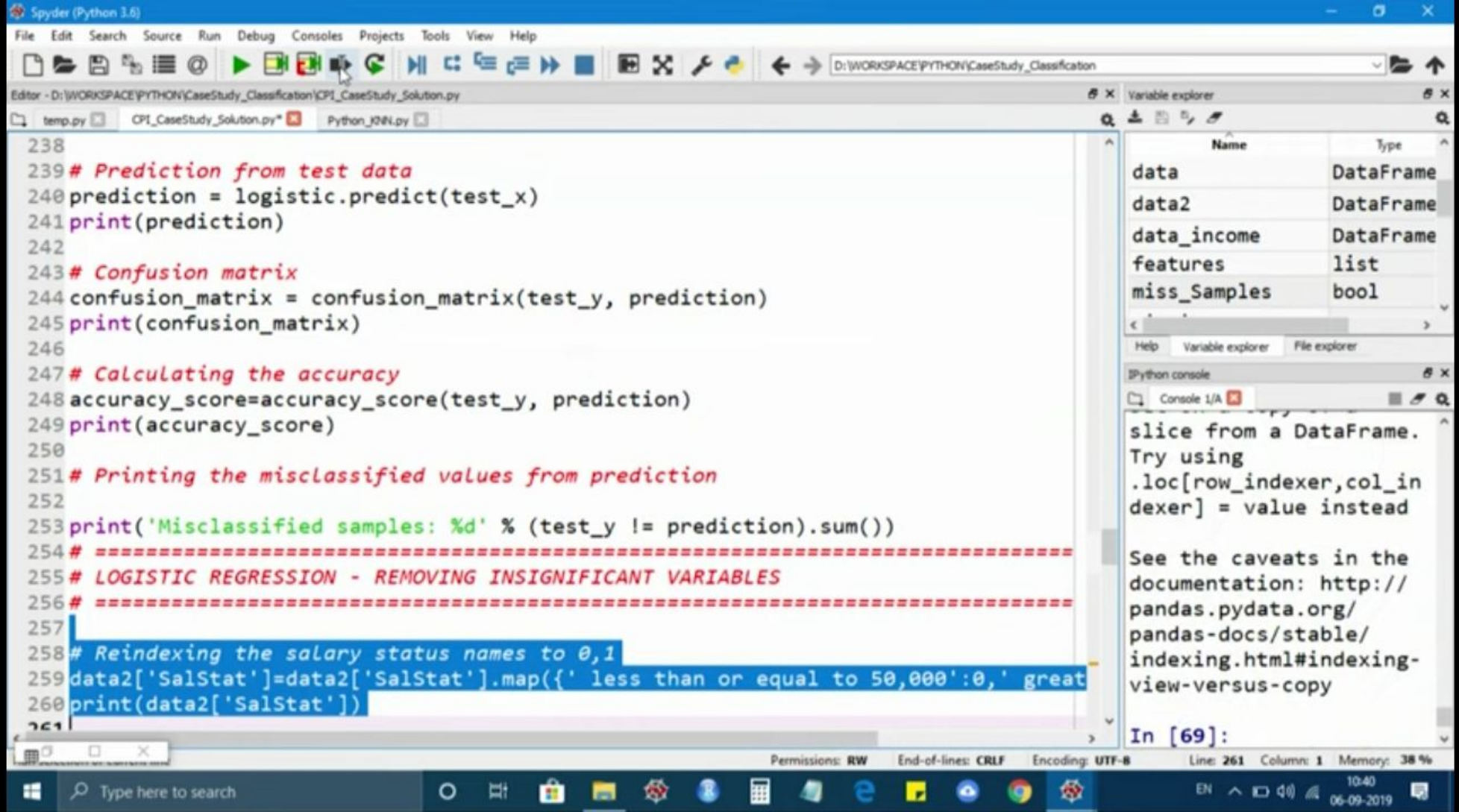
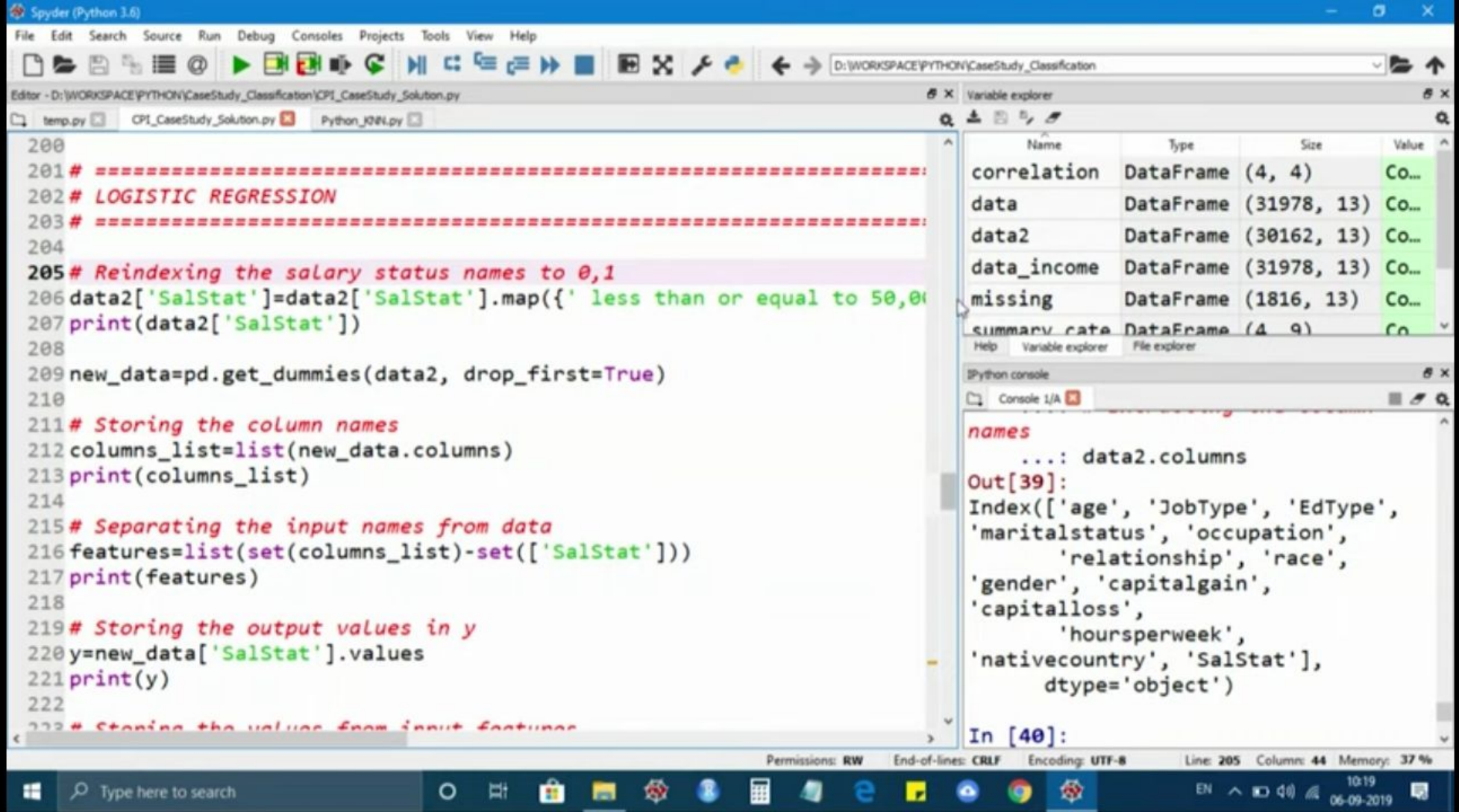
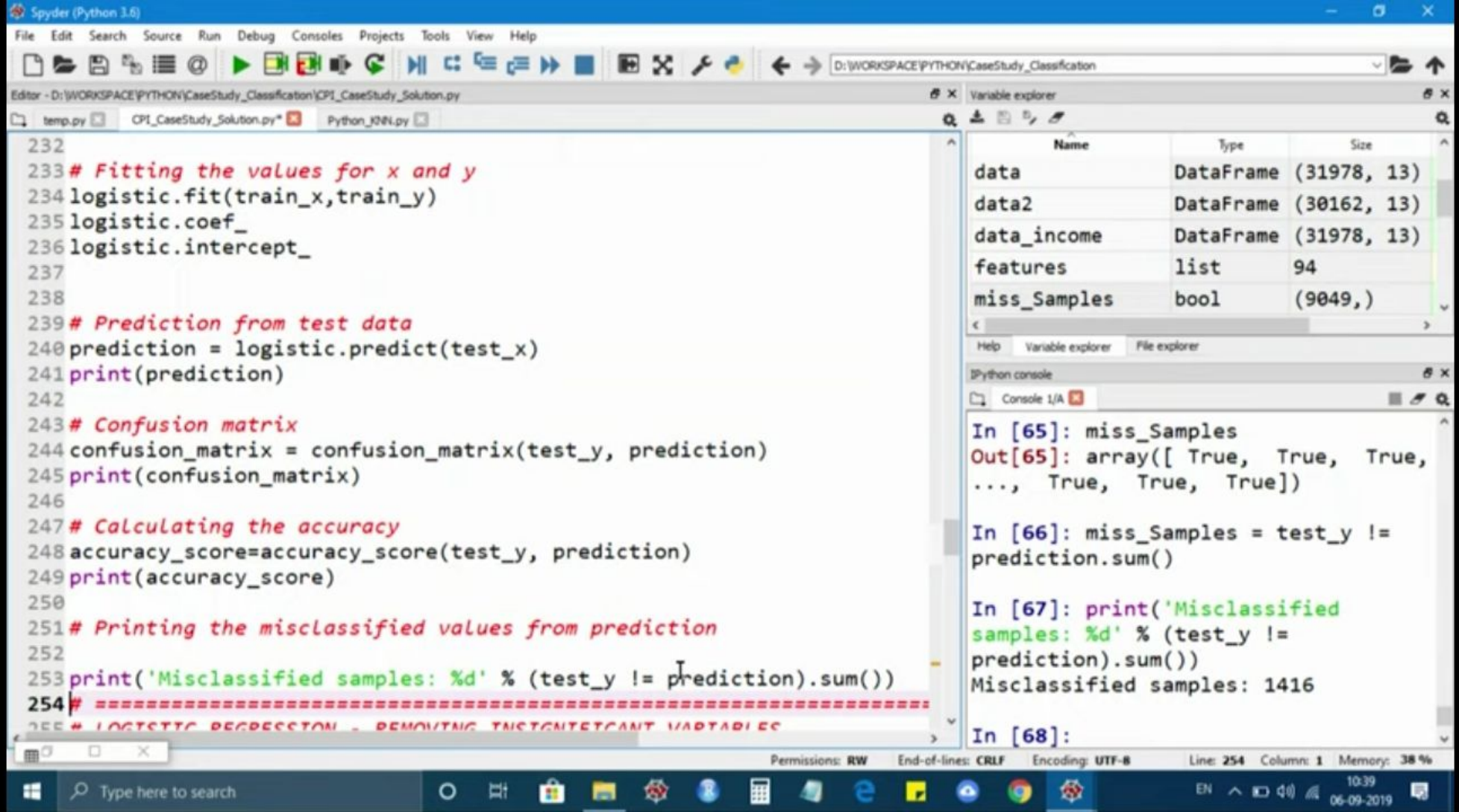

Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\WORKSPACE\PYTHON\CaseStudy_Classification
Editor - D:\WORKSPACE\PYTHON\CaseStudy_Classification\CPI_CaseStudy_S
temp.py
CPI_CaseStudy_Solution.py
Python_KNN.py
200
201 # ==========
202 # LOGISTIC REGRESSION
203 # ==========
204
205 # Reindexing the salary sta
206 data2['SalStat']=data2['Sal
207 print(data2['SalStat'])
208
209 new_data=pd.get_dummies(dat
210
211 # Storing the column names
212 columns_list=list(new_data.
213 print(columns_list)
214
215 # Separating the input name
216 features=list(set(columns_l
217 print(features)
218
219 # Storing the output values
220 y=new_data['SalStat'].values
221 print(y)
222
data2 - DataFrame
Index
:algain
capitalloss
hoursperweek
nativecountry
SalStat
0 0 28 United... 0
1 0 40 United... 0
2 0 40 United... 1
3 0 40 Mexico 0
4 0 35 United... 0
5 0 40 United... 0
6 0 40 United... 0
7 0 40 United... 0
9 0 40 United... 0
10 0 55 United... 1
11 0 40 United... 1
12 0 40 United... 1
13 0 40 United... 1
Format
Resize
Background color
Column min/max
OK
Cancel
Variable explorer
Name
Type
correlation DataFrame
data DataFrame
data2 DataFrame
data_income DataFrame
missing DataFrame
Help
Variable explorer
File explorer
IPython console
Console 1/A
Try using
.loc[row_indexer,col_i
ndexer] = value
instead
See the caveats in the
documentation: http://
pandas.pydata.org/
pandas-docs/stable/
indexing.html#indexing
-view-versus-copy
In [41]:
Permissions: RW
End-of-lines: CRLF
Encoding: UTF-8
Line: 208
Column: 1
Memory: 37 %
Type here to search
EN
10:19
06-09-2019

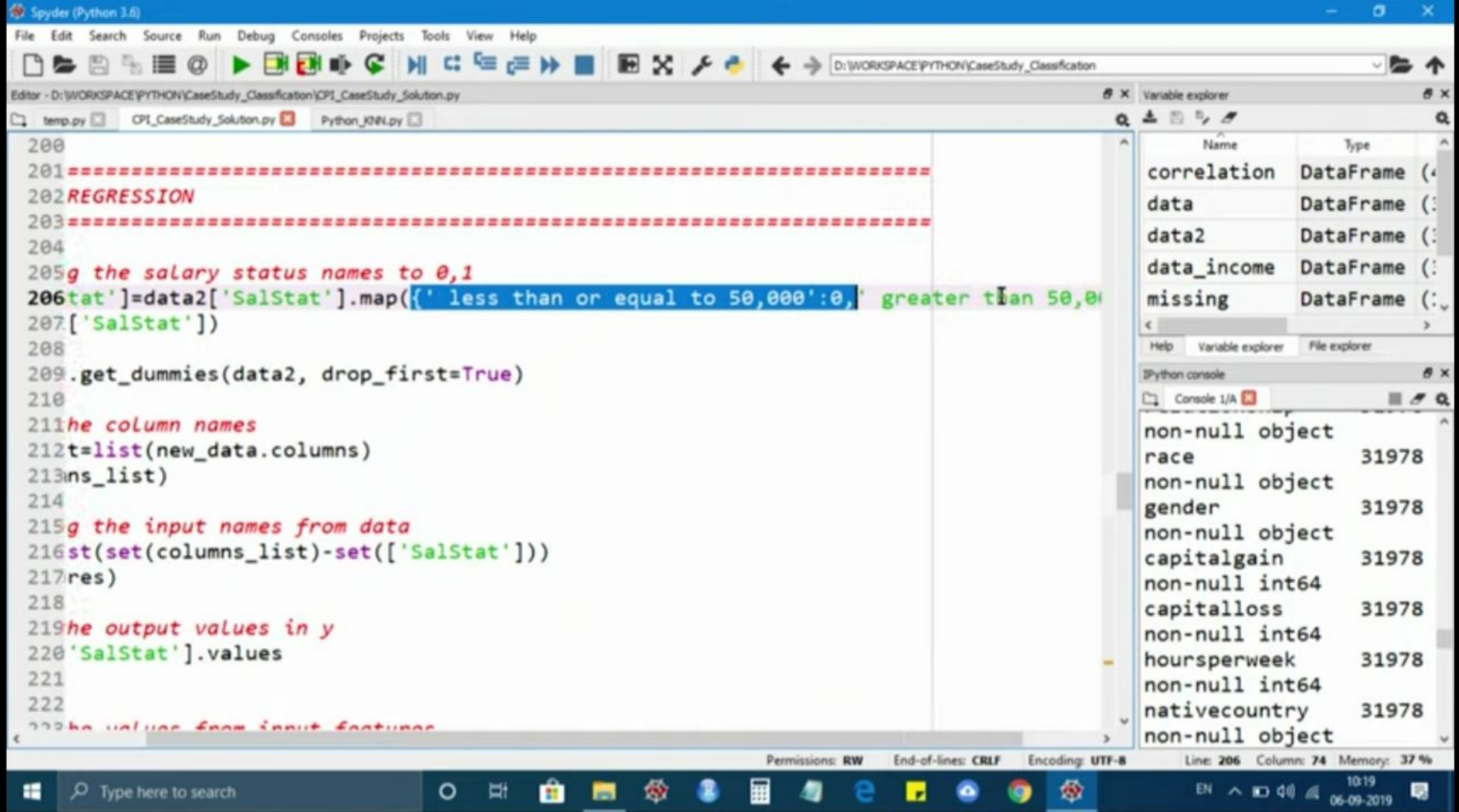

Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\WORKSPACE\PYTHON\CaseStudy_Classification
Editor - D:\WORKSPACE\PYTHON\CaseStudy_Classification\CPI_CaseStudy_Solution.py
temp.py
CPI_CaseStudy_Solution.py*
Python_KNN.py
241 print(prediction)
242
243 # Confusion matrix
244 confusion_matrix = confusion_matrix(test_y, prediction)
245 print(confusion_matrix)
246
247 # Calculating the accuracy
248 accuracy_score=accuracy_score(test_y, prediction)
249 print(accuracy_score)
250
251 # Printing the misclassified values from prediction
252
253 print('Misclassified samples: %d' % (test_y != prediction).sum())
254 # ==
255 # LOGISTIC REGRESSION - REMOVING INSIGNIFICANT VARIABLES
256 # ==
257
258 # Reindexing the salary status names to 0,1
259 data2['SalStat']=data2['SalStat'].map({' less than or equal to 50,000':0,' great
260 print(data2['SalStat'])
261
262 cols = ['gender','nativecountry','race','JobType']
263 new_data = data2.drop(cols,axis = 1)
Variable explorer
Name Type
correlation DataFrame
data DataFrame
data2 DataFrame
data_income DataFrame
features list
Help Variable explorer File explorer
IPython console
Console 1/A
See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy
In [69]: cols = ['gender','nativecountry','race','JobType']
In [70]:
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 263 Column: 1 Memory: 38 %
Type here to search
EN 10:41 06-09-2019

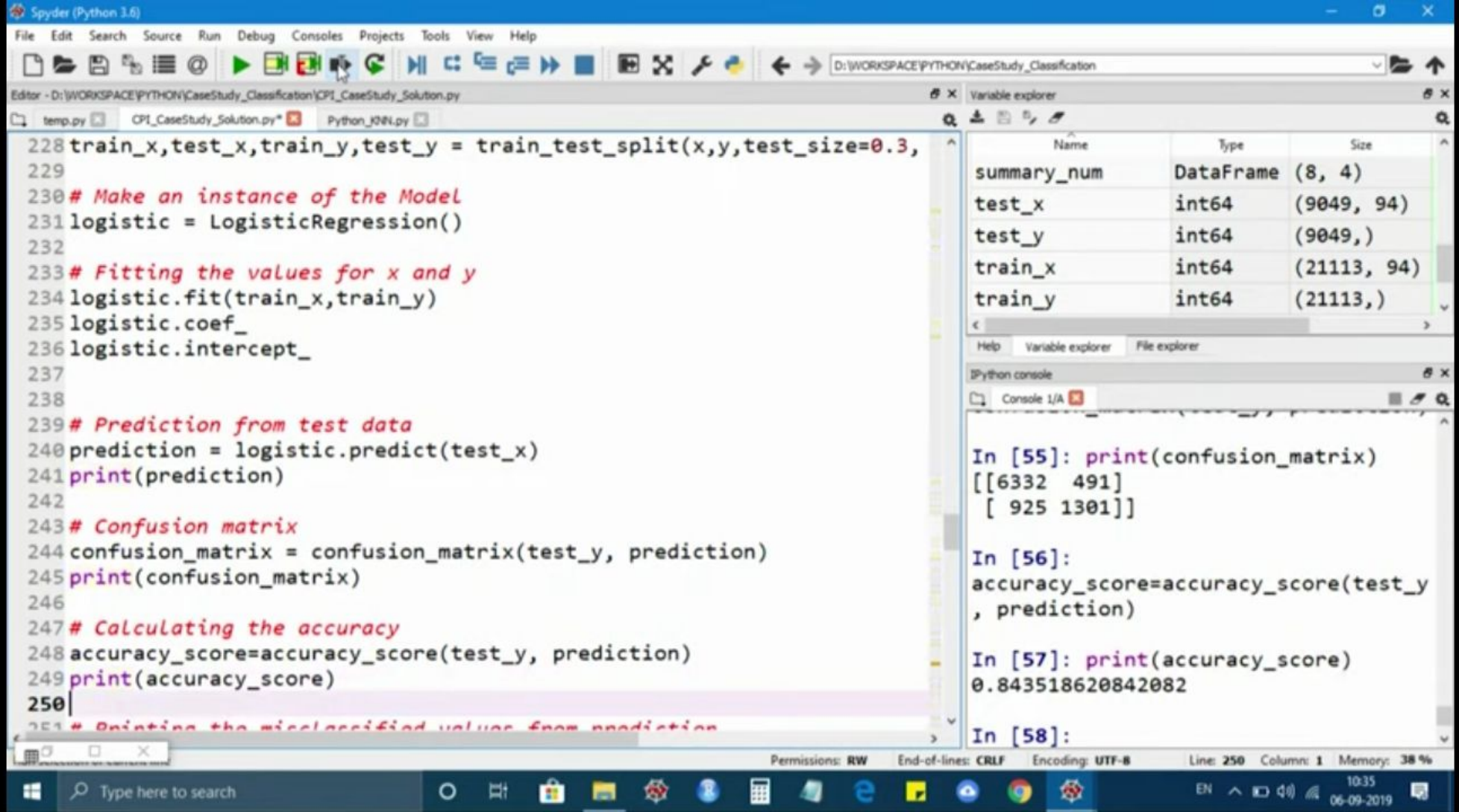

Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\WORKSPACE\PYTHON\CaseStudy_Classification
Editor - D:\WORKSPACE\PYTHON\CaseStudy_Classification\CPI_CaseStudy_Solution.py
temp.py
CPI_CaseStudy_Solution.py
Python_KNN.py
200
201 # ==========
202 # LOGISTIC REGRESSION
203 # ==========
204
205 # Reindexing the salary status names to 0,1
206 data2['SalStat']=data2['SalStat'].map({' less than or equal to 50,000':0,' greate
207 print(data2['SalStat'])
208
209 new_data=pd.get_dummies(data2, drop_first=True)
210
211 # Storing the column names
212 columns_list=list(new_data.columns)
213 print(columns_list)
214
215 # Separating the input names from data
216 features=list(set(columns_list)-set(['SalStat']))
217 print(features)
218
219 # Storing the output values in y
220 y=new_data['SalStat'].values
221 print(y)
222
Variable explorer
Name Type
correlation DataFrame
data DataFrame
data2 DataFrame
data_income DataFrame
missing DataFrame
Help Variable explorer File explorer
IPython console
Console 1/A
non-null object
race 31978
non-null object
gender 31978
non-null object
capitalgain 31978
non-null int64
capitalloss 31978
non-null int64
hoursperweek 31978
non-null int64
nativecountry 31978
non-null object
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 205 Column: 44 Memory: 37 %
Type here to search
EN
10:19
06-09-2019

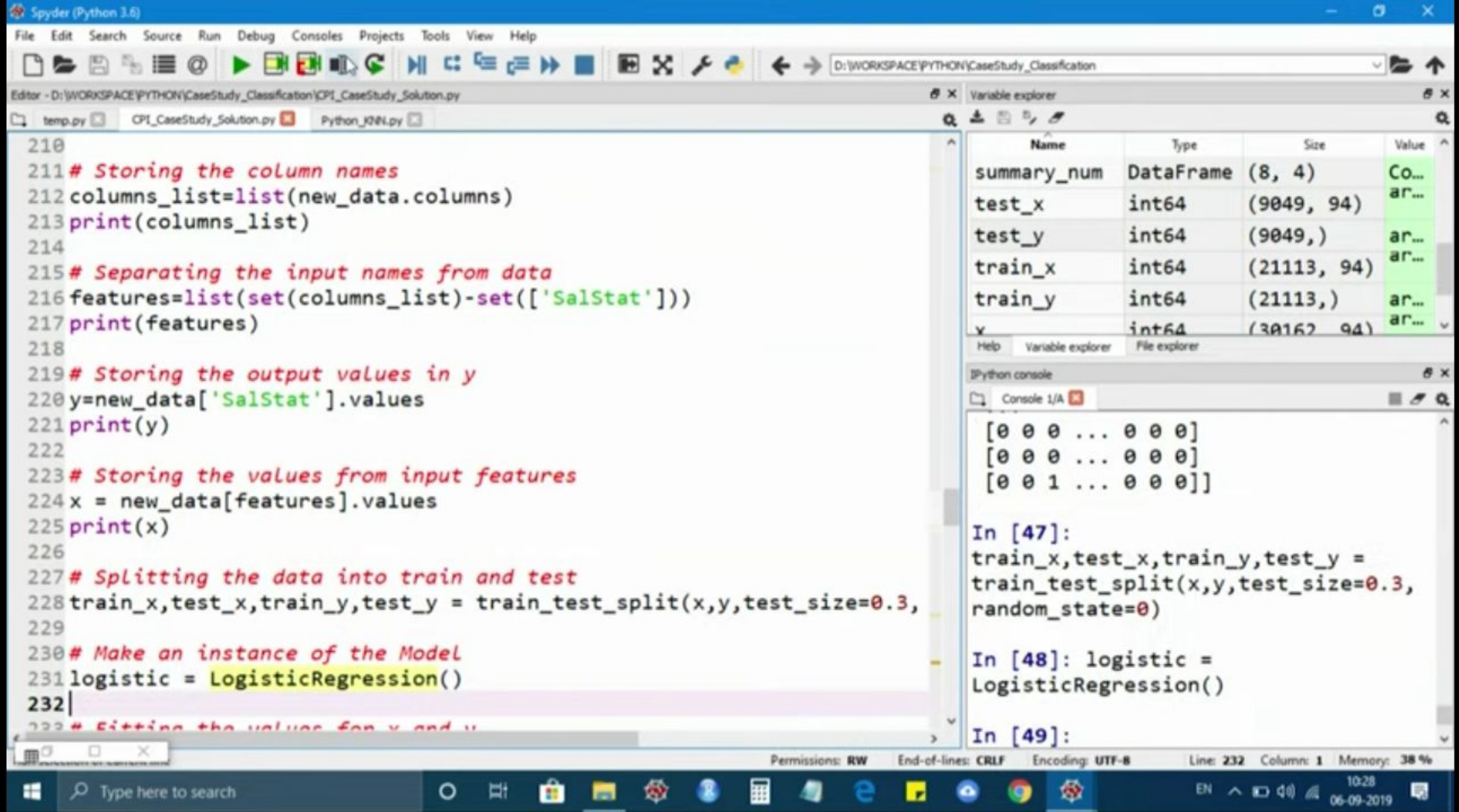

Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\WORKSPACE\PYTHON\CaseStudy_Classification
Editor - D:\WORKSPACE\PYTHON\CaseStudy_Classification\CPI_CaseStudy_Solution.py
temp.py
CPI_CaseStudy_Solution.py*
Python_KNN.py
238
239 # Prediction from test data
240 prediction = logistic.predict(test_x)
241 print(prediction)
242
243 # Confusion matrix
244 confusion_matrix = confusion_matrix(test_y, prediction)
245 print(confusion_matrix)
246
247 # Calculating the accuracy
248 accuracy_score=accuracy_score(test_y, prediction)
249 print(accuracy_score)
250
251 # Printing the misclassified values from prediction
252
253 print('Misclassified samples: %d' % (test_y != prediction).sum())
254 # ==========
255 # LOGISTIC REGRESSION - REMOVING INSIGNIFICANT VARIABLES
256 # ==========
257
258 # Reindexing the salary status names to 0,1
259 data2['SalStat']=data2['SalStat'].map({' less than or equal to 50,000':0,' great
260 print(data2['SalStat'])
Variable explorer
Name Type
data DataFrame
data2 DataFrame
data_income DataFrame
features list
miss_Samples bool
Help Variable explorer File explorer
IPython console
Console 1/A
slice from a DataFrame.
Try using
.loc[row_indexer,col_in
dexer] = value instead
See the caveats in the
documentation: http://
pandas.pydata.org/
pandas-docs/stable/
indexing.html#indexing-
view-versus-copy
In [69]:
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 261 Column: 1 Memory: 38 %
Type here to search
EN
10:40
06-09-2019

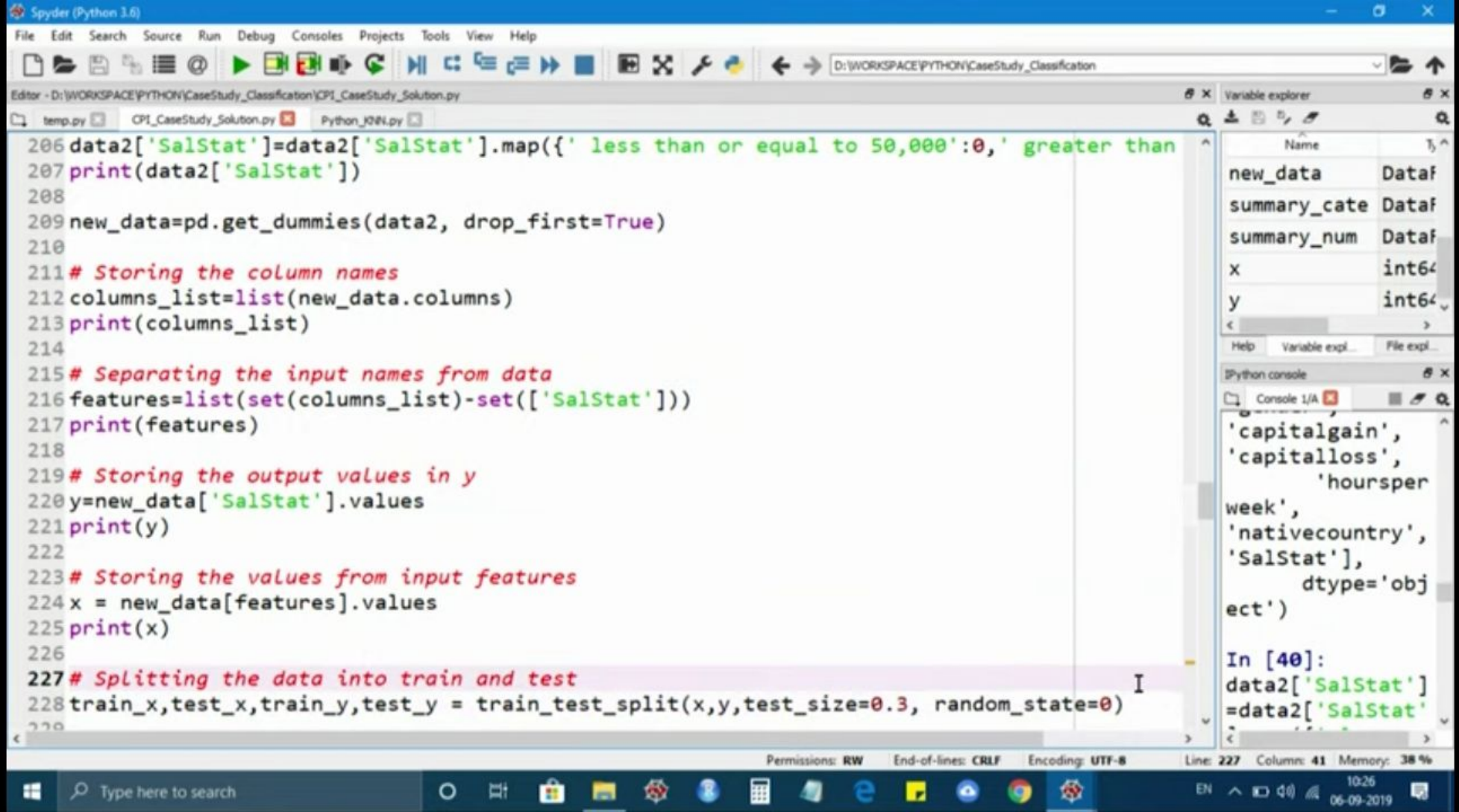

Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\WORKSPACE\PYTHON\CaseStudy_Classification
Editor - D:\WORKSPACE\PYTHON\CaseStudy_Classification\CPI_CaseStudy_Solution.py
temp.py
CPI_CaseStudy_Solution.py
Python_KNN.py
200
201 # ==========
202 # LOGISTIC REGRESSION
203 # ==========
204
205 # Reindexing the salary status names to 0,1
206 data2['SalStat']=data2['SalStat'].map({' less than or equal to 50,0
207 print(data2['SalStat'])
208
209 new_data=pd.get_dummies(data2, drop_first=True)
210
211 # Storing the column names
212 columns_list=list(new_data.columns)
213 print(columns_list)
214
215 # Separating the input names from data
216 features=list(set(columns_list)-set(['SalStat']))
217 print(features)
218
219 # Storing the output values in y
220 y=new_data['SalStat'].values
221 print(y)
222
Variable explorer
Name Type Size Value
correlation DataFrame (4, 4) Co...
data DataFrame (31978, 13) Co...
data2 DataFrame (30162, 13) Co...
data_income DataFrame (31978, 13) Co...
missing DataFrame (1816, 13) Co...
summary_cate DataFrame (4, 9) Co
Help Variable explorer File explorer
IPython console
Console 1/A
names
...: data2.columns
Out[39]:
Index(['age', 'JobType', 'EdType',
'maritalstatus', 'occupation',
'relationship', 'race',
'gender', 'capitalgain',
'capitalloss',
'hoursperweek',
'nativecountry', 'SalStat'],
dtype='object')
In [40]:
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 205 Column: 44 Memory: 37 %
Type here to search
EN
10:19
06-09-2019

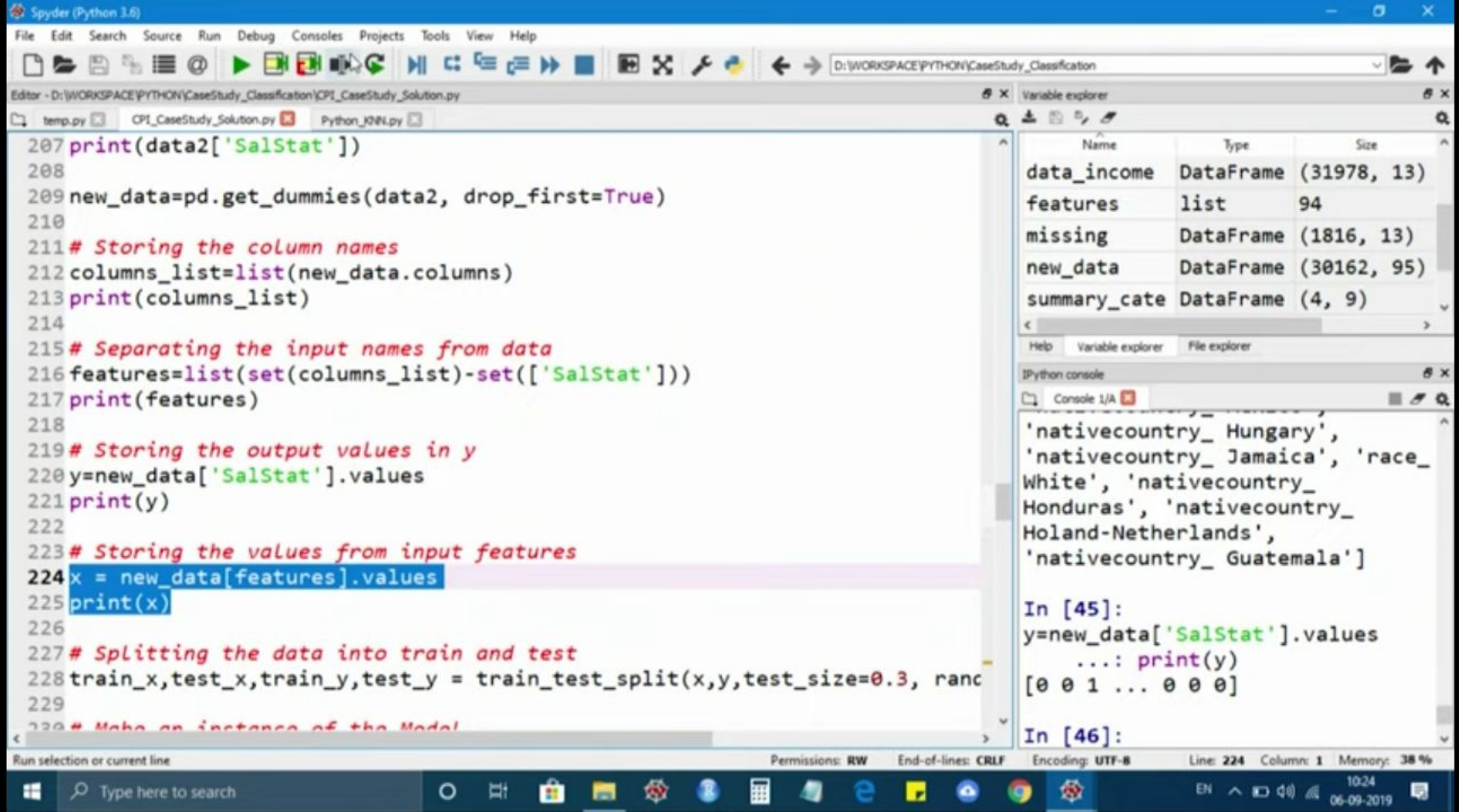

Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\WORKSPACE\PYTHON\CaseStudy_Classification
Editor - D:\WORKSPACE\PYTHON\CaseStudy_Classification\CPI_CaseStudy_Solution.py
temp.py
CPI_CaseStudy_Solution.py*
Python_KNN.py
232
233 # Fitting the values for x and y
234 logistic.fit(train_x,train_y)
235 logistic.coef_
236 logistic.intercept_
237
238
239 # Prediction from test data
240 prediction = logistic.predict(test_x)
241 print(prediction)
242
243 # Confusion matrix
244 confusion_matrix = confusion_matrix(test_y, prediction)
245 print(confusion_matrix)
246
247 # Calculating the accuracy
248 accuracy_score=accuracy_score(test_y, prediction)
249 print(accuracy_score)
250
251 # Printing the misclassified values from prediction
252
253 print('Misclassified samples: %d' % (test_y != prediction).sum())
254 # ==
Variable explorer
Name Type Size
data DataFrame (31978, 13)
data2 DataFrame (30162, 13)
data_income DataFrame (31978, 13)
features list 94
miss_Samples bool (9049,)
Help Variable explorer File explorer
IPython console
Console 1/A
In [65]: miss_Samples
Out[65]: array([True, True, True,
..., True, True, True])
In [66]: miss_Samples = test_y !=
prediction.sum()
In [67]: print('Misclassified
samples: %d' % (test_y !=
prediction).sum())
Misclassified samples: 1416
In [68]:
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 254 Column: 1 Memory: 38 %
Type here to search
EN 10:39 06-09-2019

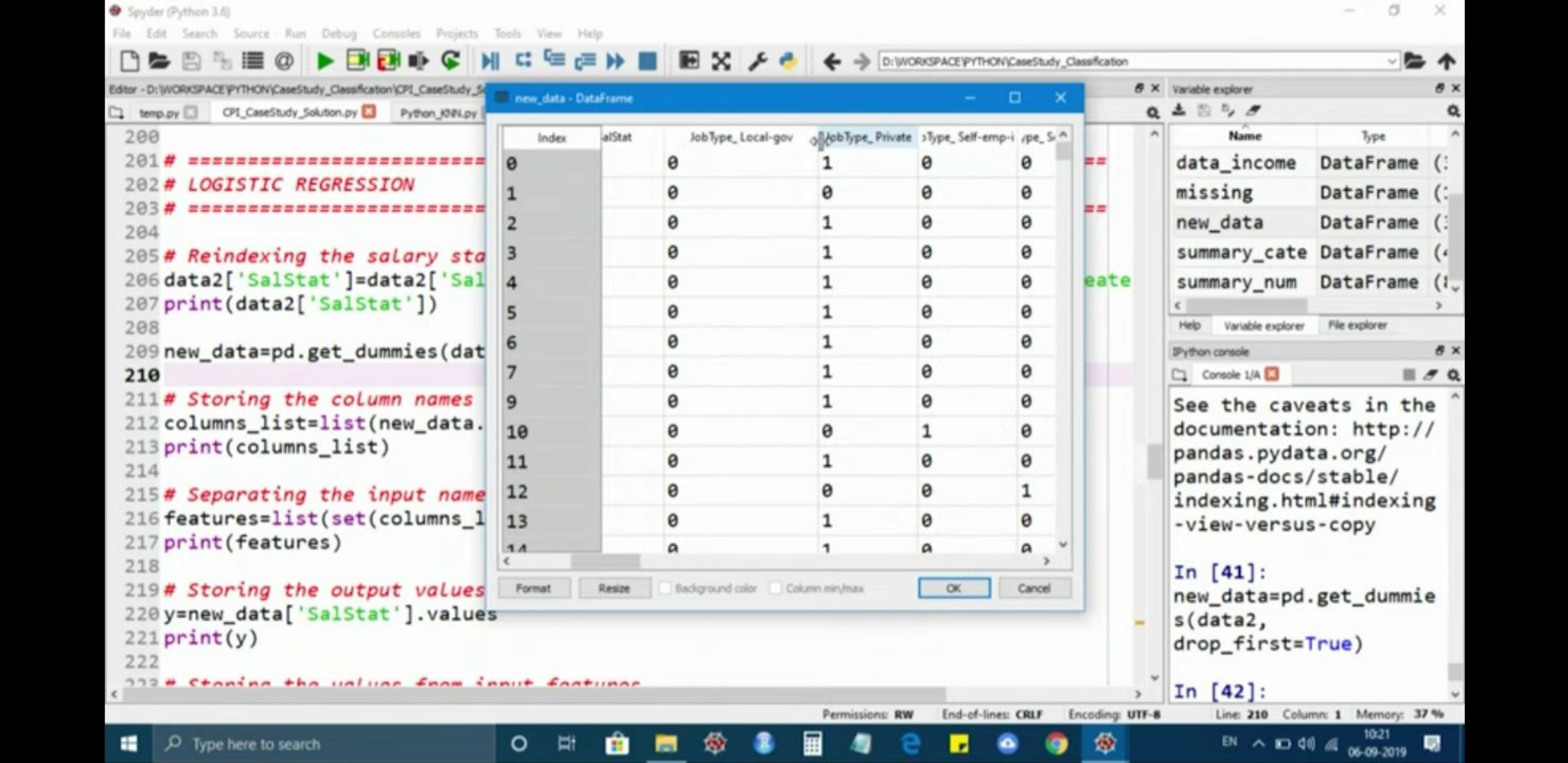

Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\WORKSPACE\PYTHON\CaseStudy_Classification
Editor - D:\WORKSPACE\PYTHON\CaseStudy_Classification\CPI_CaseStudy_Solution.py
temp.py
CPI_CaseStudy_Solution.py
Python_KNN.py
200
201 ==========
202 REGRESSION
203 ==========
204
205 g the salary status names to 0,1
206 tat']=data2['SalStat'].map({' less than or equal to 50,000':0,' greater than 50,0
207 ['SalStat'])
208
209 .get_dummies(data2, drop_first=True)
210
211 he column names
212 t=list(new_data.columns)
213 ns_list)
214
215 g the input names from data
216 st(set(columns_list)-set(['SalStat']))
217 res)
218
219 he output values in y
220 'SalStat'].values
221
222
Variable explorer
Name Type
correlation DataFrame
data DataFrame
data2 DataFrame
data_income DataFrame
missing DataFrame
Help Variable explorer File explorer
IPython console
Console 1/A
non-null object
race 31978
non-null object
gender 31978
non-null object
capitalgain 31978
non-null int64
capitalloss 31978
non-null int64
hoursperweek 31978
non-null int64
nativecountry 31978
non-null object
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 206 Column: 74 Memory: 37 %
Type here to search
EN
10:19
06-09-2019

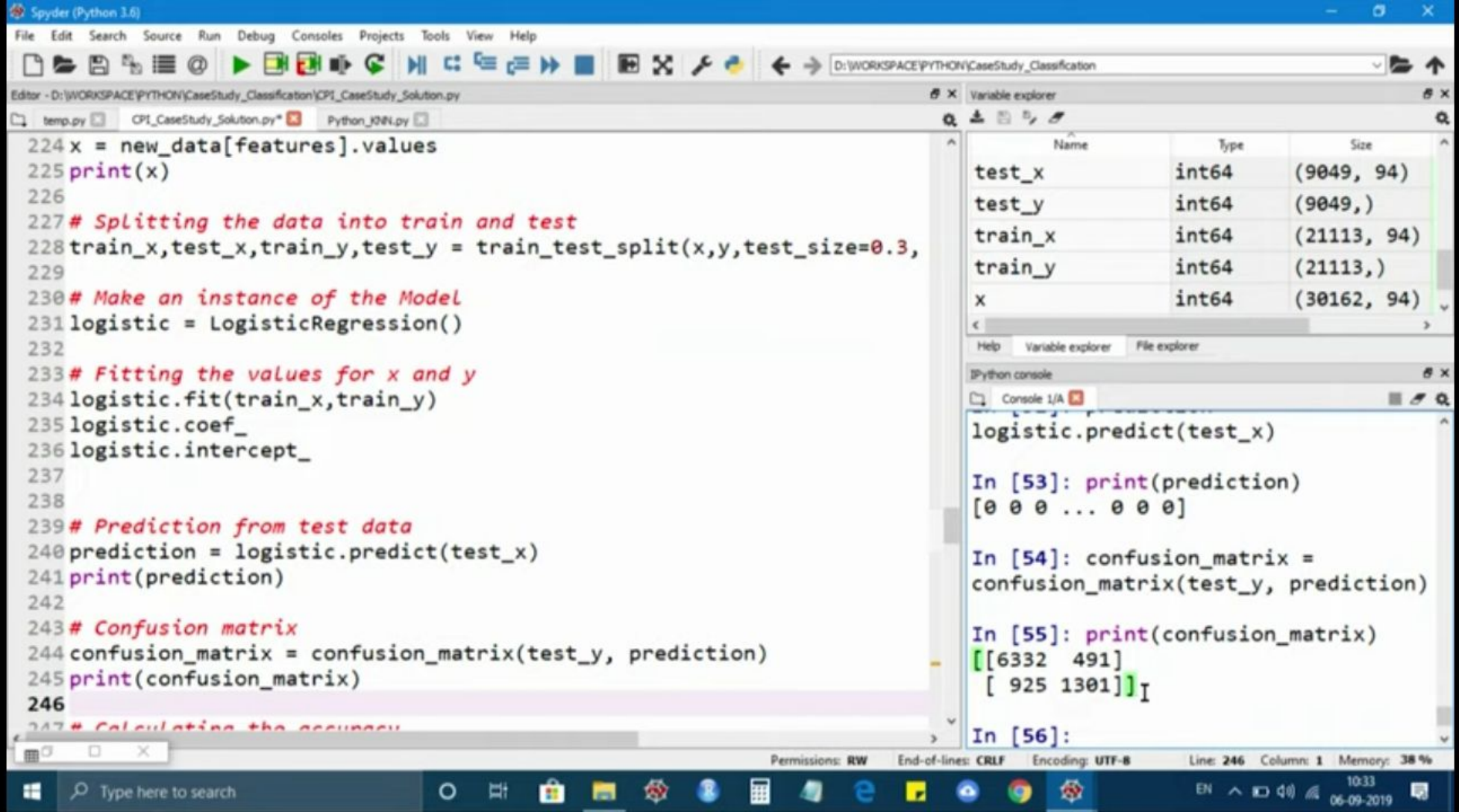

Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\WORKSPACE\PYTHON\CaseStudy_Classification
Editor - D:\WORKSPACE\PYTHON\CaseStudy_Classification\CPI_CaseStudy_Solution.py
temp.py
CPI_CaseStudy_Solution.py*
Python_KNN.py
train_x,test_x,train_y,test_y = train_test_split(x,y,test_size=0.3,
# Make an instance of the Model
logistic = LogisticRegression()
# Fitting the values for x and y
logistic.fit(train_x,train_y)
logistic.coef_
logistic.intercept_
# Prediction from test data
prediction = logistic.predict(test_x)
print(prediction)
# Confusion matrix
confusion_matrix = confusion_matrix(test_y, prediction)
print(confusion_matrix)
# Calculating the accuracy
accuracy_score=accuracy_score(test_y, prediction)
print(accuracy_score)
Variable explorer
Name Type Size
summary_num DataFrame (8, 4)
test_x int64 (9049, 94)
test_y int64 (9049,)
train_x int64 (21113, 94)
train_y int64 (21113,)
Help Variable explorer File explorer
IPython console
Console 1/A
In [55]: print(confusion_matrix)
[[6332 491]
[925 1301]]
In [56]:
accuracy_score=accuracy_score(test_y
, prediction)
In [57]: print(accuracy_score)
0.843518620842082
In [58]:
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 250 Column: 1 Memory: 38 %
Type here to search
10:35
06-09-2019

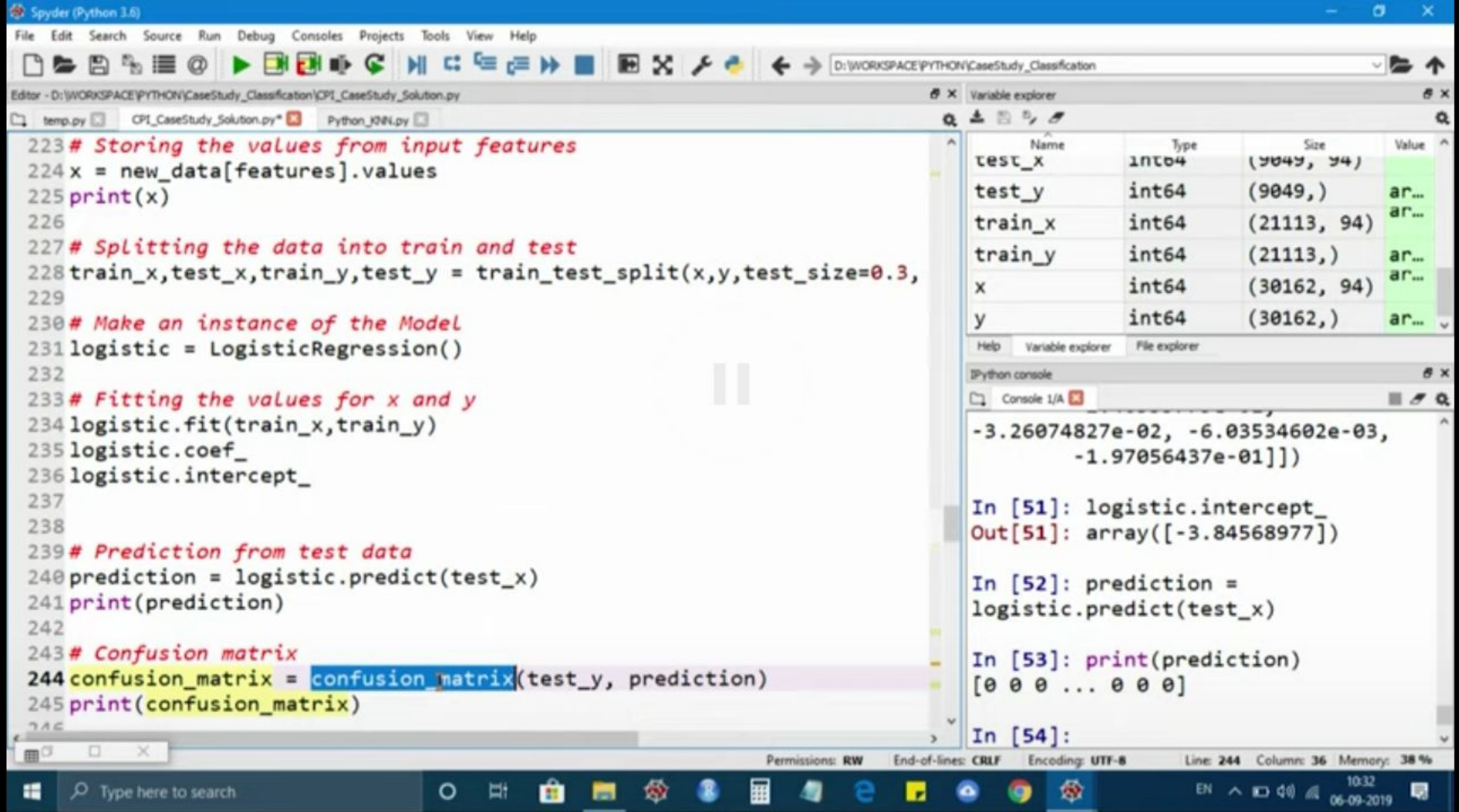

Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\WORKSPACE\PYTHON\CaseStudy_Classification
Editor - D:\WORKSPACE\PYTHON\CaseStudy_Classification\CPI_CaseStudy_Solution.py
temp.py
CPI_CaseStudy_Solution.py
Python_KNN.py
210
211 # Storing the column names
212 columns_list=list(new_data.columns)
213 print(columns_list)
214
215 # Separating the input names from data
216 features=list(set(columns_list)-set(['SalStat']))
217 print(features)
218
219 # Storing the output values in y
220 y=new_data['SalStat'].values
221 print(y)
222
223 # Storing the values from input features
224 x = new_data[features].values
225 print(x)
226
227 # Splitting the data into train and test
228 train_x,test_x,train_y,test_y = train_test_split(x,y,test_size=0.3,
229
230 # Make an instance of the Model
231 logistic = LogisticRegression()
232
Variable explorer
Name Type Size Value
summary_num DataFrame (8, 4) Co...
test_x int64 (9049, 94) ar...
test_y int64 (9049,) ar...
train_x int64 (21113, 94) ar...
train_y int64 (21113,) ar...
x int64 (30162, 94) ar...
Help Variable explorer File explorer
IPython console
Console 1/A
[0 0 0 ... 0 0 0]
[0 0 0 ... 0 0 0]
[0 0 1 ... 0 0 0]]
In [47]:
train_x,test_x,train_y,test_y =
train_test_split(x,y,test_size=0.3,
random_state=0)
In [48]: logistic =
LogisticRegression()
In [49]:
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 232 Column: 1 Memory: 38 %
Type here to search
EN
10:28
06-09-2019

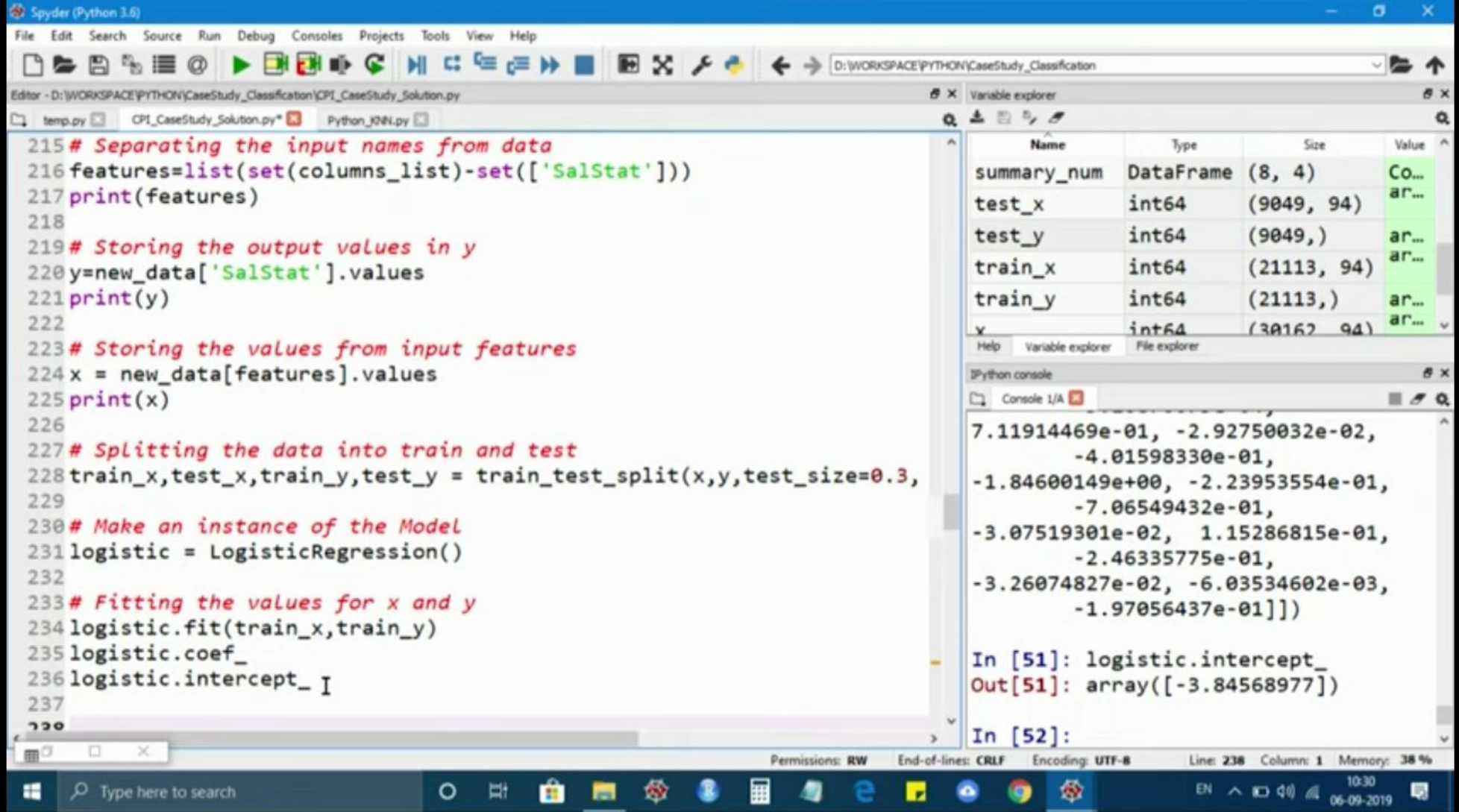

Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\WORKSPACE\PYTHON\CaseStudy_Classification
Editor - D:\WORKSPACE\PYTHON\CaseStudy_Classification\CPI_CaseStudy_Solution.py
temp.py
CPI_CaseStudy_Solution.py
Python_KNN.py
206 data2['SalStat']=data2['SalStat'].map({' less than or equal to 50,000':0,' greater than
207 print(data2['SalStat'])
208
209 new_data=pd.get_dummies(data2, drop_first=True)
210
211 # Storing the column names
212 columns_list=list(new_data.columns)
213 print(columns_list)
214
215 # Separating the input names from data
216 features=list(set(columns_list)-set(['SalStat']))
217 print(features)
218
219 # Storing the output values in y
220 y=new_data['SalStat'].values
221 print(y)
222
223 # Storing the values from input features
224 x = new_data[features].values
225 print(x)
226
227 # Splitting the data into train and test
228 train_x,test_x,train_y,test_y = train_test_split(x,y,test_size=0.3, random_state=0)
Variable explorer
Name
new_data
summary_cate
summary_num
x
y
Help
Variable expl...
File expl...
IPython console
Console 1/A
'capitalgain',
'capitalloss',
'hoursper
week',
'nativecountry',
'SalStat'],
dtype='obj
ect')
In [40]:
data2['SalStat']
=data2['SalStat'
Permissions: RW
End-of-lines: CRLF
Encoding: UTF-8
Line: 227
Column: 41
Memory: 38 %
Type here to search
EN
10:26
06-09-2019

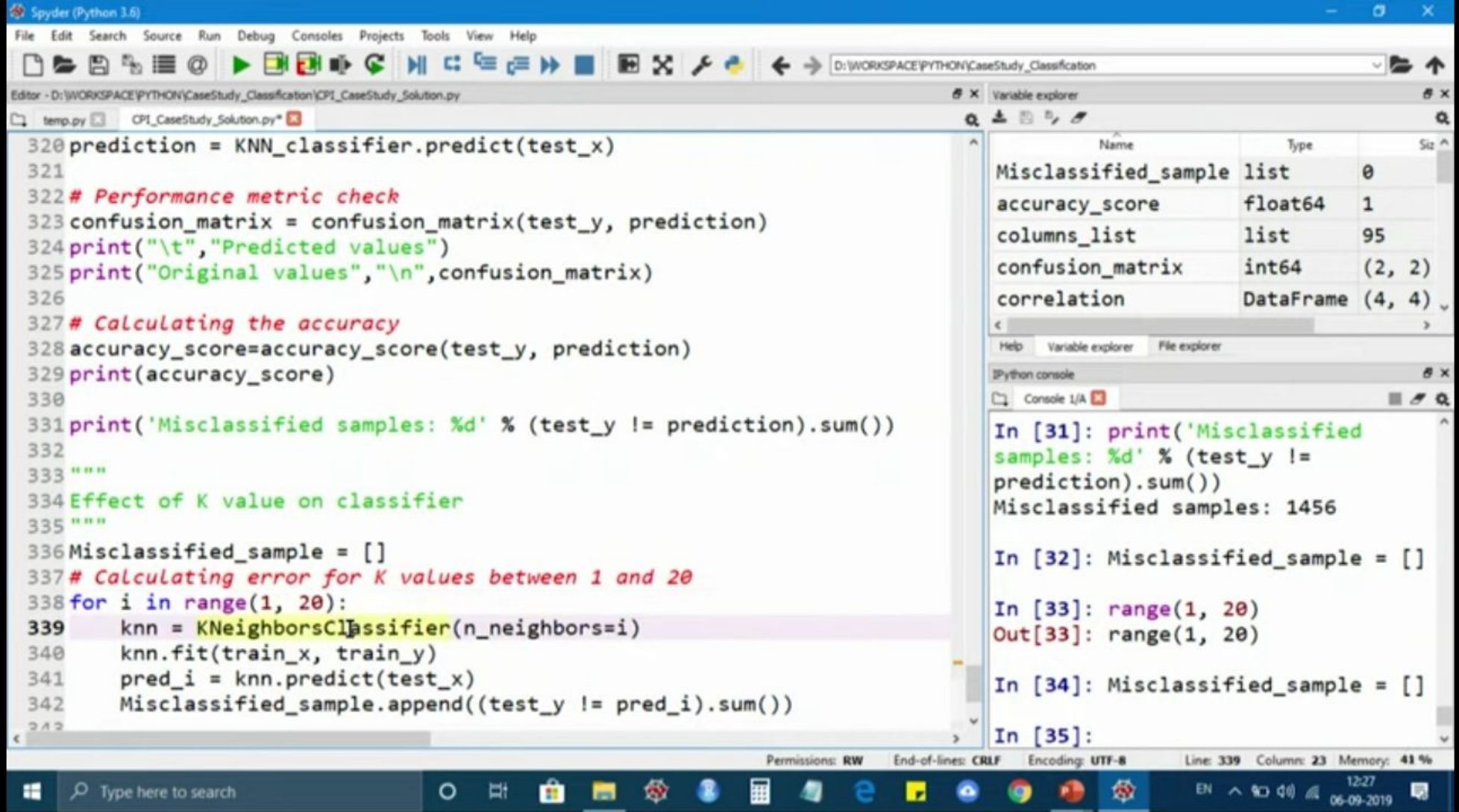

Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\WORKSPACE\PYTHON\CaseStudy_Classification
Editor - D:\WORKSPACE\PYTHON\CaseStudy_Classification\CPI_CaseStudy_Solution.py
temp.py
CPI_CaseStudy_Solution.py
Python_KNN.py
207 print(data2['SalStat'])
208
209 new_data=pd.get_dummies(data2, drop_first=True)
210
211 # Storing the column names
212 columns_list=list(new_data.columns)
213 print(columns_list)
214
215 # Separating the input names from data
216 features=list(set(columns_list)-set(['SalStat']))
217 print(features)
218
219 # Storing the output values in y
220 y=new_data['SalStat'].values
221 print(y)
222
223 # Storing the values from input features
224 x = new_data[features].values
225 print(x)
226
227 # Splitting the data into train and test
228 train_x,test_x,train_y,test_y = train_test_split(x,y,test_size=0.3, ranc
229
Variable explorer
Name Type Size
data_income DataFrame (31978, 13)
features list 94
missing DataFrame (1816, 13)
new_data DataFrame (30162, 95)
summary_cate DataFrame (4, 9)
Help Variable explorer File explorer
IPython console
Console 1/A
'nativecountry_ Hungary',
'nativecountry_ Jamaica', 'race_
White', 'nativecountry_
Honduras', 'nativecountry_
Holand-Netherlands',
'nativecountry_ Guatemala']
In [45]:
y=new_data['SalStat'].values
...: print(y)
[0 0 1 ... 0 0 0]
In [46]:
Run selection or current line
Permissions: RW
End-of-lines: CRLF
Encoding: UTF-8
Line: 224
Column: 1
Memory: 38 %
Type here to search
EN
10:24
06-09-2019

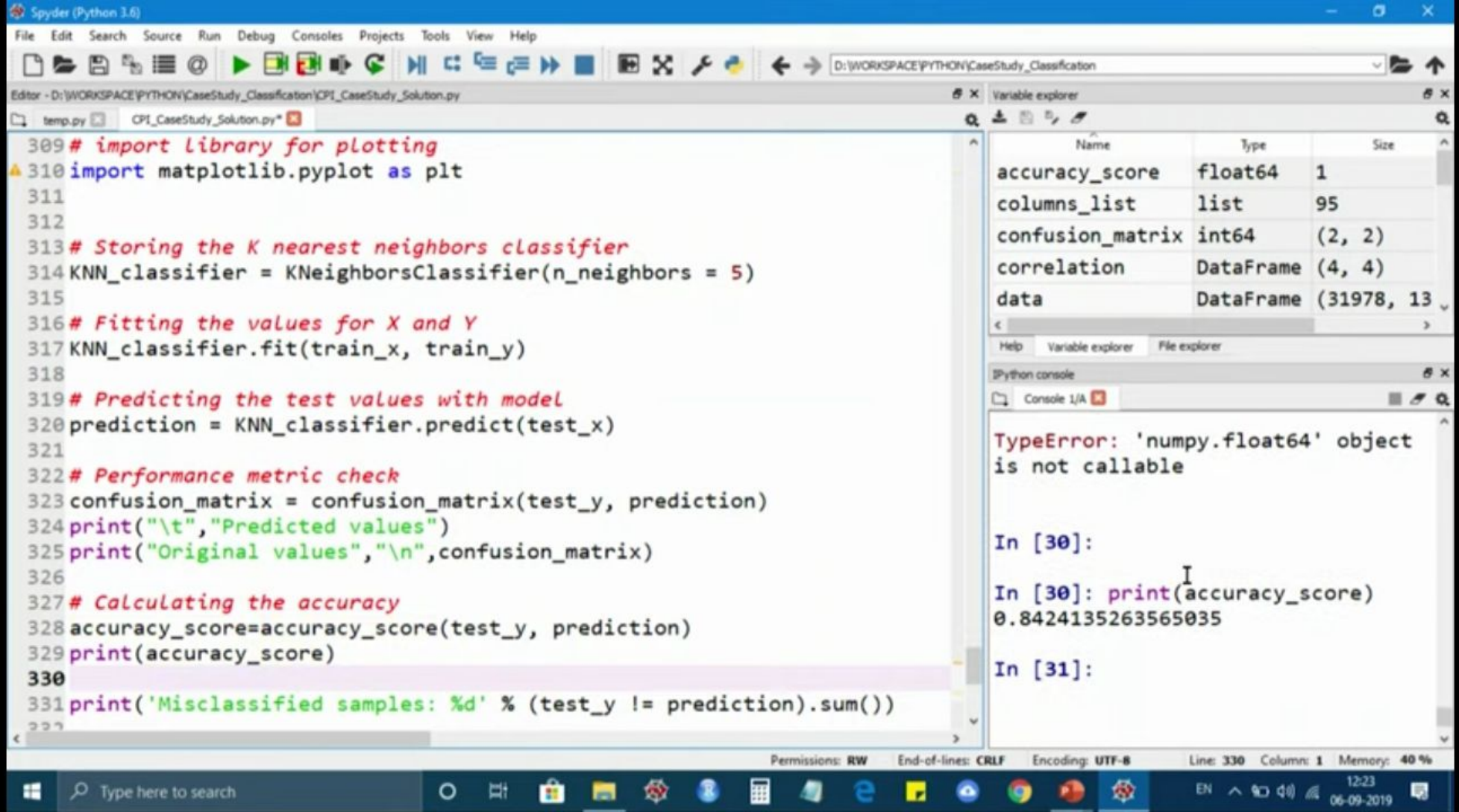

Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\WORKSPACE\PYTHON\CaseStudy_Classification
Editor - D:\WORKSPACE\PYTHON\CaseStudy_Classification\CPI_CaseStudy_S
temp.py
CPI_CaseStudy_Solution.py
Python_KNN.py
200
201 # ==========
202 # LOGISTIC REGRESSION
203 # ==========
204
205 # Reindexing the salary sta
206 data2['SalStat']=data2['Sal
207 print(data2['SalStat'])
208
209 new_data=pd.get_dummies(dat
210
211 # Storing the column names
212 columns_list=list(new_data.
213 print(columns_list)
214
215 # Separating the input name
216 features=list(set(columns_l
217 print(features)
218
219 # Storing the output values
220 y=new_data['SalStat'].values
221 print(y)
222
new_data - DataFrame
Index
alStat
JobType_ Local-gov
JobType_ Private
Type_ Self-emp-i
0 0 1 0 0
1 0 0 0 0
2 0 1 0 0
3 0 1 0 0
4 0 1 0 0
5 0 1 0 0
6 0 1 0 0
7 0 1 0 0
9 0 1 0 0
10 0 0 1 0
11 0 1 0 0
12 0 0 0 1
13 0 1 0 0
Format
Resize
Background color
Column min/max
OK
Cancel
Variable explorer
Name
Type
data_income DataFrame
missing DataFrame
new_data DataFrame
summary_cate DataFrame
summary_num DataFrame
Help
Variable explorer
File explorer
IPython console
Console 1/A
See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy
In [41]:
new_data=pd.get_dummies(data2, drop_first=True)
In [42]:
Permissions: RW
End-of-lines: CRLF
Encoding: UTF-8
Line: 210
Column: 1
Memory: 37 %
Type here to search
EN
10:21
06-09-2019

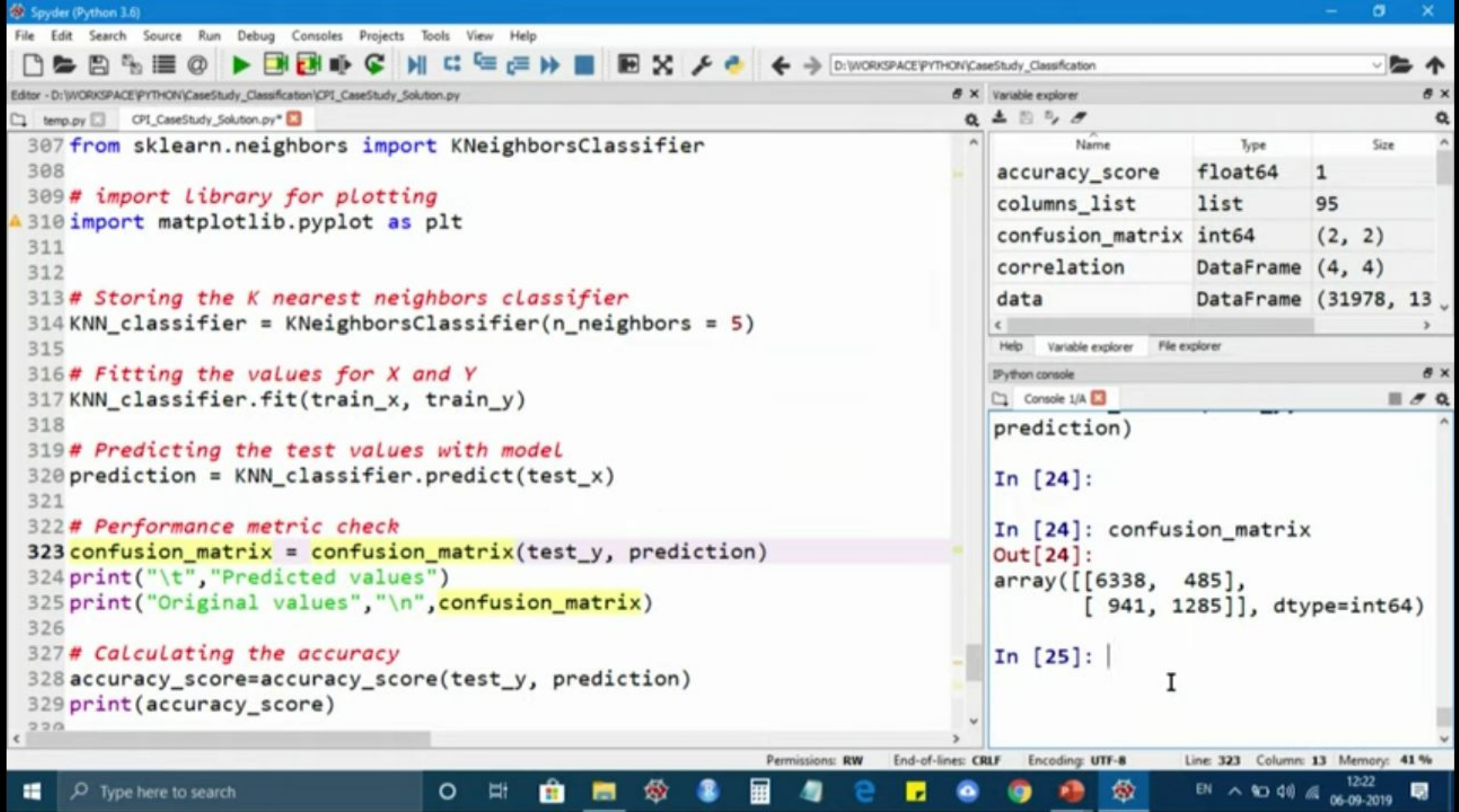

Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\WORKSPACE\PYTHON\CaseStudy_Classification
Editor - D:\WORKSPACE\PYTHON\CaseStudy_Classification\CPI_CaseStudy_Solution.py
temp.py
CPI_CaseStudy_Solution.py*
Python_KNN.py
224 x = new_data[features].values
225 print(x)
226
227 # Splitting the data into train and test
228 train_x,test_x,train_y,test_y = train_test_split(x,y,test_size=0.3,
229
230 # Make an instance of the Model
231 logistic = LogisticRegression()
232
233 # Fitting the values for x and y
234 logistic.fit(train_x,train_y)
235 logistic.coef_
236 logistic.intercept_
237
238
239 # Prediction from test data
240 prediction = logistic.predict(test_x)
241 print(prediction)
242
243 # Confusion matrix
244 confusion_matrix = confusion_matrix(test_y, prediction)
245 print(confusion_matrix)
246
Variable explorer
Name Type Size
test_x int64 (9049, 94)
test_y int64 (9049,)
train_x int64 (21113, 94)
train_y int64 (21113,)
x int64 (30162, 94)
Help Variable explorer File explorer
IPython console
Console 1/A
logistic.predict(test_x)
In [53]: print(prediction)
[0 0 0 ... 0 0 0]
In [54]: confusion_matrix =
confusion_matrix(test_y, prediction)
In [55]: print(confusion_matrix)
[[6332 491]
[925 1301]]
In [56]:
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 246 Column: 1 Memory: 38 %
Type here to search
EN
10:33
06-09-2019

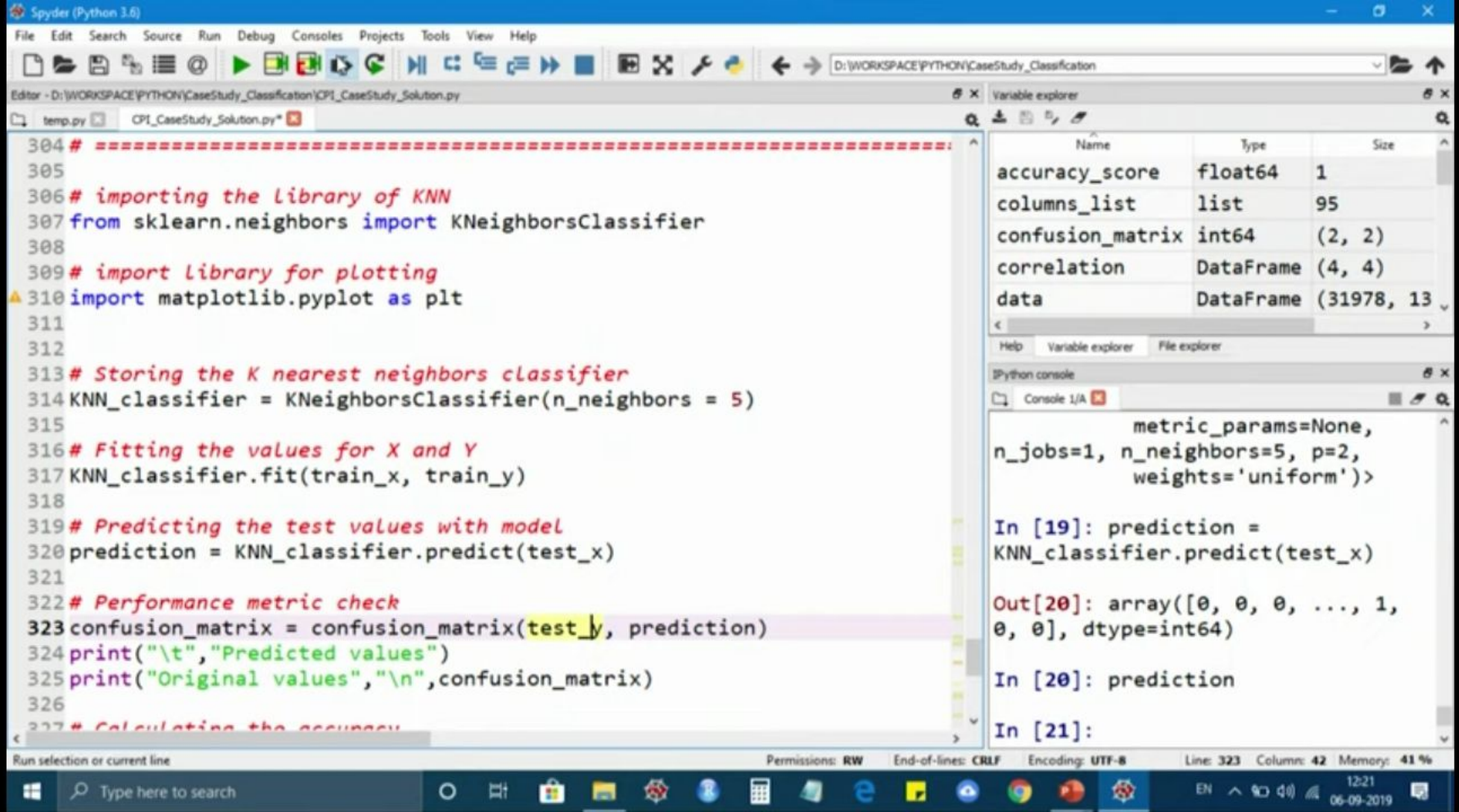

Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\WORKSPACE\PYTHON\CaseStudy_Classification
Editor - D:\WORKSPACE\PYTHON\CaseStudy_Classification\CPI_CaseStudy_Solution.py
temp.py
CPI_CaseStudy_Solution.py*
Python_JDBi.py
223 # Storing the values from input features
224 x = new_data[features].values
225 print(x)
226
227 # Splitting the data into train and test
228 train_x,test_x,train_y,test_y = train_test_split(x,y,test_size=0.3,
229
230 # Make an instance of the Model
231 logistic = LogisticRegression()
232
233 # Fitting the values for x and y
234 logistic.fit(train_x,train_y)
235 logistic.coef_
236 logistic.intercept_
237
238
239 # Prediction from test data
240 prediction = logistic.predict(test_x)
241 print(prediction)
242
243 # Confusion matrix
244 confusion_matrix = confusion_matrix(test_y, prediction)
245 print(confusion_matrix)
Variable explorer
Name Type Size Value
test_y int64 (9049,) ar...
train_x int64 (21113, 94) ar...
train_y int64 (21113,) ar...
x int64 (30162, 94) ar...
y int64 (30162,) ar...
Help Variable explorer File explorer
IPython console
Console 1/A
-3.26074827e-02, -6.03534602e-03,
-1.97056437e-01]])
In [51]: logistic.intercept_
Out[51]: array([-3.84568977])
In [52]: prediction =
logistic.predict(test_x)
In [53]: print(prediction)
[0 0 0 ... 0 0 0]
In [54]:
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 244 Column: 36 Memory: 38 %
Type here to search
EN
10:32
06-09-2019

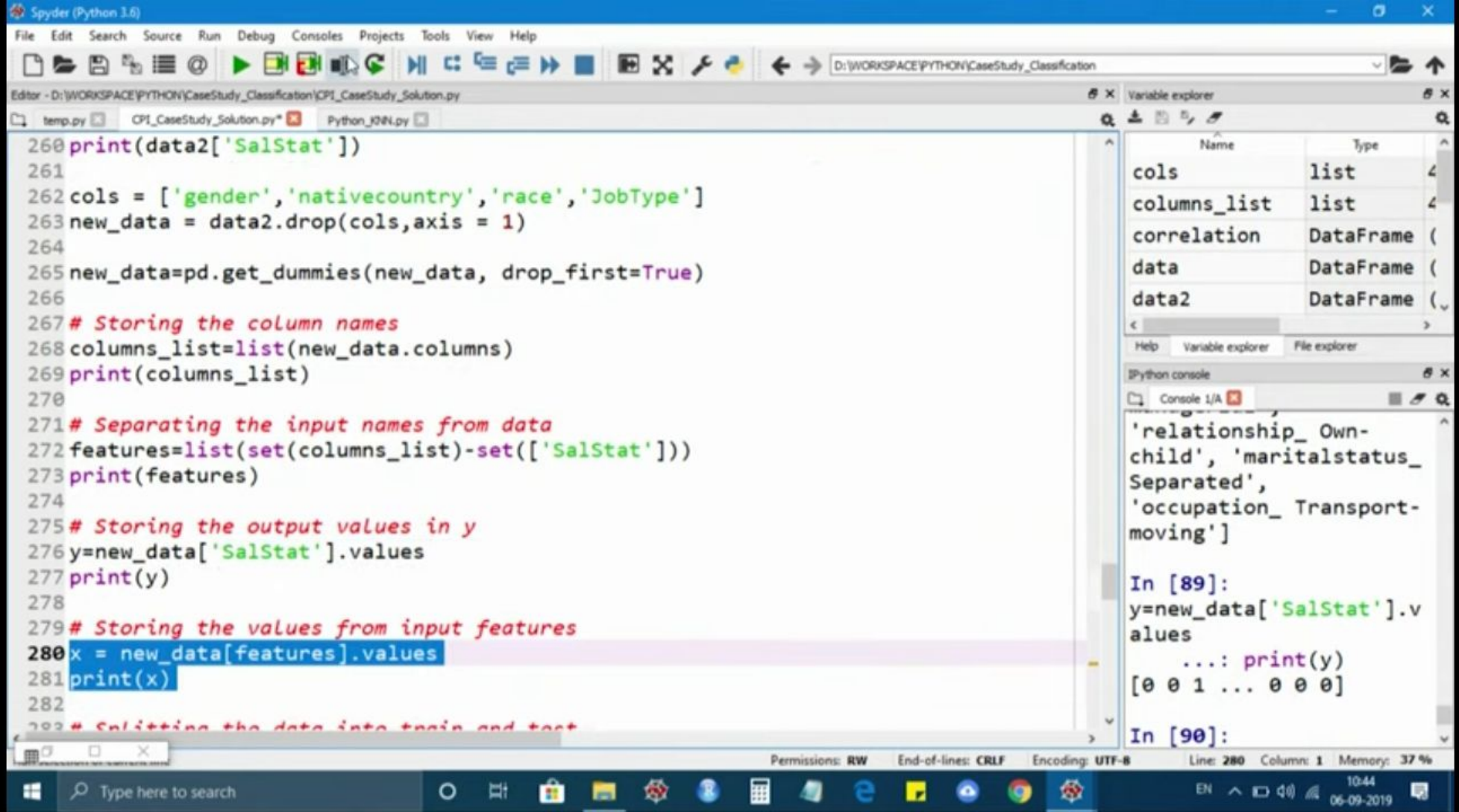

Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\WORKSPACE\PYTHON\CaseStudy_Classification
Editor - D:\WORKSPACE\PYTHON\CaseStudy_Classification\CPI_CaseStudy_Solution.py
temp.py
CPI_CaseStudy_Solution.py*
Python_KNN.py
215 # Separating the input names from data
216 features=list(set(columns_list)-set(['SalStat']))
217 print(features)
218
219 # Storing the output values in y
220 y=new_data['SalStat'].values
221 print(y)
222
223 # Storing the values from input features
224 x = new_data[features].values
225 print(x)
226
227 # Splitting the data into train and test
228 train_x,test_x,train_y,test_y = train_test_split(x,y,test_size=0.3,
229
230 # Make an instance of the Model
231 logistic = LogisticRegression()
232
233 # Fitting the values for x and y
234 logistic.fit(train_x,train_y)
235 logistic.coef_
236 logistic.intercept_
237
Variable explorer
Name Type Size Value
summary_num DataFrame (8, 4) Co...
test_x int64 (9049, 94) ar...
test_y int64 (9049,) ar...
train_x int64 (21113, 94) ar...
train_y int64 (21113,) ar...
x int64 (30162, 94) ar...
Help Variable explorer File explorer
IPython console
Console 1/A
7.11914469e-01, -2.92750032e-02,
-4.01598330e-01,
-1.84600149e+00, -2.23953554e-01,
-7.06549432e-01,
-3.07519301e-02, 1.15286815e-01,
-2.46335775e-01,
-3.26074827e-02, -6.03534602e-03,
-1.97056437e-01]])
In [51]: logistic.intercept_
Out[51]: array([-3.84568977])
In [52]:
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 238 Column: 1 Memory: 38 %
Type here to search
EN 10:30 06-09-2019

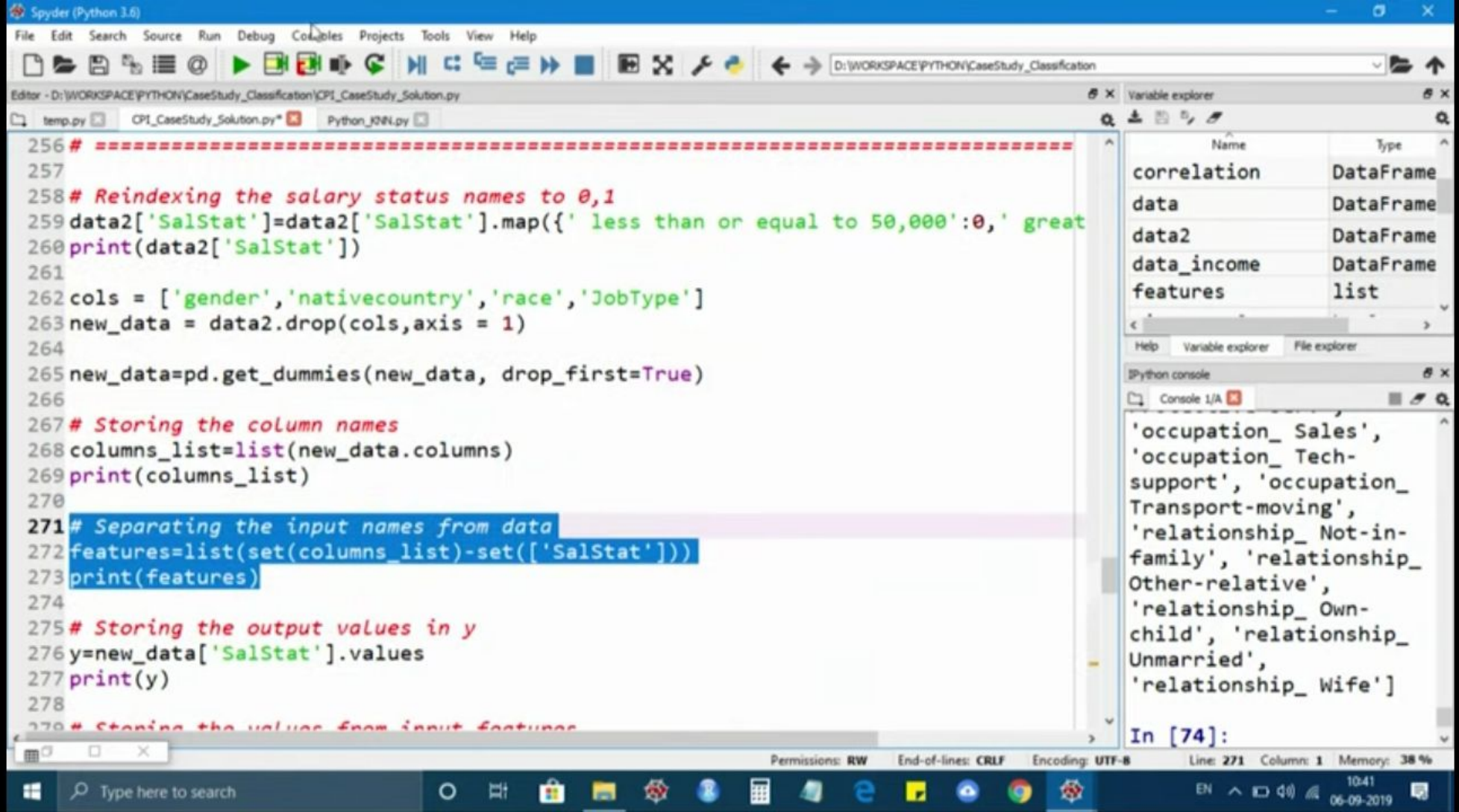

Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\WORKSPACE\PYTHON\CaseStudy_Classification
Editor - D:\WORKSPACE\PYTHON\CaseStudy_Classification\CPI_CaseStudy_Solution.py
temp.py
CPI_CaseStudy_Solution.py*
320 prediction = KNN_classifier.predict(test_x)
321
322 # Performance metric check
323 confusion_matrix = confusion_matrix(test_y, prediction)
324 print("\t","Predicted values")
325 print("Original values","\n",confusion_matrix)
326
327 # Calculating the accuracy
328 accuracy_score=accuracy_score(test_y, prediction)
329 print(accuracy_score)
330
331 print('Misclassified samples: %d' % (test_y != prediction).sum())
332
333 """
334 Effect of K value on classifier
335 """
336 Misclassified_sample = []
337 # Calculating error for K values between 1 and 20
338 for i in range(1, 20):
339 knn = KNeighborsClassifier(n_neighbors=i)
340 knn.fit(train_x, train_y)
341 pred_i = knn.predict(test_x)
342 Misclassified_sample.append((test_y != pred_i).sum())
Variable explorer
Name Type Siz
Misclassified_sample list 0
accuracy_score float64 1
columns_list list 95
confusion_matrix int64 (2, 2)
correlation DataFrame (4, 4)
Help Variable explorer File explorer
IPython console
Console 1/A
In [31]: print('Misclassified samples: %d' % (test_y != prediction).sum())
Misclassified samples: 1456
In [32]: Misclassified_sample = []
In [33]: range(1, 20)
Out[33]: range(1, 20)
In [34]: Misclassified_sample = []
In [35]:
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 339 Column: 23 Memory: 41 %
Type here to search
EN 12:27 06-09-2019

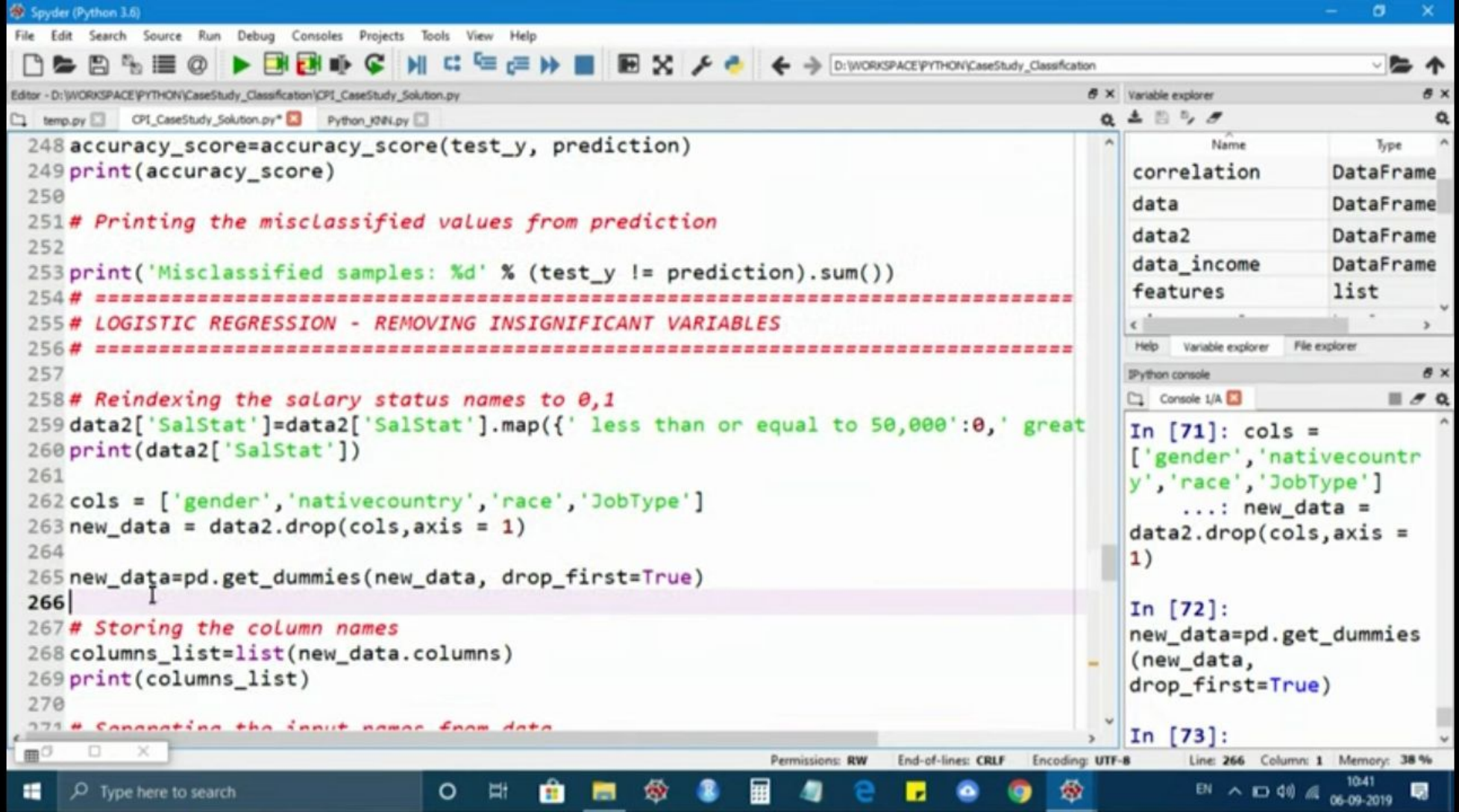

Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\WORKSPACE\PYTHON\CaseStudy_Classification
Editor - D:\WORKSPACE\PYTHON\CaseStudy_Classification\CPI_CaseStudy_Solution.py
temp.py
CPI_CaseStudy_Solution.py*
309 # import library for plotting
310 import matplotlib.pyplot as plt
311
312
313 # Storing the K nearest neighbors classifier
314 KNN_classifier = KNeighborsClassifier(n_neighbors = 5)
315
316 # Fitting the values for X and Y
317 KNN_classifier.fit(train_x, train_y)
318
319 # Predicting the test values with model
320 prediction = KNN_classifier.predict(test_x)
321
322 # Performance metric check
323 confusion_matrix = confusion_matrix(test_y, prediction)
324 print("\t","Predicted values")
325 print("Original values","\n",confusion_matrix)
326
327 # Calculating the accuracy
328 accuracy_score=accuracy_score(test_y, prediction)
329 print(accuracy_score)
330
331 print('Misclassified samples: %d' % (test_y != prediction).sum())
Variable explorer
Name Type Size
accuracy_score float64 1
columns_list list 95
confusion_matrix int64 (2, 2)
correlation DataFrame (4, 4)
data DataFrame (31978, 13
Help Variable explorer File explorer
IPython console
Console 1/A
TypeError: 'numpy.float64' object is not callable
In [30]:
In [30]: print(accuracy_score)
0.8424135263565035
In [31]:
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 330 Column: 1 Memory: 40 %
Type here to search
12:23
06-09-2019

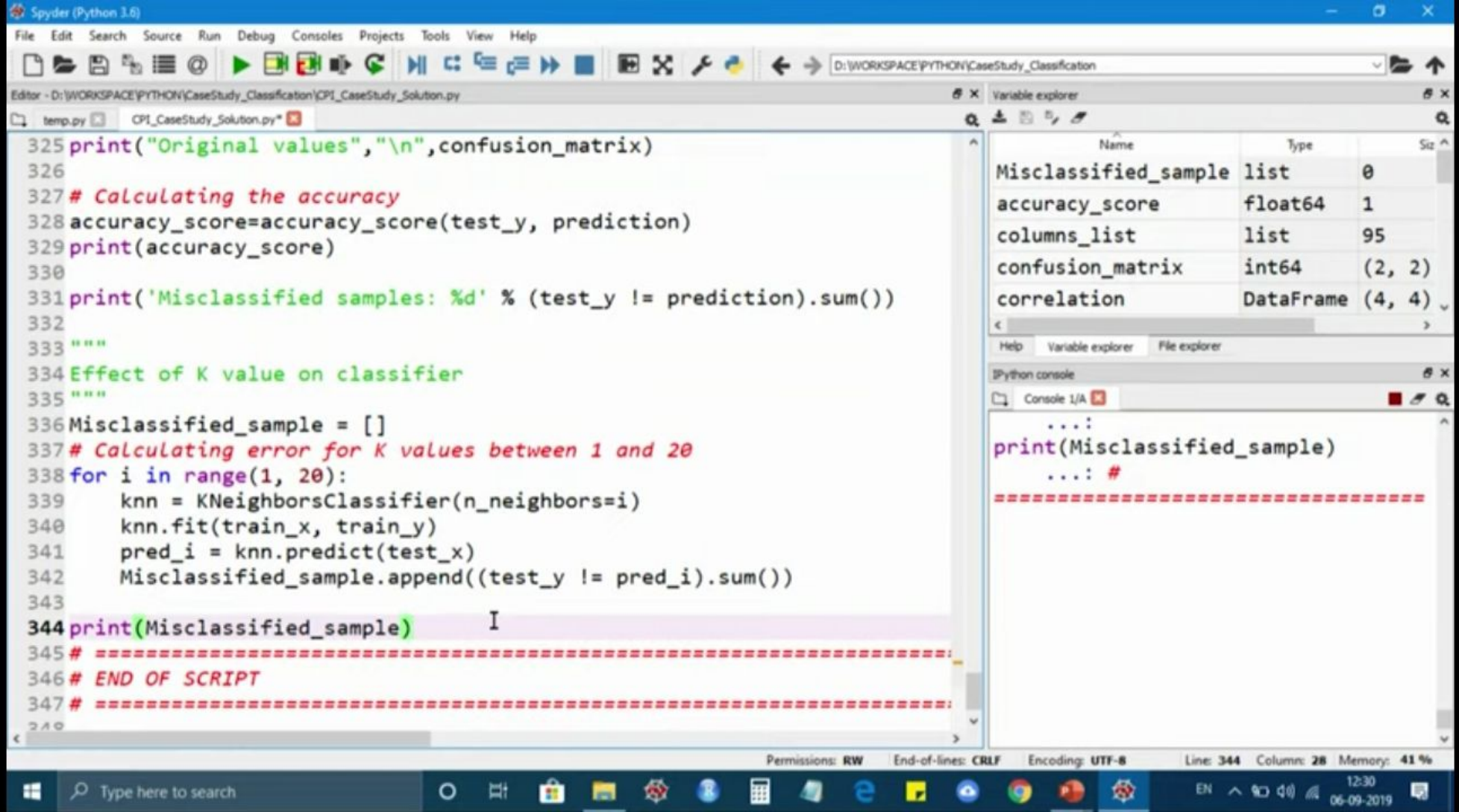

Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\WORKSPACE\PYTHON\CaseStudy_Classification
Editor - D:\WORKSPACE\PYTHON\CaseStudy_Classification\CPI_CaseStudy_Solution.py
temp.py
CPI_CaseStudy_Solution.py*
307 from sklearn.neighbors import KNeighborsClassifier
308
309 # import library for plotting
310 import matplotlib.pyplot as plt
311
312
313 # Storing the K nearest neighbors classifier
314 KNN_classifier = KNeighborsClassifier(n_neighbors = 5)
315
316 # Fitting the values for X and Y
317 KNN_classifier.fit(train_x, train_y)
318
319 # Predicting the test values with model
320 prediction = KNN_classifier.predict(test_x)
321
322 # Performance metric check
323 confusion_matrix = confusion_matrix(test_y, prediction)
324 print("\t","Predicted values")
325 print("Original values","\n",confusion_matrix)
326
327 # Calculating the accuracy
328 accuracy_score=accuracy_score(test_y, prediction)
329 print(accuracy_score)
Variable explorer
Name Type Size
accuracy_score float64 1
columns_list list 95
confusion_matrix int64 (2, 2)
correlation DataFrame (4, 4)
data DataFrame (31978, 13
Help Variable explorer File explorer
IPython console
Console 1/A
prediction)
In [24]:
In [24]: confusion_matrix
Out[24]:
array([[6338, 485],
[941, 1285]], dtype=int64)
In [25]:
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 323 Column: 13 Memory: 41 %
Type here to search
12:22
06-09-2019

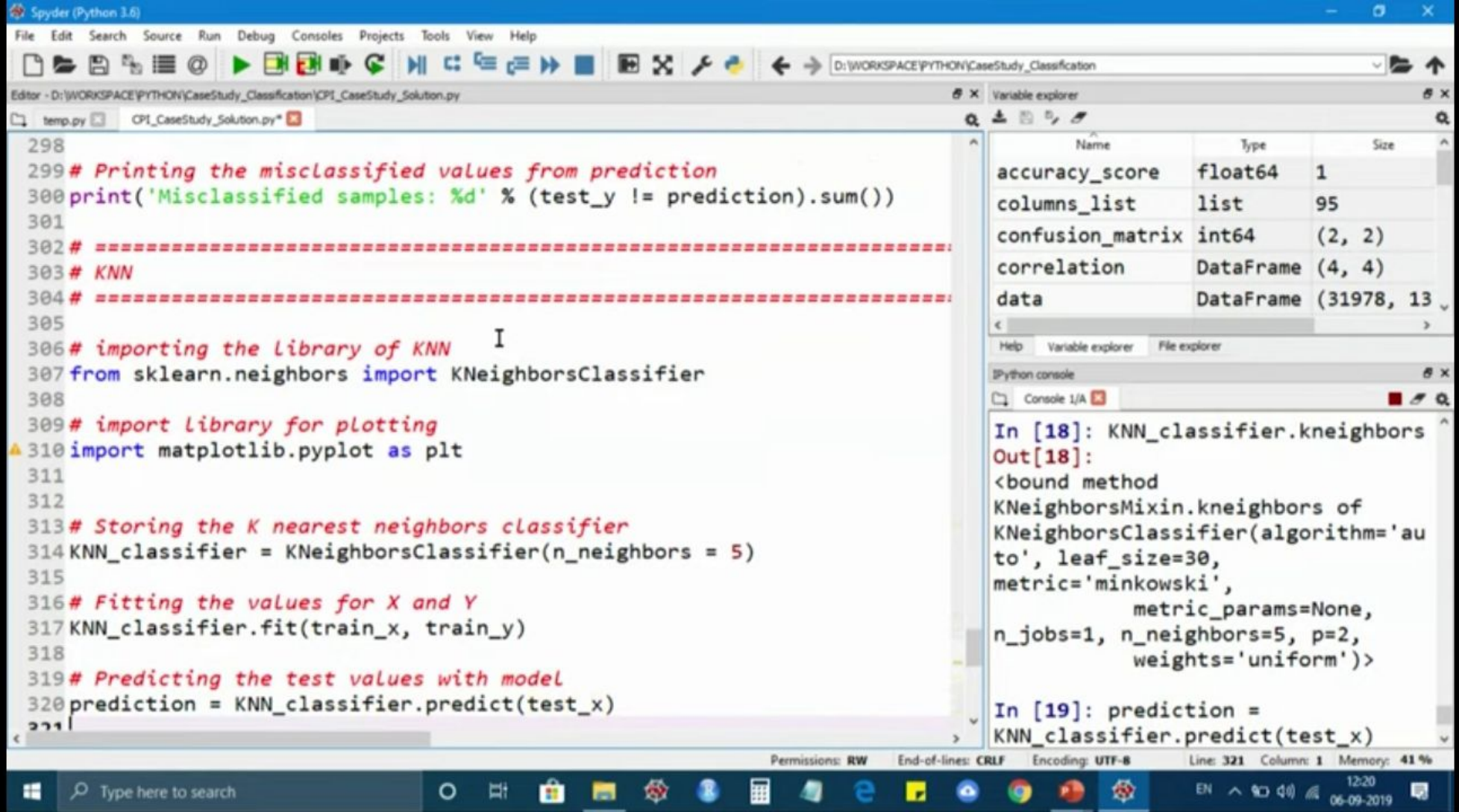

Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\WORKSPACE\PYTHON\CaseStudy_Classification
Editor - D:\WORKSPACE\PYTHON\CaseStudy_Classification\CPI_CaseStudy_Solution.py
temp.py
CPI_CaseStudy_Solution.py*
304 # ==========
305
306 # importing the library of KNN
307 from sklearn.neighbors import KNeighborsClassifier
308
309 # import library for plotting
310 import matplotlib.pyplot as plt
311
312
313 # Storing the K nearest neighbors classifier
314 KNN_classifier = KNeighborsClassifier(n_neighbors = 5)
315
316 # Fitting the values for X and Y
317 KNN_classifier.fit(train_x, train_y)
318
319 # Predicting the test values with model
320 prediction = KNN_classifier.predict(test_x)
321
322 # Performance metric check
323 confusion_matrix = confusion_matrix(test_y, prediction)
324 print("\t","Predicted values")
325 print("Original values","\n",confusion_matrix)
326
Variable explorer
Name Type Size
accuracy_score float64 1
columns_list list 95
confusion_matrix int64 (2, 2)
correlation DataFrame (4, 4)
data DataFrame (31978, 13
Help Variable explorer File explorer
IPython console
Console 1/A
metric_params=None,
n_jobs=1, n_neighbors=5, p=2,
weights='uniform')>
In [19]: prediction =
KNN_classifier.predict(test_x)
Out[20]: array([0, 0, 0, ..., 1,
0, 0], dtype=int64)
In [20]: prediction
In [21]:
Run selection or current line
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 323 Column: 42 Memory: 41 %
Type here to search
EN
12:21
06-09-2019

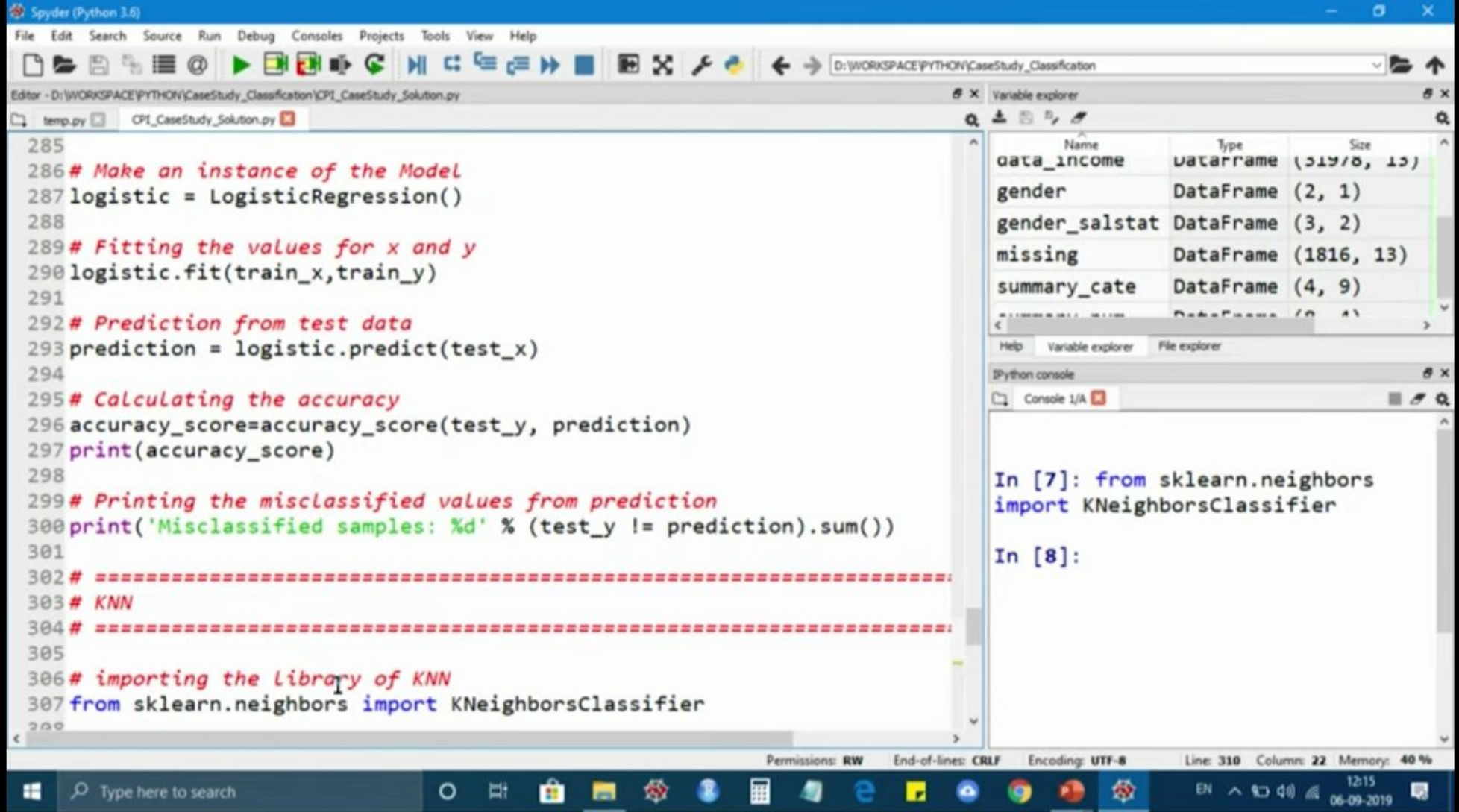

Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\WORKSPACE\PYTHON\CaseStudy_Classification
Editor - D:\WORKSPACE\PYTHON\CaseStudy_Classification\CPI_CaseStudy_Solution.py
temp.py
CPI_CaseStudy_Solution.py*
Python_KNN.py
print(data2['SalStat'])
cols = ['gender','nativecountry','race','JobType']
new_data = data2.drop(cols,axis = 1)
new_data=pd.get_dummies(new_data, drop_first=True)
# Storing the column names
columns_list=list(new_data.columns)
print(columns_list)
# Separating the input names from data
features=list(set(columns_list)-set(['SalStat']))
print(features)
# Storing the output values in y
y=new_data['SalStat'].values
print(y)
# Storing the values from input features
x = new_data[features].values
print(x)
Variable explorer
Name Type
cols list
columns_list list
correlation DataFrame
data DataFrame
data2 DataFrame
Help Variable explorer File explorer
IPython console
Console 1/A
'relationship_ Own-child', 'maritalstatus_ Separated', 'occupation_ Transport-moving']
In [89]:
y=new_data['SalStat'].values
...: print(y)
[0 0 1 ... 0 0 0]
In [90]:
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 280 Column: 1 Memory: 37 %
Type here to search
10:44
06-09-2019

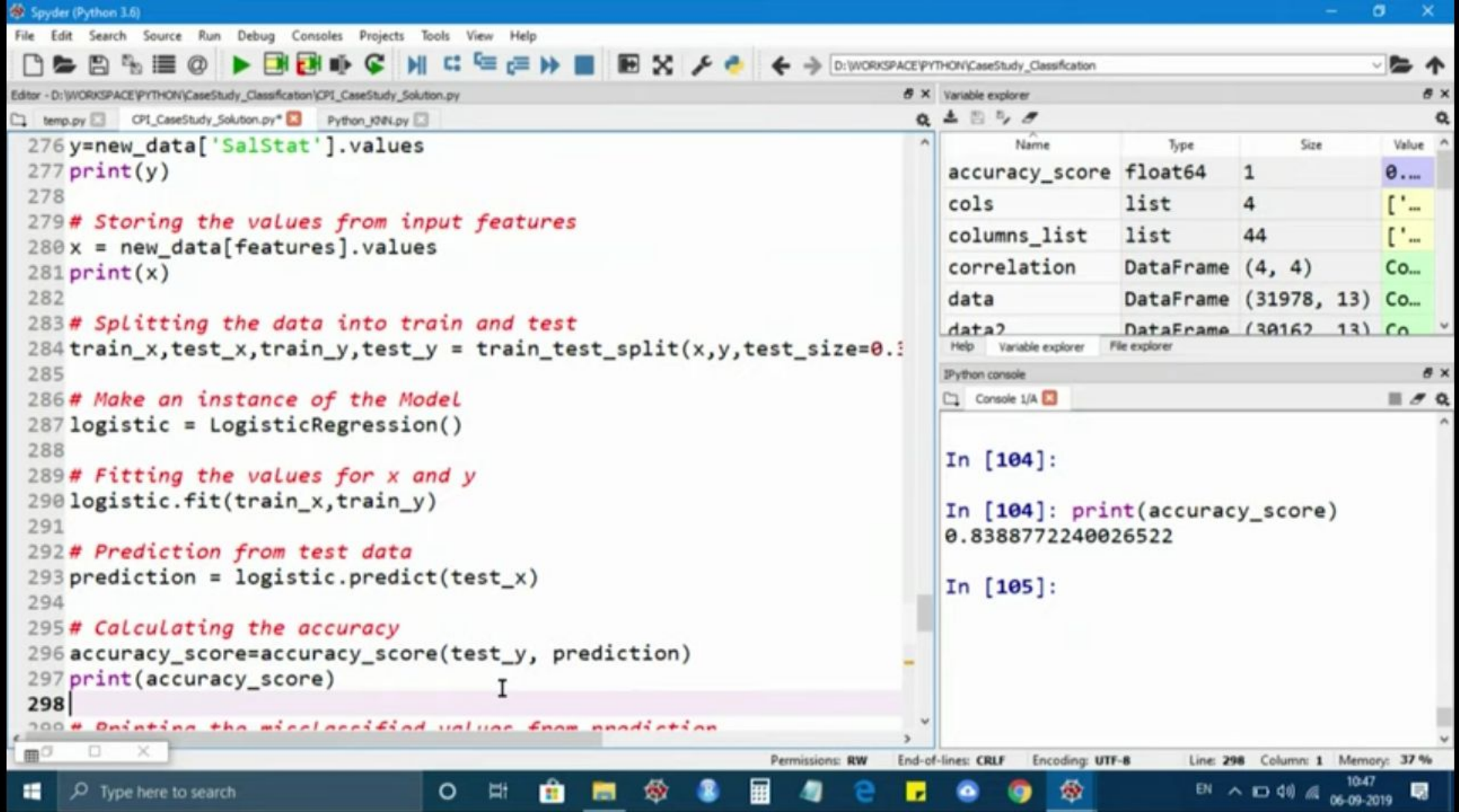

Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\WORKSPACE\PYTHON\CaseStudy_Classification
Editor - D:\WORKSPACE\PYTHON\CaseStudy_Classification\CPI_CaseStudy_Solution.py
temp.py
CPI_CaseStudy_Solution.py*
Python_KNN.py
256 # ==========
257
258 # Reindexing the salary status names to 0,1
259 data2['SalStat']=data2['SalStat'].map({' less than or equal to 50,000':0,' great
260 print(data2['SalStat'])
261
262 cols = ['gender','nativecountry','race','JobType']
263 new_data = data2.drop(cols,axis = 1)
264
265 new_data=pd.get_dummies(new_data, drop_first=True)
266
267 # Storing the column names
268 columns_list=list(new_data.columns)
269 print(columns_list)
270
271 # Separating the input names from data
272 features=list(set(columns_list)-set(['SalStat']))
273 print(features)
274
275 # Storing the output values in y
276 y=new_data['SalStat'].values
277 print(y)
278
Variable explorer
Name Type
correlation DataFrame
data DataFrame
data2 DataFrame
data_income DataFrame
features list
Help Variable explorer File explorer
IPython console
Console 1/A
'occupation_ Sales',
'occupation_ Tech-
support', 'occupation_
Transport-moving',
'relationship_ Not-in-
family', 'relationship_
Other-relative',
'relationship_ Own-
child', 'relationship_
Unmarried',
'relationship_ Wife']
In [74]:
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 271 Column: 1 Memory: 38 %
Type here to search
EN 10:41 06-09-2019

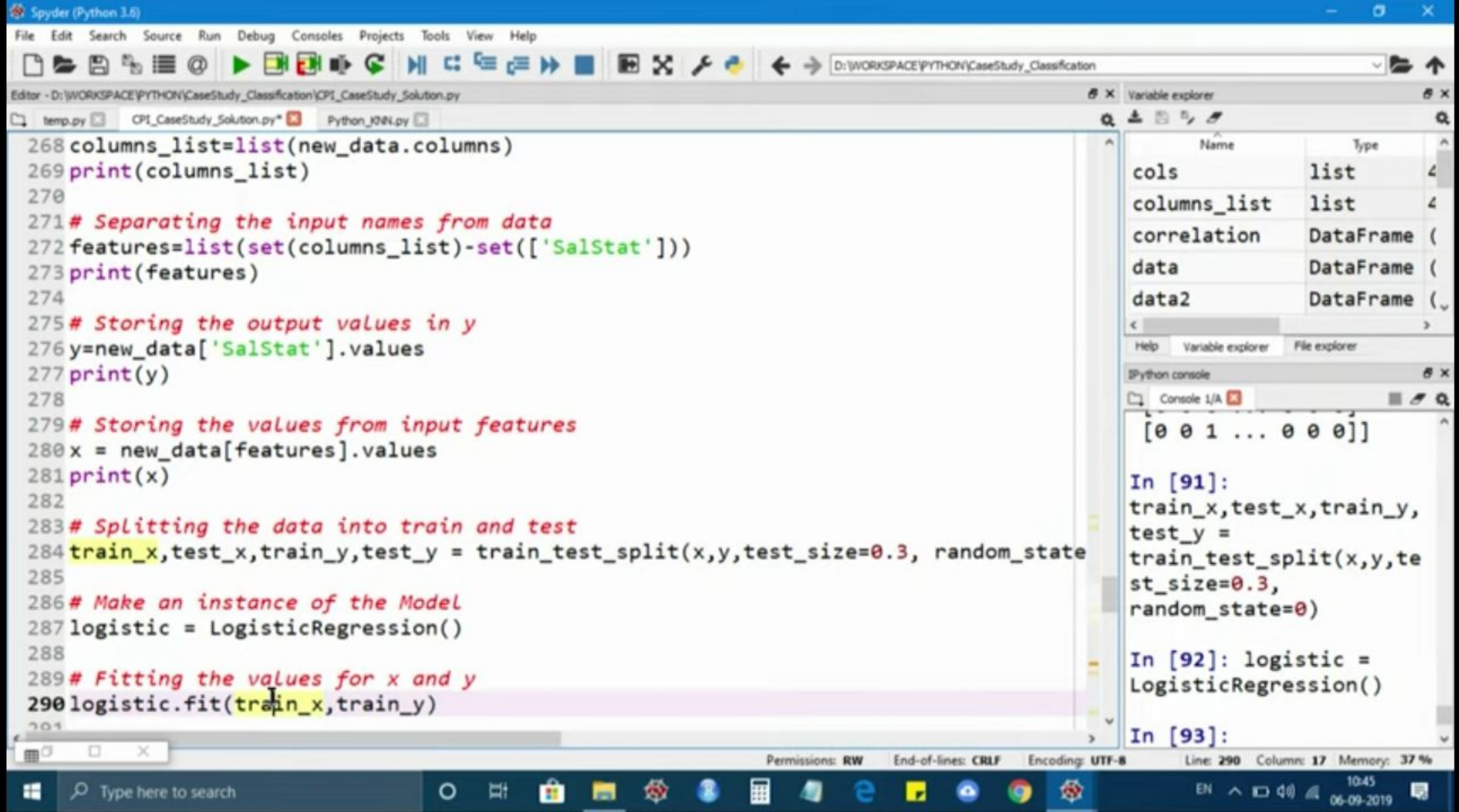

Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\WORKSPACE\PYTHON\CaseStudy_Classification
Editor - D:\WORKSPACE\PYTHON\CaseStudy_Classification\CPI_CaseStudy_Solution.py
temp.py
CPI_CaseStudy_Solution.py*
Python_KNN.py
248 accuracy_score=accuracy_score(test_y, prediction)
249 print(accuracy_score)
250
251 # Printing the misclassified values from prediction
252
253 print('Misclassified samples: %d' % (test_y != prediction).sum())
254 # ==
255 # LOGISTIC REGRESSION - REMOVING INSIGNIFICANT VARIABLES
256 # ==
257
258 # Reindexing the salary status names to 0,1
259 data2['SalStat']=data2['SalStat'].map({' less than or equal to 50,000':0,' great
260 print(data2['SalStat'])
261
262 cols = ['gender','nativecountry','race','JobType']
263 new_data = data2.drop(cols,axis = 1)
264
265 new_data=pd.get_dummies(new_data, drop_first=True)
266
267 # Storing the column names
268 columns_list=list(new_data.columns)
269 print(columns_list)
270
Variable explorer
Name Type
correlation DataFrame
data DataFrame
data2 DataFrame
data_income DataFrame
features list
Help Variable explorer File explorer
IPython console
Console 1/A
In [71]: cols =
['gender','nativecountr
y','race','JobType']
...: new_data =
data2.drop(cols,axis =
1)
In [72]:
new_data=pd.get_dummies
(new_data,
drop_first=True)
In [73]:
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 266 Column: 1 Memory: 38 %
Type here to search
EN
10:41
06-09-2019


Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\WORKSPACE\PYTHON\CaseStudy_Classification
Editor - D:\WORKSPACE\PYTHON\CaseStudy_Classification\CPI_CaseStudy_Solution.py
temp.py
CPI_CaseStudy_Solution.py*
print("Original values","\n",confusion_matrix)
# Calculating the accuracy
accuracy_score=accuracy_score(test_y, prediction)
print(accuracy_score)
print('Misclassified samples: %d' % (test_y != prediction).sum())
"""
Effect of K value on classifier
"""
Misclassified_sample = []
# Calculating error for K values between 1 and 20
for i in range(1, 20):
knn = KNeighborsClassifier(n_neighbors=i)
knn.fit(train_x, train_y)
pred_i = knn.predict(test_x)
Misclassified_sample.append((test_y != pred_i).sum())
print(Misclassified_sample)
# ==
# END OF SCRIPT
# ==
Variable explorer
Name Type Siz
Misclassified_sample list 0
accuracy_score float64 1
columns_list list 95
confusion_matrix int64 (2, 2)
correlation DataFrame (4, 4)
Help Variable explorer File explorer
IPython console
Console 1/A
...:
print(Misclassified_sample)
...: #
==========
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 344 Column: 28 Memory: 41 %
Type here to search
EN 12:30 06-09-2019


Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\WORKSPACE\PYTHON\CaseStudy_Classification
Editor - D:\WORKSPACE\PYTHON\CaseStudy_Classification\CPI_CaseStudy_Solution.py
temp.py
CPI_CaseStudy_Solution.py*
298
299 # Printing the misclassified values from prediction
300 print('Misclassified samples: %d' % (test_y != prediction).sum())
301
302 # ==
303 # KNN
304 # ==
305
306 # importing the library of KNN
307 from sklearn.neighbors import KNeighborsClassifier
308
309 # import library for plotting
310 import matplotlib.pyplot as plt
311
312
313 # Storing the K nearest neighbors classifier
314 KNN_classifier = KNeighborsClassifier(n_neighbors = 5)
315
316 # Fitting the values for X and Y
317 KNN_classifier.fit(train_x, train_y)
318
319 # Predicting the test values with model
320 prediction = KNN_classifier.predict(test_x)
321
Variable explorer
Name Type Size
accuracy_score float64 1
columns_list list 95
confusion_matrix int64 (2, 2)
correlation DataFrame (4, 4)
data DataFrame (31978, 13
Help Variable explorer File explorer
IPython console
Console 1/A
In [18]: KNN_classifier.kneighbors
Out[18]:
<bound method
KNeighborsMixin.kneighbors of
KNeighborsClassifier(algorithm='au
to', leaf_size=30,
metric='minkowski',
metric_params=None,
n_jobs=1, n_neighbors=5, p=2,
weights='uniform')>
In [19]: prediction =
KNN_classifier.predict(test_x)
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 321 Column: 1 Memory: 41 %
Type here to search
EN
12:20
06-09-2019


Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\WORKSPACE\PYTHON\CaseStudy_Classification
Editor - D:\WORKSPACE\PYTHON\CaseStudy_Classification\CPI_CaseStudy_Solution.py
temp.py
CPI_CaseStudy_Solution.py
285
286 # Make an instance of the Model
287 logistic = LogisticRegression()
288
289 # Fitting the values for x and y
290 logistic.fit(train_x,train_y)
291
292 # Prediction from test data
293 prediction = logistic.predict(test_x)
294
295 # Calculating the accuracy
296 accuracy_score=accuracy_score(test_y, prediction)
297 print(accuracy_score)
298
299 # Printing the misclassified values from prediction
300 print('Misclassified samples: %d' % (test_y != prediction).sum())
301
302 # ==
303 # KNN
304 # ==
305
306 # importing the library of KNN
307 from sklearn.neighbors import KNeighborsClassifier
Variable explorer
Name Type Size
data_income DataFrame (31978, 13)
gender DataFrame (2, 1)
gender_salstat DataFrame (3, 2)
missing DataFrame (1816, 13)
summary_cate DataFrame (4, 9)
Help Variable explorer File explorer
IPython console
Console 1/A
In [7]: from sklearn.neighbors
import KNeighborsClassifier
In [8]:
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 310 Column: 22 Memory: 40 %
Type here to search
EN 12:15 06-09-2019


Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\WORKSPACE\PYTHON\CaseStudy_Classification
Editor - D:\WORKSPACE\PYTHON\CaseStudy_Classification\CPI_CaseStudy_Solution.py
temp.py
CPI_CaseStudy_Solution.py*
Python_KNN.py
y=new_data['SalStat'].values
print(y)
# Storing the values from input features
x = new_data[features].values
print(x)
# Splitting the data into train and test
train_x,test_x,train_y,test_y = train_test_split(x,y,test_size=0.
# Make an instance of the Model
logistic = LogisticRegression()
# Fitting the values for x and y
logistic.fit(train_x,train_y)
# Prediction from test data
prediction = logistic.predict(test_x)
# Calculating the accuracy
accuracy_score=accuracy_score(test_y, prediction)
print(accuracy_score)
Variable explorer
Name Type Size Value
accuracy_score float64 1 0.…
cols list 4 ['…
columns_list list 44 ['…
correlation DataFrame (4, 4) Co…
data DataFrame (31978, 13) Co…
data2 DataFrame (30162, 13) Co
Help Variable explorer File explorer
IPython console
Console 1/A
In [104]:
In [104]: print(accuracy_score)
0.8388772240026522
In [105]:
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 298 Column: 1 Memory: 37 %
Type here to search
EN
10:47
06-09-2019


Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\WORKSPACE\PYTHON\CaseStudy_Classification
Editor - D:\WORKSPACE\PYTHON\CaseStudy_Classification\CPI_CaseStudy_Solution.py
temp.py
CPI_CaseStudy_Solution.py*
Python_KNN.py
columns_list=list(new_data.columns)
print(columns_list)
# Separating the input names from data
features=list(set(columns_list)-set(['SalStat']))
print(features)
# Storing the output values in y
y=new_data['SalStat'].values
print(y)
# Storing the values from input features
x = new_data[features].values
print(x)
# Splitting the data into train and test
train_x,test_x,train_y,test_y = train_test_split(x,y,test_size=0.3, random_state
# Make an instance of the Model
logistic = LogisticRegression()
# Fitting the values for x and y
logistic.fit(train_x,train_y)
Variable explorer
Name
Type
cols
list
columns_list
list
correlation
DataFrame
data
DataFrame
data2
DataFrame
Help
Variable explorer
File explorer
IPython console
Console 1/A
[0 0 1 ... 0 0 0]]
In [91]:
train_x,test_x,train_y,
test_y =
train_test_split(x,y,te
st_size=0.3,
random_state=0)
In [92]: logistic =
LogisticRegression()
In [93]:
Permissions: RW
End-of-lines: CRLF
Encoding: UTF-8
Line: 290
Column: 17
Memory: 37 %
Type here to search
10:45
06-09-2019