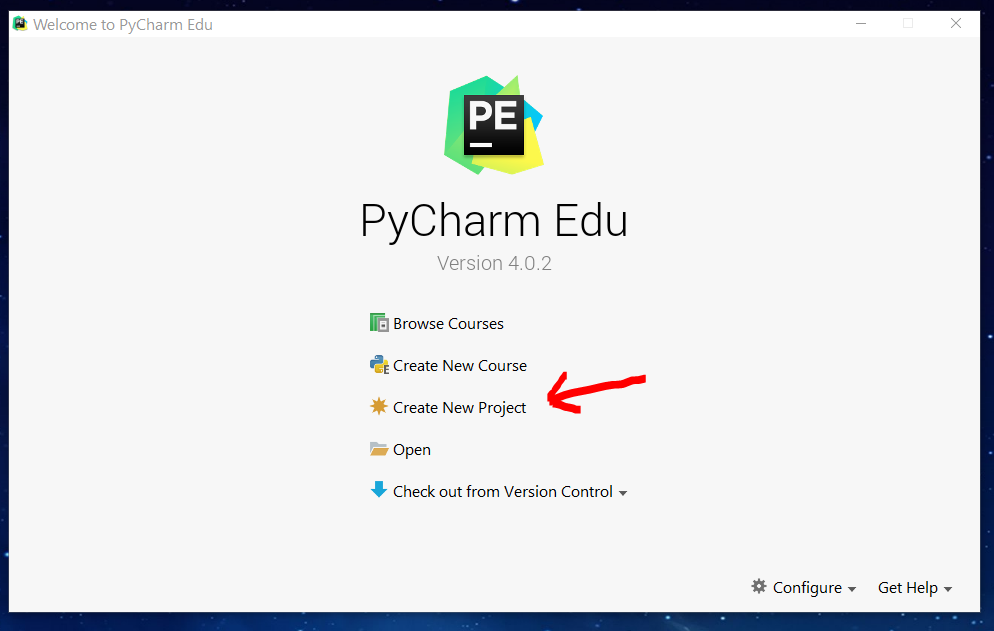
**Create BAS test project in python**

Expeiment = Titanic 1 [Predictive Exp.]

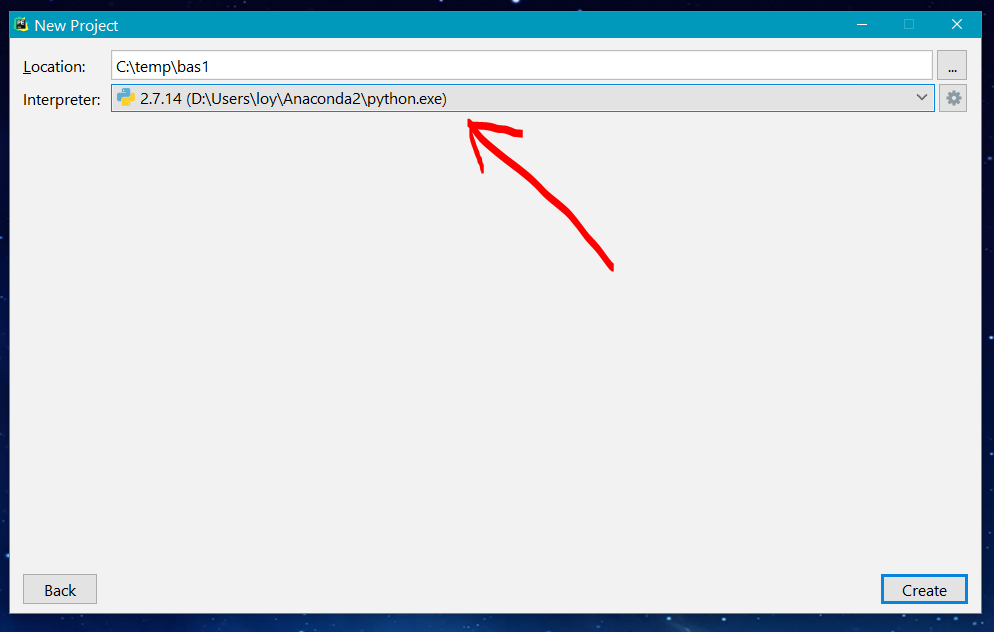
Python version = 2.7

IDE = Pycharm

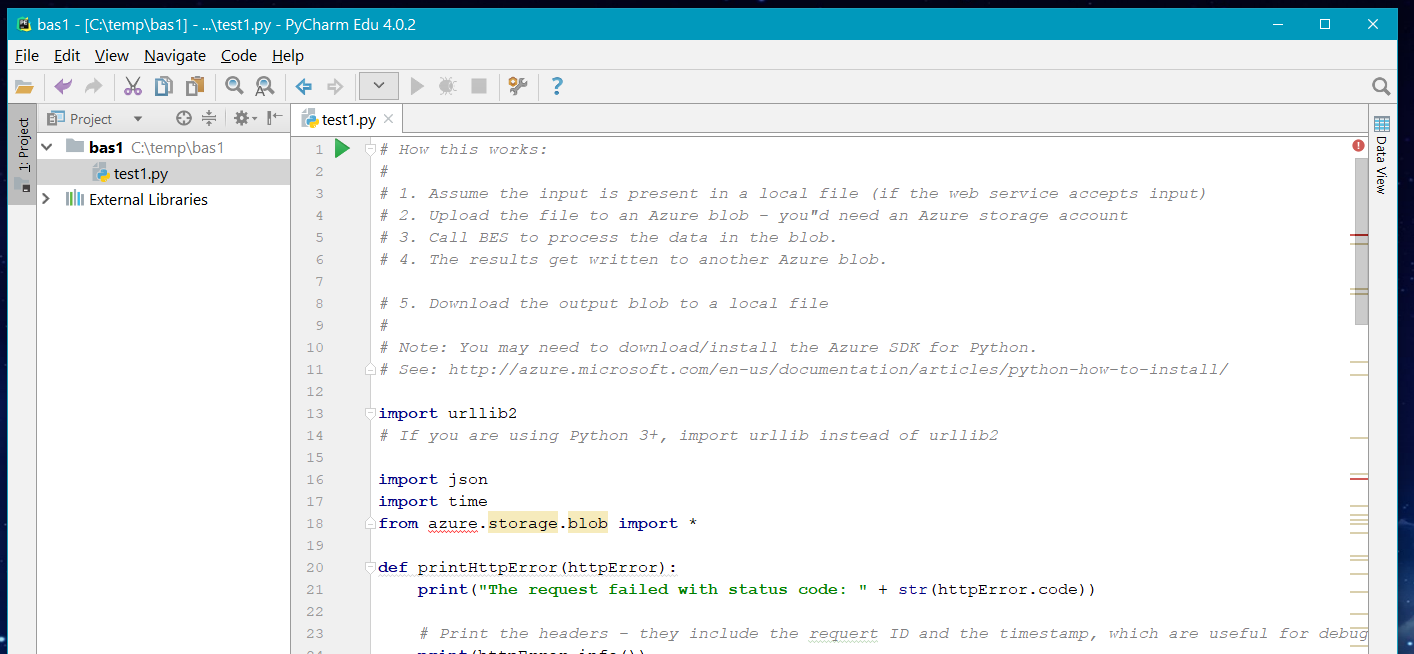
Create new project

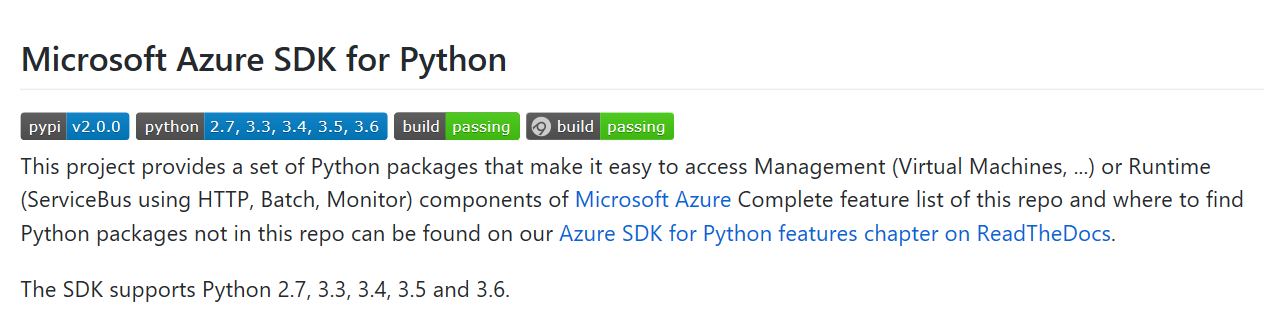


Create project bas1 in python 2.7



Add python file / add python code from BAS api sample page





Install Azure SDK using PIP

1. Go to interpreter folder
2. Open CMD or Power shell
3. ***pip install azure-storage-blob*** # Install the latest Storage management library

OR

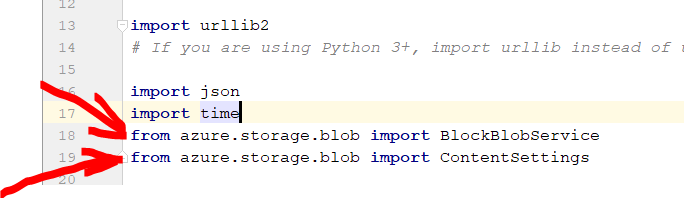
***pip install azure*** # install all package, take longer

<https://github.com/Azure/azure-sdk-for-python>

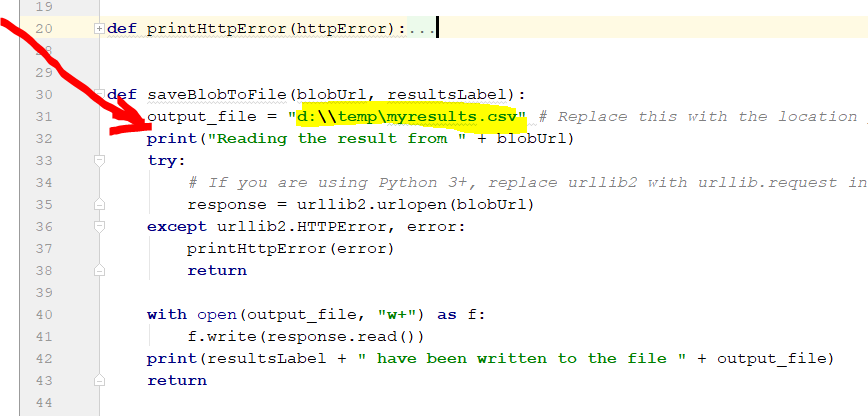
\*\*\* if package went to wrong folder, just copy all azure pages to the correct folder

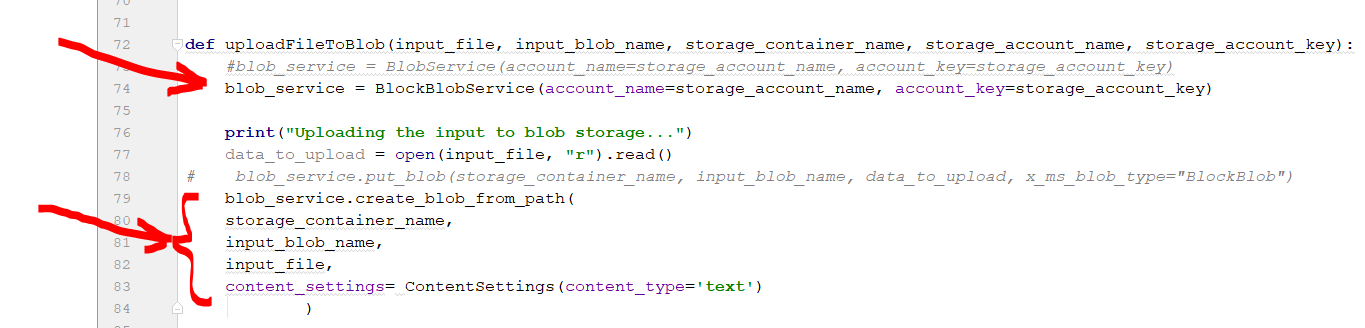
D:\Users\loy\Anaconda2\Lib\site-packages

Change from blob to BlockBlobService

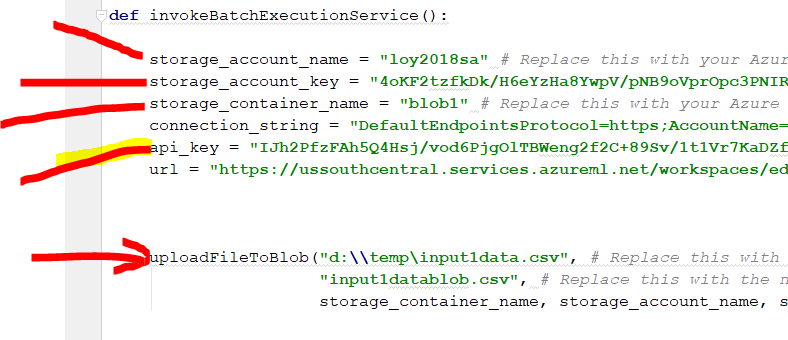


Set output file





Set account key



Input file csv

Need column title. No new line at the last letter (don’t press enter).

Save this file to d:\\temp\input1data.csv

PassengerId,Survived,Pclass,Name,Sex,Age,SibSp,Parch,Ticket,Fare,Cabin,Embarked

1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,,S

2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C

3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,,S

Output result file written in to hard disk

Survived,PassengerClass,Gender,Age,SiblingSpouse,ParentChild,FarePrice,PortEmbarkation,Scored Labels,Scored Probabilities

0,3,male,22,1,0,7.25,S,0,1.76168905454688E-05

1,1,female,38,1,0,71.2833,C,1,0.999995827674866

1,3,female,26,0,0,7.925,S,1,0.979553580284119

*# How this works:  
#  
# 1. Assume the input is present in a local file (if the web service accepts input)  
# 2. Upload the file to an Azure blob - you"d need an Azure storage account  
# 3. Call BES to process the data in the blob.  
# 4. The results get written to another Azure blob.  
  
# 5. Download the output blob to a local file  
#  
# Note: You may need to download/install the Azure SDK for Python.  
# See: http://azure.microsoft.com/en-us/documentation/articles/python-how-to-install/***import** urllib2  
*# If you are using Python 3+, import urllib instead of urllib2***import** json  
**import** time  
**from** azure.storage.blob **import** BlockBlobService  
**from** azure.storage.blob **import** ContentSettings  
  
**def** printHttpError(httpError):  
 **print**(**"The request failed with status code: "** + str(httpError.code))  
  
 *# Print the headers - they include the requert ID and the timestamp, which are useful for debugging the failure* **print**(httpError.info())  
  
 **print**(json.loads(httpError.read()))  
 **return  
  
  
def** saveBlobToFile(blobUrl, resultsLabel):  
 output\_file = **"d:\\temp\myresults.csv"** *# Replace this with the location you would like to use for your output file* **print**(**"Reading the result from "** + blobUrl)  
 **try**:  
 *# If you are using Python 3+, replace urllib2 with urllib.request in the following code* response = urllib2.urlopen(blobUrl)  
 **except** urllib2.HTTPError, error:  
 printHttpError(error)  
 **return  
  
 with** open(output\_file, **"w+"**) **as** f:  
 f.write(response.read())  
 **print**(resultsLabel + **" have been written to the file "** + output\_file)  
 **return  
  
  
def** processResults(result):  
  
  
 first = True  
 results = result[**"Results"**]  
 **for** outputName **in** results:  
 result\_blob\_location = results[outputName]  
 sas\_token = result\_blob\_location[**"SasBlobToken"**]  
 base\_url = result\_blob\_location[**"BaseLocation"**]  
 relative\_url = result\_blob\_location[**"RelativeLocation"**]  
  
 **print**(**"The results for "** + outputName + **" are available at the following Azure Storage location:"**)  
 **print**(**"BaseLocation: "** + base\_url)  
 **print**(**"RelativeLocation: "** + relative\_url)  
 **print**(**"SasBlobToken: "** + sas\_token)  
  
  
 **if** (first):  
 first = False  
 url3 = base\_url + relative\_url + sas\_token  
 saveBlobToFile(url3, **"The results for "** + outputName)  
 **return  
  
  
  
def** uploadFileToBlob(input\_file, input\_blob\_name, storage\_container\_name, storage\_account\_name, storage\_account\_key):  
 *#blob\_service = BlobService(account\_name=storage\_account\_name, account\_key=storage\_account\_key)* blob\_service = BlockBlobService(account\_name=storage\_account\_name, account\_key=storage\_account\_key)  
  
 **print**(**"Uploading the input to blob storage..."**)  
 data\_to\_upload = open(input\_file, **"r"**).read()  
*# blob\_service.put\_blob(storage\_container\_name, input\_blob\_name, data\_to\_upload, x\_ms\_blob\_type="BlockBlob")* blob\_service.create\_blob\_from\_path(  
 storage\_container\_name,  
 input\_blob\_name,  
 input\_file,  
 content\_settings= ContentSettings(content\_type=**'text'**)  
 )  
  
**def** invokeBatchExecutionService():  
  
 storage\_account\_name = **"loy2018sa"** *# Replace this with your Azure Storage Account name* storage\_account\_key = **"4oKF2tzfkDk/H6eYzHa8YwpV/pNB9oVprOpc3PNIRrL/EduRP6/o2css1tX4p47ateS8AfT2DUetjgLv4Tr3hg=="** *# Replace this with your Azure Storage Key* storage\_container\_name = **"blob1"** *# Replace this with your Azure Storage Container name* connection\_string = **"DefaultEndpointsProtocol=https;AccountName="** + storage\_account\_name + **";AccountKey="** + storage\_account\_key  
 api\_key = **"IJh2PfzFAh5Q4Hsj/vod6PjgOlTBWeng2f2C+89Sv/1t1Vr7KaDZfequmXPzhAZNs9KjkaklAcSuRvTLy47/yw=="** *# Replace this with the API key for the web service* url = **"https://ussouthcentral.services.azureml.net/workspaces/ede12cb3aaf24c7e826493f4e309f1e1/services/ad3b577804c443d08f0f30b6c8028411/jobs"** uploadFileToBlob(**"d:\\temp\input1data.csv"**, *# Replace this with the location of your input file* **"input1datablob.csv"**, *# Replace this with the name you would like to use for your Azure blob; this needs to have the same extension as the input file* storage\_container\_name, storage\_account\_name, storage\_account\_key)  
  
 payload = {  
  
 **"Inputs"**: {  
  
 **"input1"**: { **"ConnectionString"**: connection\_string, **"RelativeLocation"**: **"/"** + storage\_container\_name + **"/input1datablob.csv"** },  
 },  
  
 **"Outputs"**: {  
  
 **"output1"**: { **"ConnectionString"**: connection\_string, **"RelativeLocation"**: **"/"** + storage\_container\_name + **"/output1results.csv"** },  
 },  
 **"GlobalParameters"**: {  
}  
 }  
  
 body = str.encode(json.dumps(payload))  
 headers = { **"Content-Type"**:**"application/json"**, **"Authorization"**:(**"Bearer "** + api\_key)}  
 **print**(**"Submitting the job..."**)  
  
 *# If you are using Python 3+, replace urllib2 with urllib.request in the following code  
  
 # submit the job* req = urllib2.Request(url + **"?api-version=2.0"**, body, headers)  
 **try**:  
 response = urllib2.urlopen(req)  
 **except** urllib2.HTTPError, error:  
 printHttpError(error)  
 **return** result = response.read()  
 job\_id = result[1:-1] *# remove the enclosing double-quotes* **print**(**"Job ID: "** + job\_id)  
  
  
 *# If you are using Python 3+, replace urllib2 with urllib.request in the following code  
 # start the job* **print**(**"Starting the job..."**)  
 req = urllib2.Request(url + **"/"** + job\_id + **"/start?api-version=2.0"**, **""**, headers)  
 **try**:  
 response = urllib2.urlopen(req)  
 **except** urllib2.HTTPError, error:  
 printHttpError(error)  
 **return** url2 = url + **"/"** + job\_id + **"?api-version=2.0"  
  
 while** True:  
 **print**(**"Checking the job status..."**)  
 *# If you are using Python 3+, replace urllib2 with urllib.request in the follwing code* req = urllib2.Request(url2, headers = { **"Authorization"**:(**"Bearer "** + api\_key) })  
  
 **try**:  
 response = urllib2.urlopen(req)  
 **except** urllib2.HTTPError, error:  
 printHttpError(error)  
 **return** result = json.loads(response.read())  
 status = result[**"StatusCode"**]  
 **if** (status == 0 **or** status == **"NotStarted"**):  
 **print**(**"Job "** + job\_id + **" not yet started..."**)  
 **elif** (status == 1 **or** status == **"Running"**):  
 **print**(**"Job "** + job\_id + **" running..."**)  
 **elif** (status == 2 **or** status == **"Failed"**):  
 **print**(**"Job "** + job\_id + **" failed!"**)  
 **print**(**"Error details: "** + result[**"Details"**])  
 **break  
 elif** (status == 3 **or** status == **"Cancelled"**):  
 **print**(**"Job "** + job\_id + **" cancelled!"**)  
 **break  
 elif** (status == 4 **or** status == **"Finished"**):  
 **print**(**"Job "** + job\_id + **" finished!"**)  
  
 processResults(result)  
 **break** time.sleep(1) *# wait one second* **return**invokeBatchExecutionService()

https://docs.microsoft.com/en-us/azure/storage/blobs/storage-python-how-to-use-blob-storage