National Research University
"Higher School of Economics"

Graduate School of Business

Report

Practice 1. Bike Share Demand Forecasting with MS Azure Machine Learning

---

Area of studies: Business Informatics

Master's programme: Business Analytics and Big Data Systems

Completed by student: Akmal Kholikulov

Group number: BASB 212

Instructor: Yulia Morozova

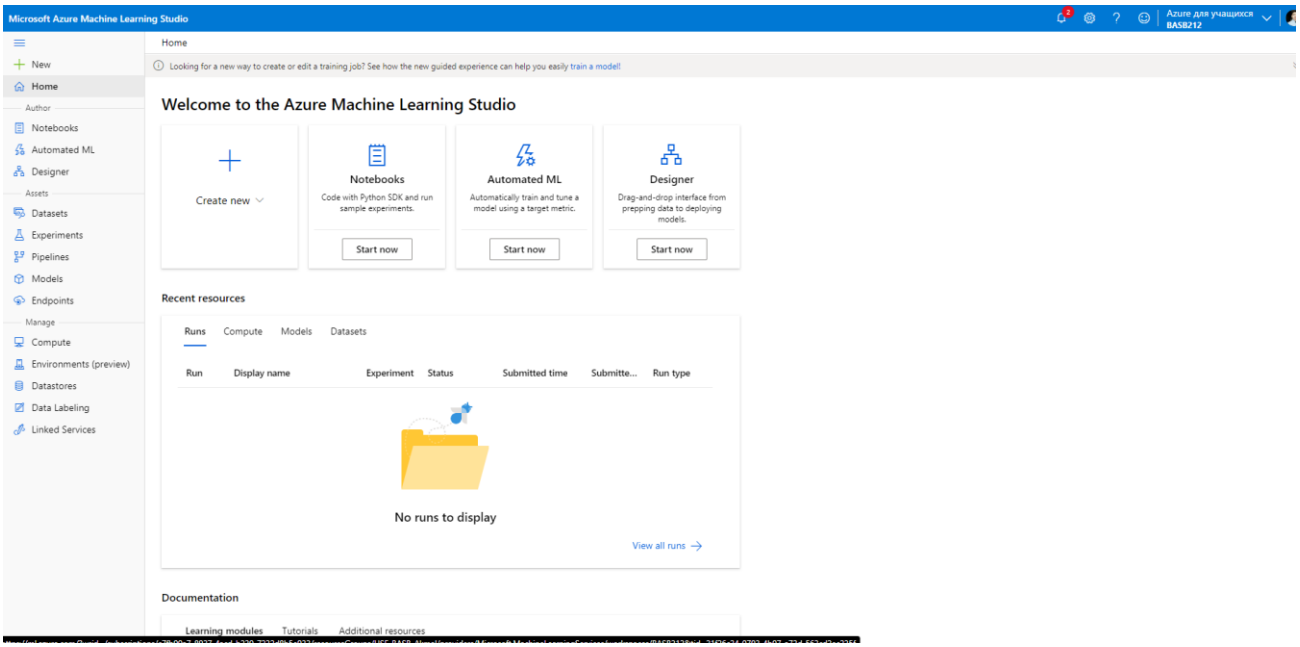
Automated Machine Learning

BikeShare Demand Forecasting

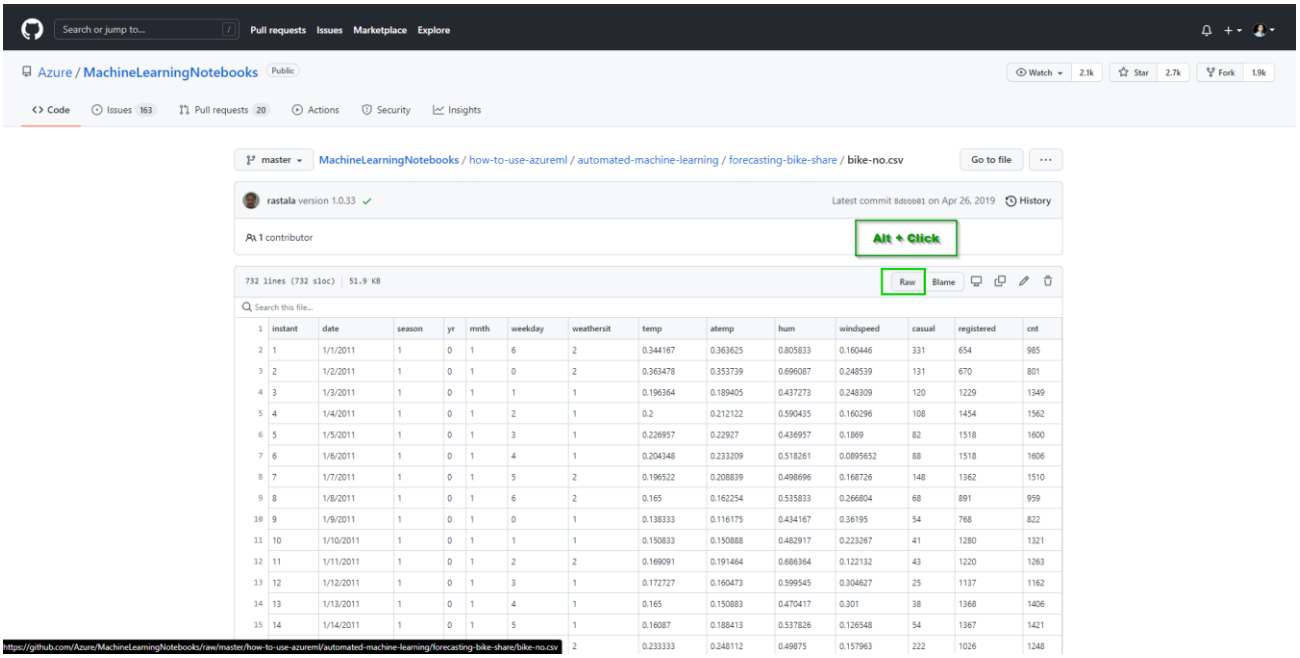
Content

- Create and load a dataset.
- Configure and run an automated ML experiment.
- Specify forecasting settings.
- Explore the experiment results.
- Deploy the best model.

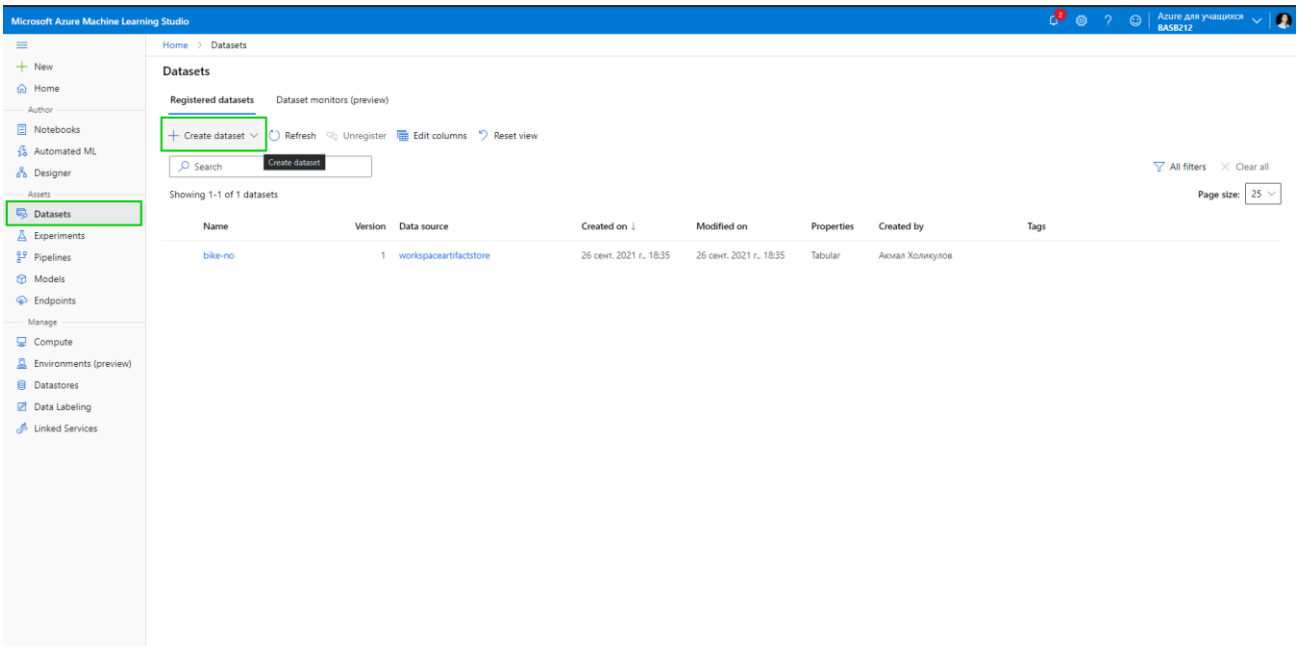
Step 1. We need to register in MS Azure create an account, an Azure Machine Learning workspace and so on.



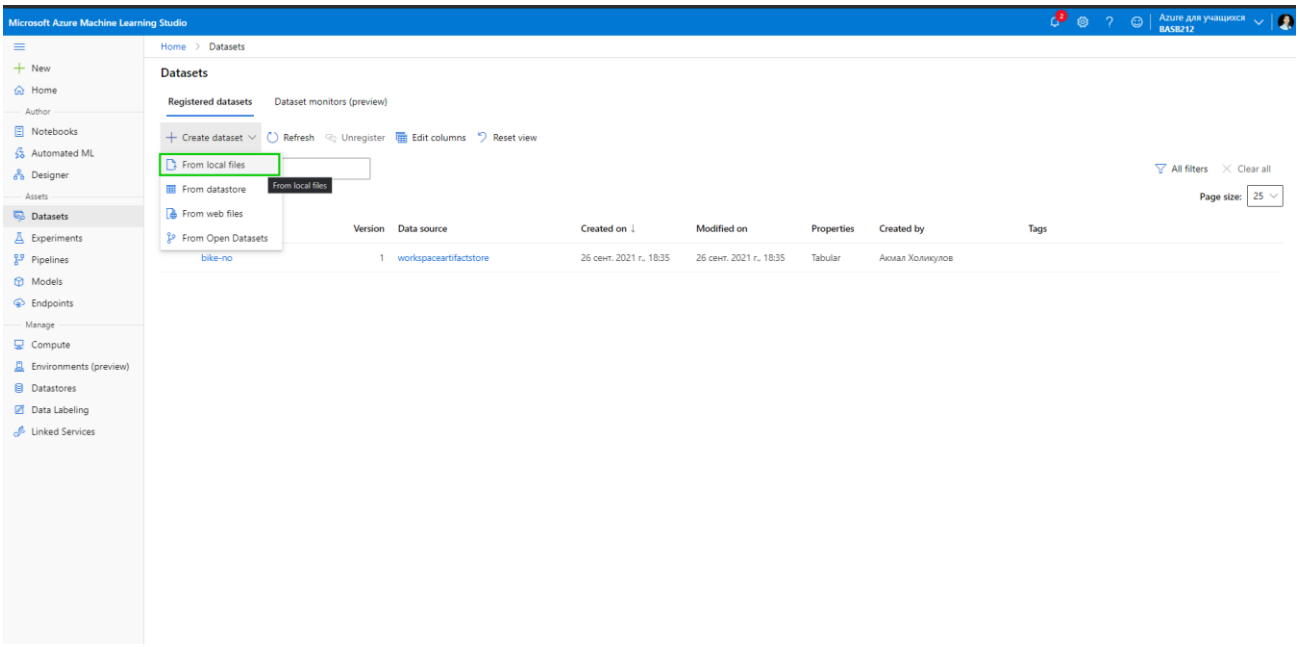
Step 2. Download datasets from github. In order to download dataset press “Alt” + Click on “Raw” simultaneously. (file will be downloaded in .txt format)



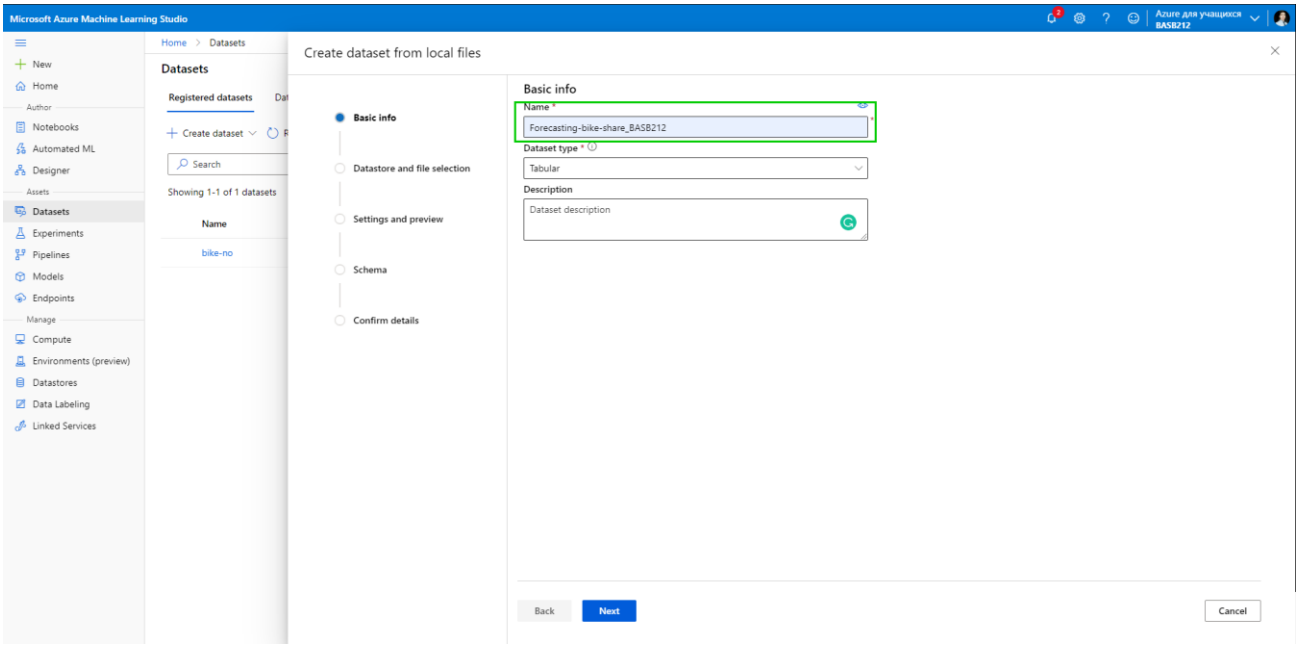
Step 3. After downloading datasets. Go back to Microsoft Azure Machine Learning Studio and Click on “+ Create dataset”



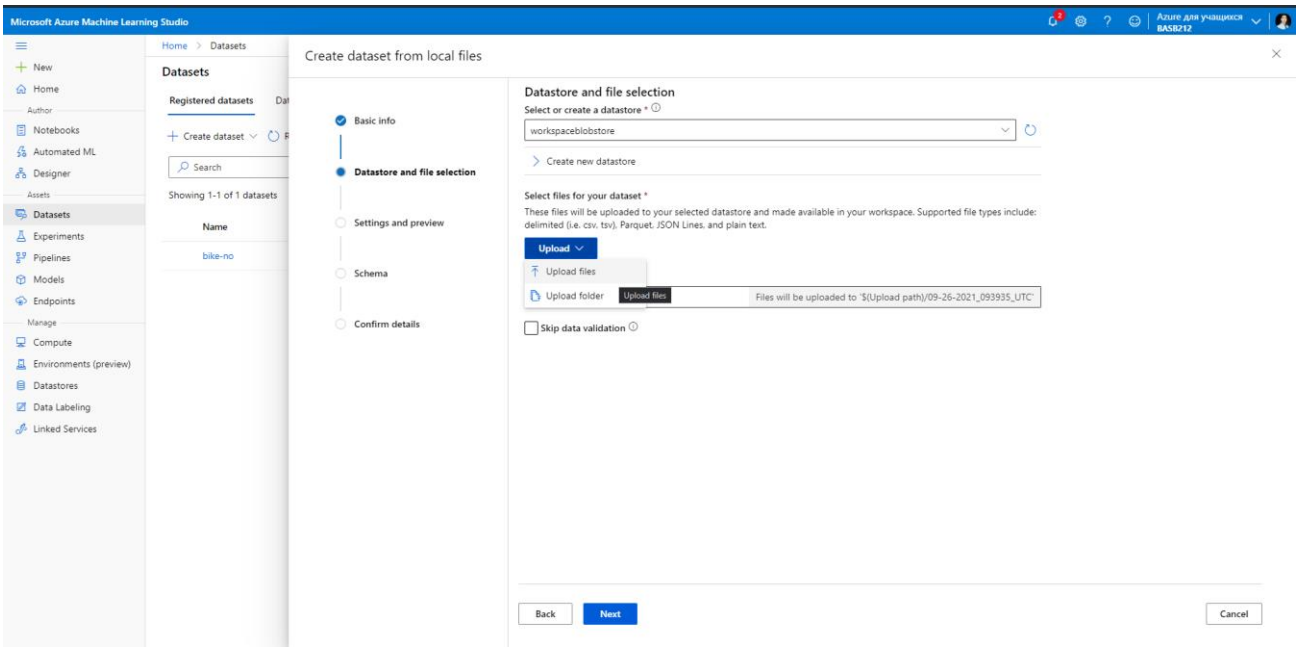
Step 4. After then, click on “From local files”



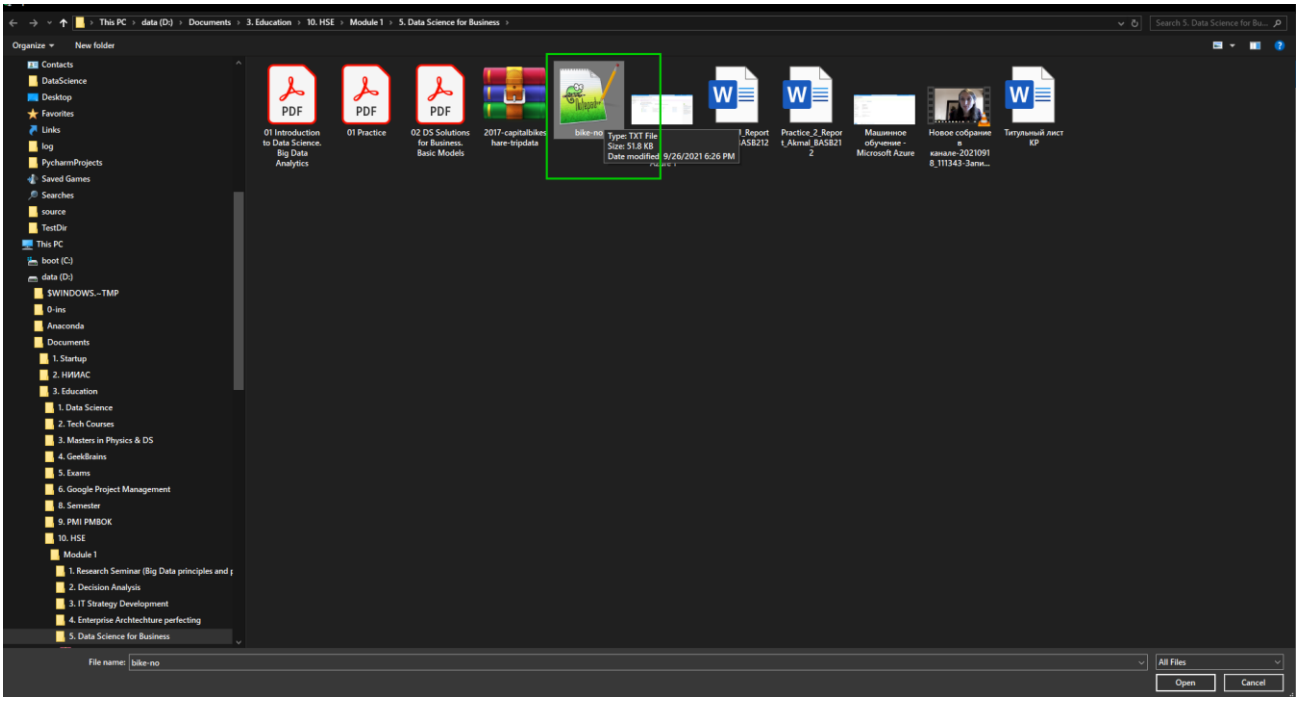
Step 5. Provide basic information. Text a name in raw “Forecasting-bike-share_BASB212”. Do not change dataset type, remain it “Tabular”. Then click “Next”



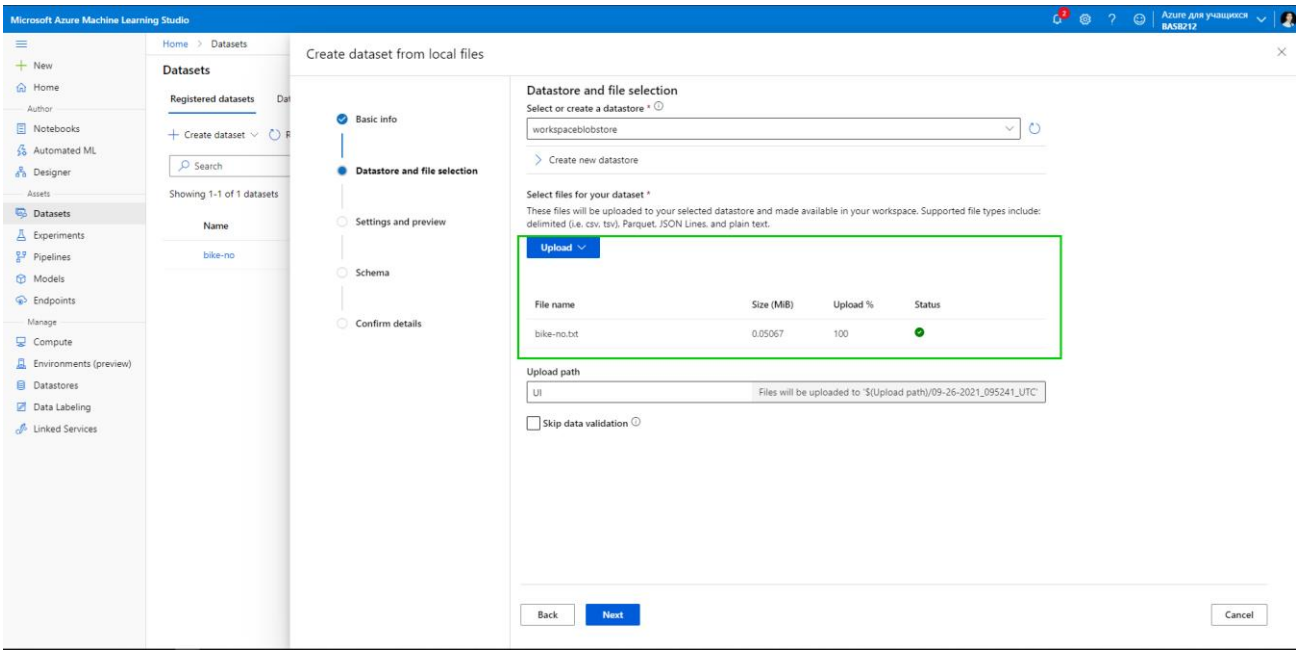
Step 6. To upload a file, click on “Upload” then “Upload files”. Then click “Next”



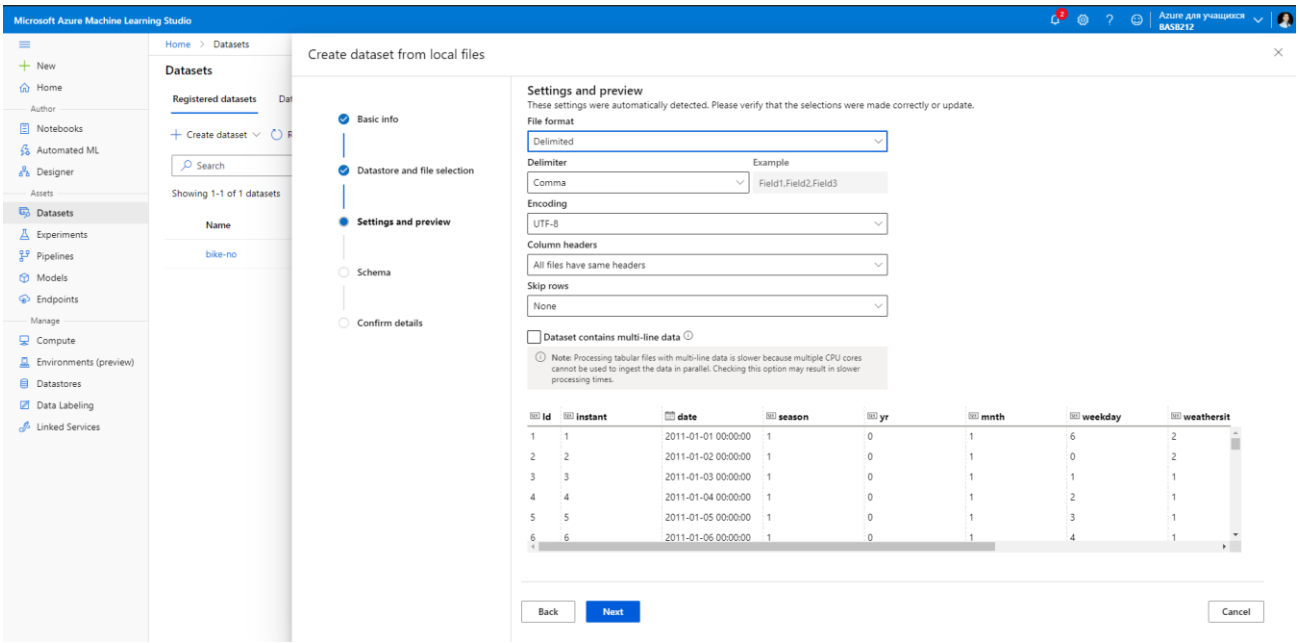
Step 7. Choose the downloaded dataset in TXT format “bike-no”. Then click “Open”



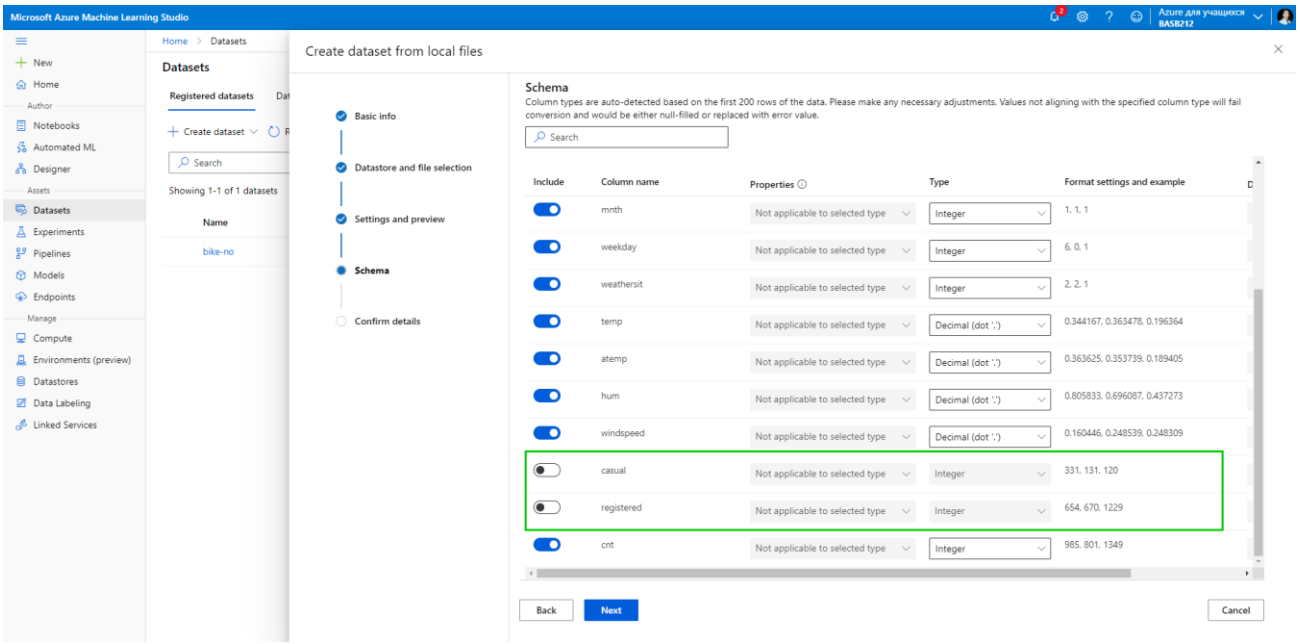
Step 8. We see that dataset “bike-no.txt” is uploaded. Then click “Next”



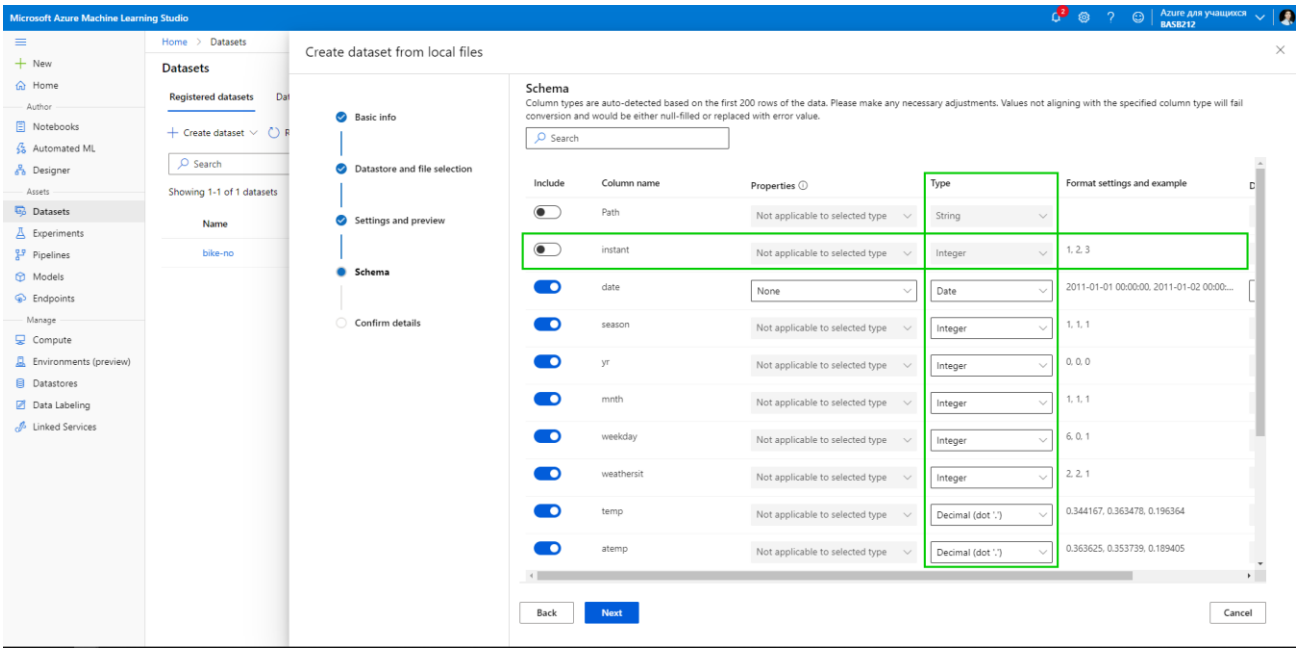
Step 9. We do not need to change anything here. Just click “Next”



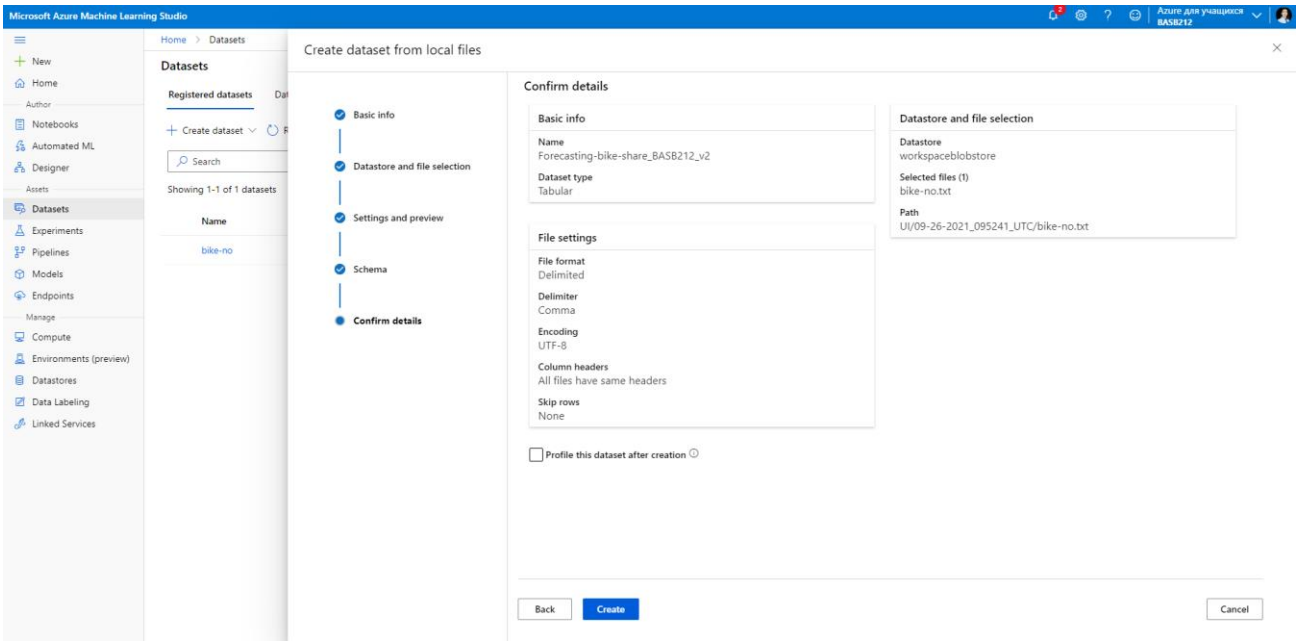
Step 10. We do not need to include in Schema “casual” and “registered”.



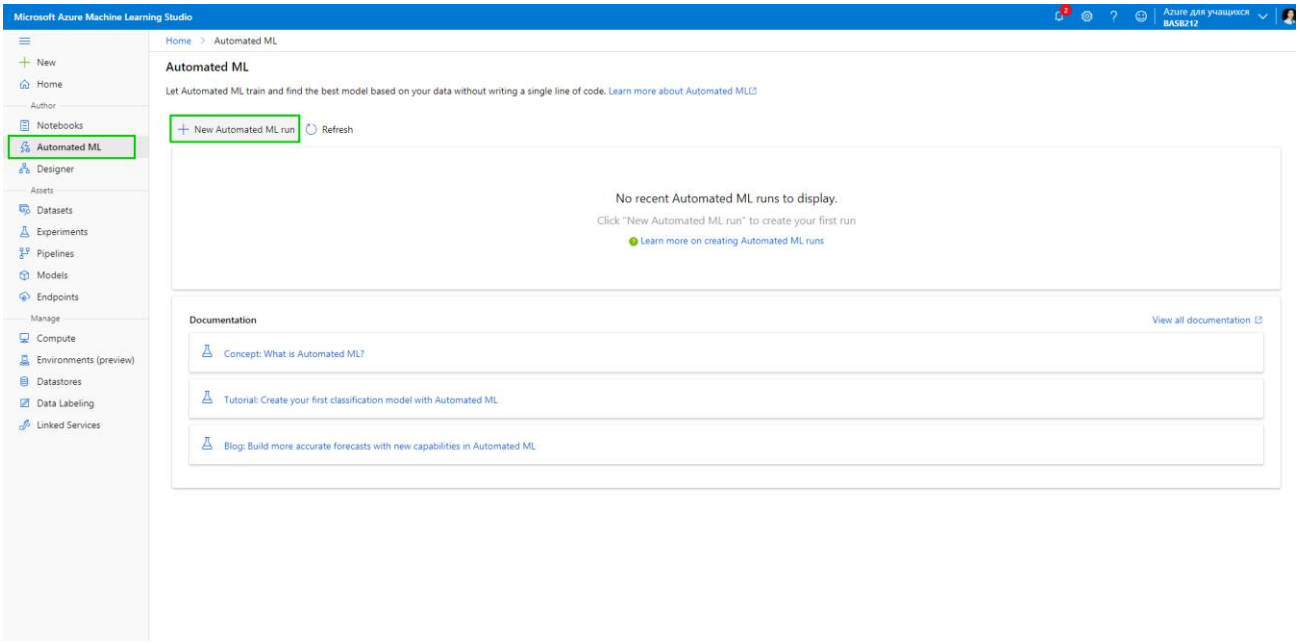
Also do not include “Instant” column. Check the “type”. If all is okay, then click “Next”



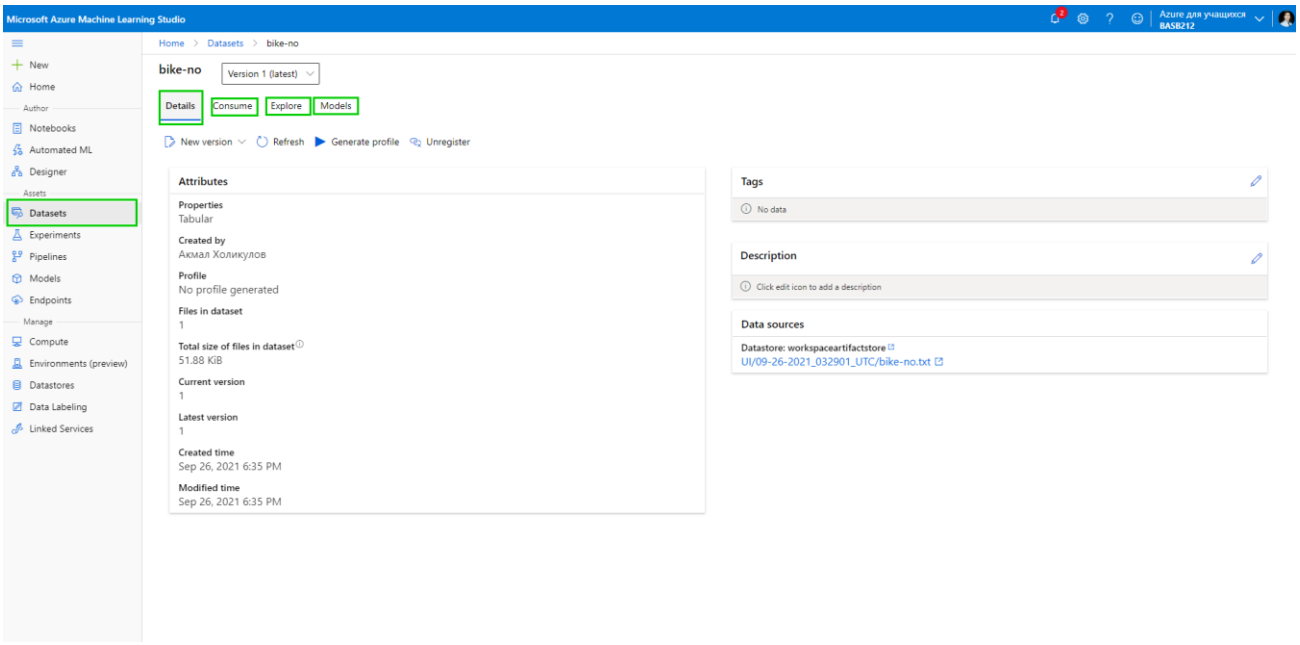
Step 11. Once check again and confirm by clicking “Create”.

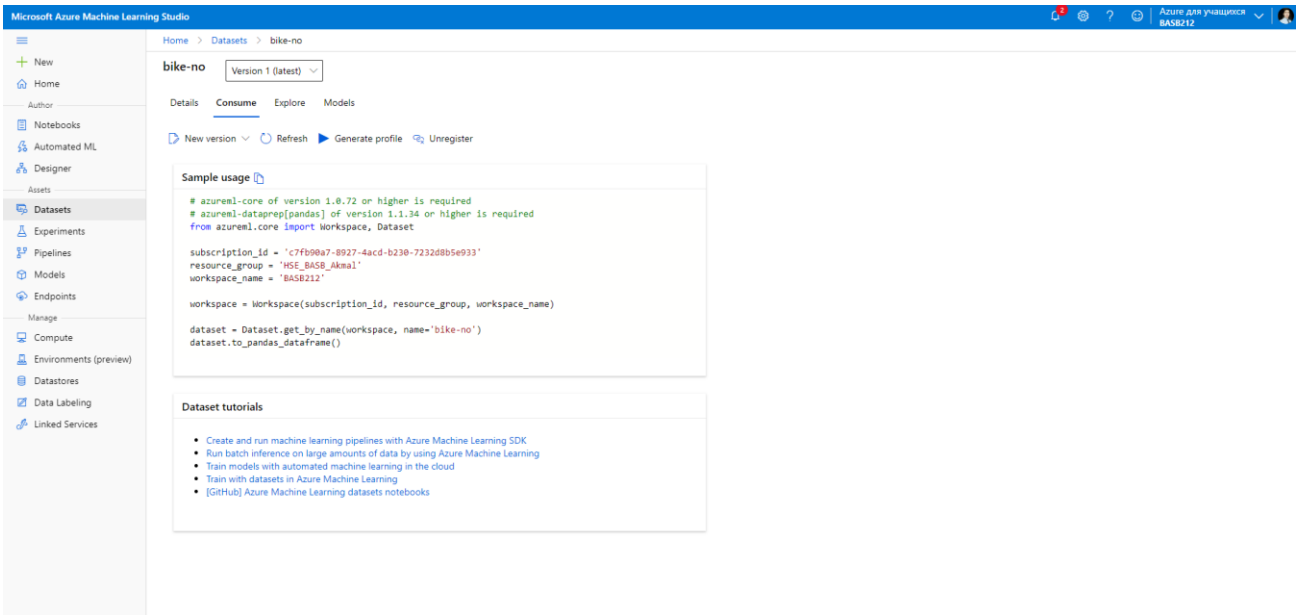


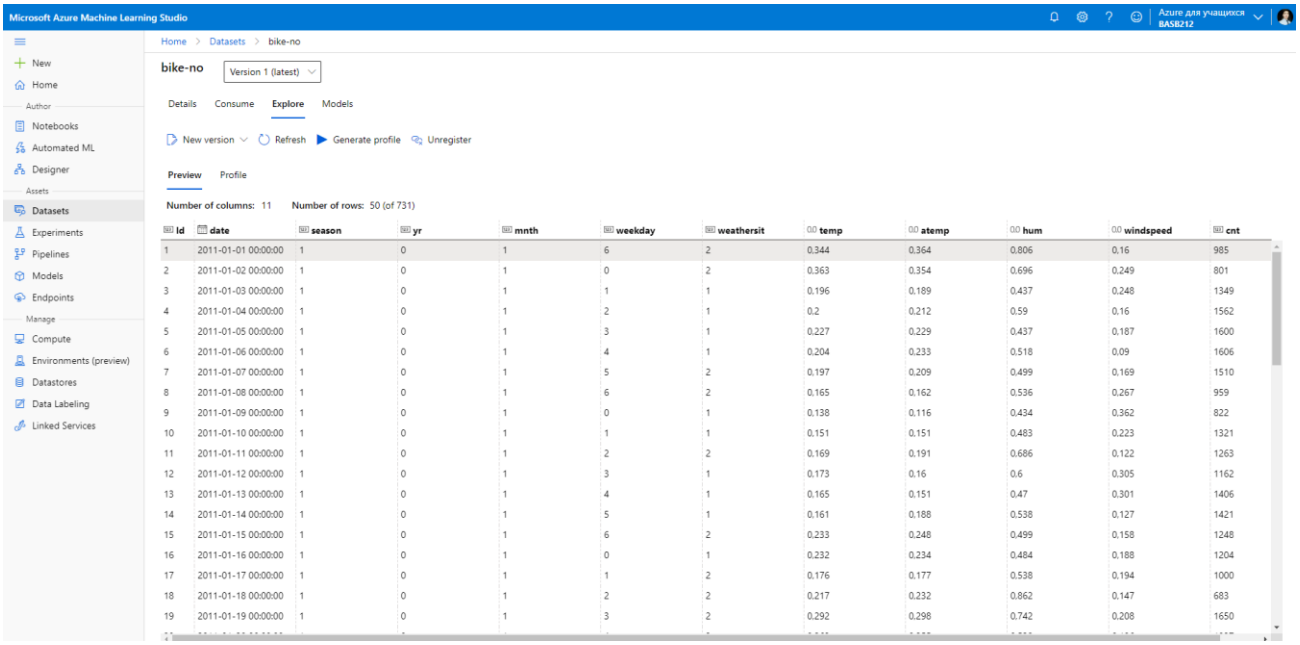
Step 12. Go to the “Automated ML” (left side of the screen). Click on “+ New automated ML run”



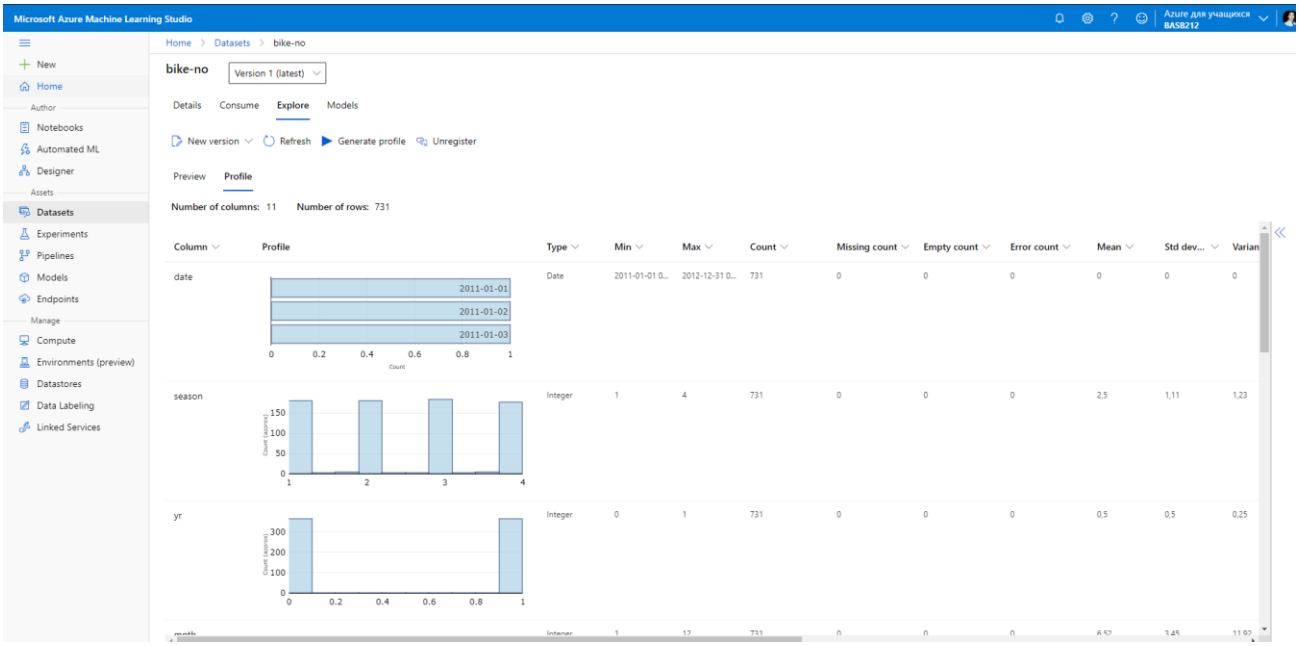
Step 13. In the meanwhile, we can see our dataset by clicking to “Datasets”. We can see “Details”, “Consume”, “Explore” and so on.





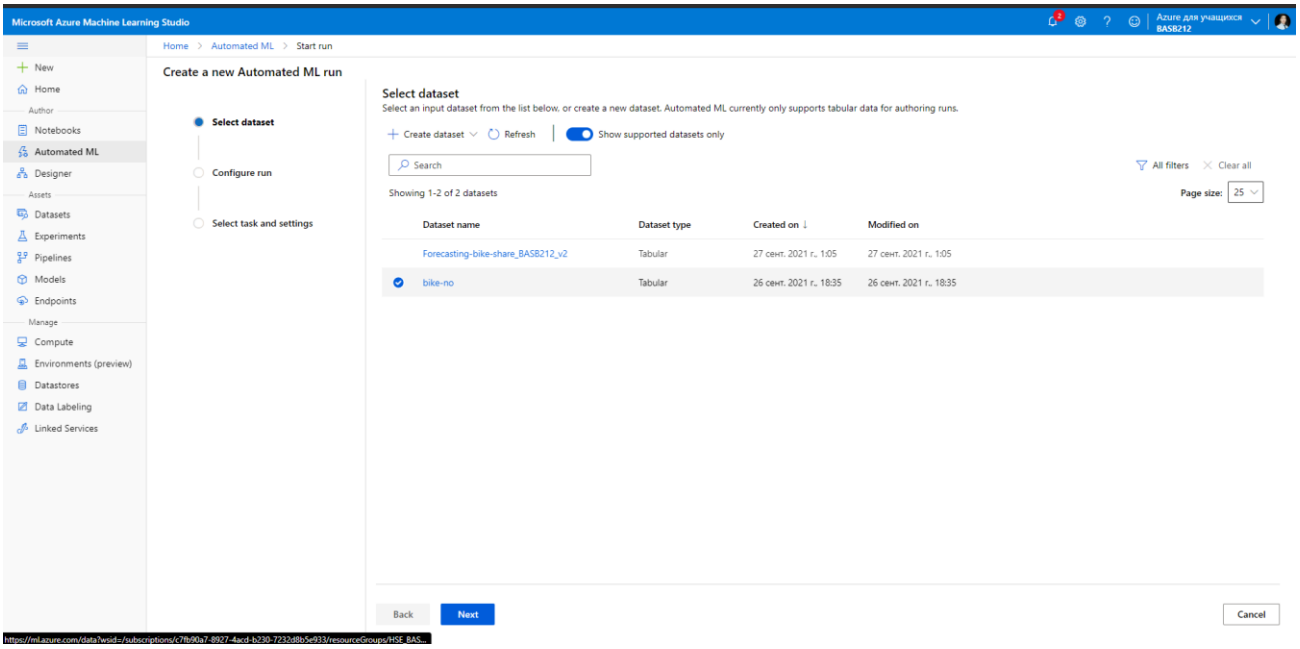


Id	date	season	yr	mnth	weekday	weathersit	temp	atemp	hum	windspeed	cnt
1	2011-01-01 00:00:00	1	0	1	6	2	0,344	0,364	0,806	0,16	985
2	2011-01-02 00:00:00	1	0	1	0	2	0,363	0,354	0,696	0,249	801
3	2011-01-03 00:00:00	1	0	1	1	1	0,196	0,189	0,437	0,248	1349
4	2011-01-04 00:00:00	1	0	1	2	1	0,2	0,212	0,59	0,16	1562
5	2011-01-05 00:00:00	1	0	1	3	1	0,227	0,229	0,437	0,187	1600
6	2011-01-06 00:00:00	1	0	1	4	1	0,204	0,233	0,518	0,09	1606
7	2011-01-07 00:00:00	1	0	1	5	2	0,197	0,209	0,499	0,169	1510
8	2011-01-08 00:00:00	1	0	1	6	2	0,165	0,162	0,536	0,267	959
9	2011-01-09 00:00:00	1	0	1	0	1	0,138	0,116	0,434	0,362	822
10	2011-01-10 00:00:00	1	0	1	1	1	0,151	0,151	0,483	0,223	1321
11	2011-01-11 00:00:00	1	0	1	2	2	0,169	0,191	0,686	0,122	1263
12	2011-01-12 00:00:00	1	0	1	3	1	0,173	0,16	0,6	0,305	1162
13	2011-01-13 00:00:00	1	0	1	4	1	0,165	0,151	0,47	0,301	1406
14	2011-01-14 00:00:00	1	0	1	5	1	0,161	0,188	0,538	0,127	1421
15	2011-01-15 00:00:00	1	0	1	6	2	0,233	0,248	0,499	0,158	1248
16	2011-01-16 00:00:00	1	0	1	0	1	0,232	0,234	0,484	0,188	1204
17	2011-01-17 00:00:00	1	0	1	1	2	0,176	0,177	0,538	0,194	1000
18	2011-01-18 00:00:00	1	0	1	2	2	0,217	0,232	0,862	0,147	683
19	2011-01-19 00:00:00	1	0	1	3	2	0,292	0,298	0,742	0,208	1650



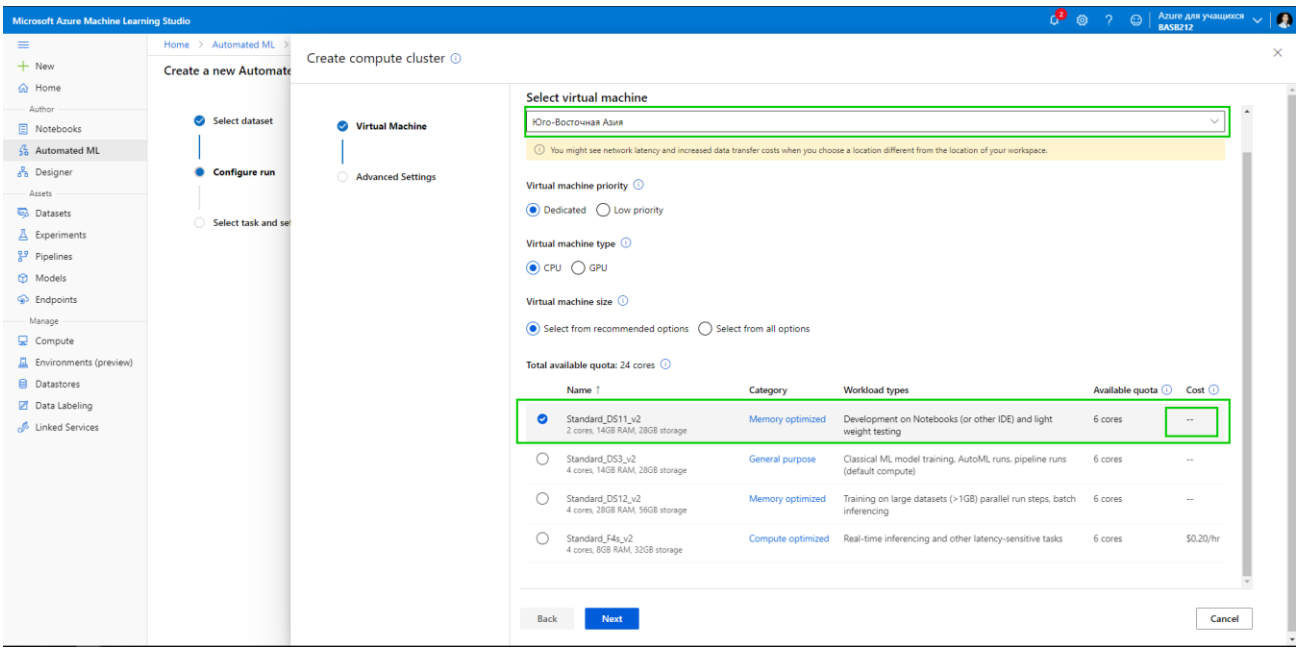
Column	Profile	Type	Min	Max	Count	Missing count	Empty count	Error count	Mean	Std dev...	Varian
date	2011-01-01 2011-01-02 2011-01-03	Date	2011-01-01 0...	2012-12-31 0...	731	0	0	0	0	0	0
season		Integer	1	4	731	0	0	0	2,5	1,11	1,23
yr		Integer	0	1	731	0	0	0	0,5	0,5	0,25
mnth		Integer	1	12	731	0	0	0	6,52	3,45	11,92

Step 14. Turn back to “Automated ML”, then click on dataset “bike-no”. Click “Next”.

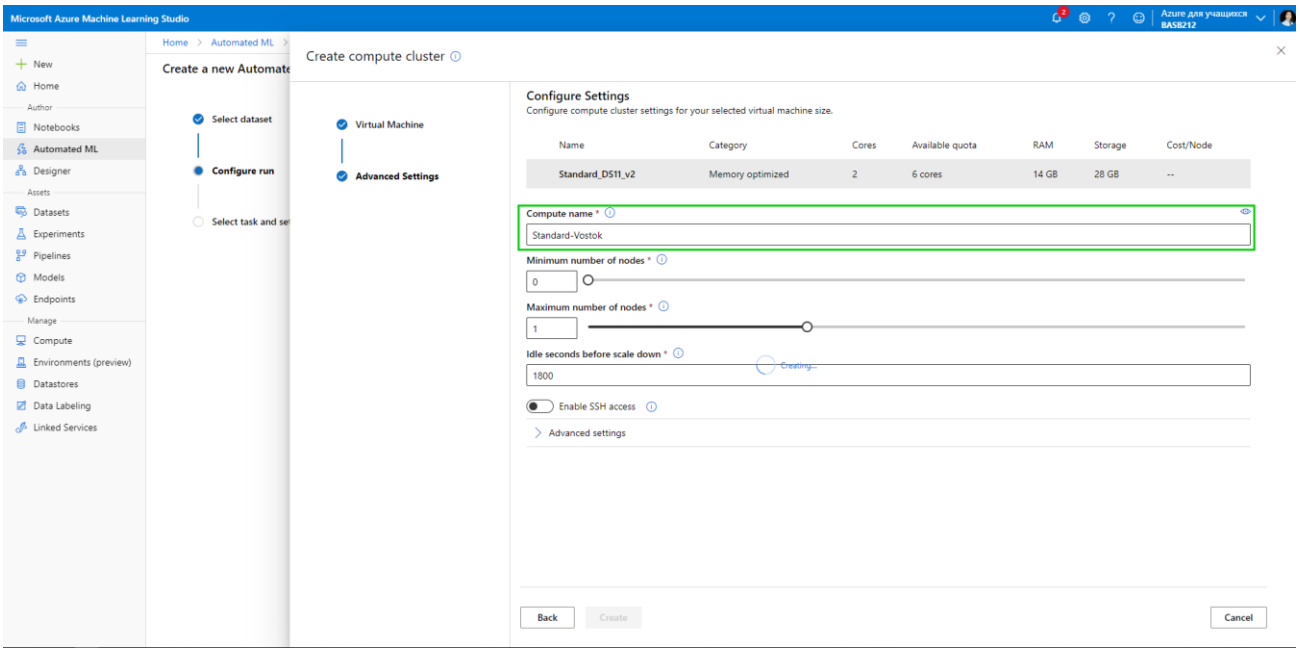


Step 15. New experimental name “ML_bakeshare_prediction_Akmal”. Target column “cnt (Integer)”.

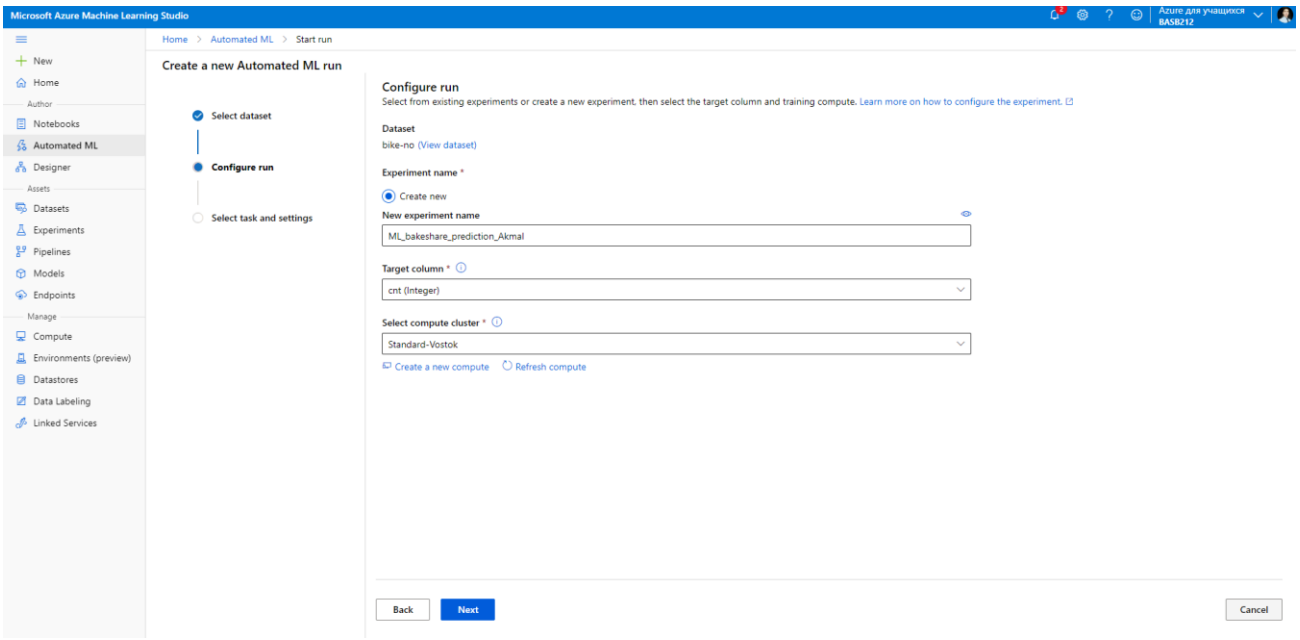
Create a new compute cluster. Click on “Create on new compute”. Find out virtual machine that is free of charge and with minimum technical parameters, since we have not a big dataset.

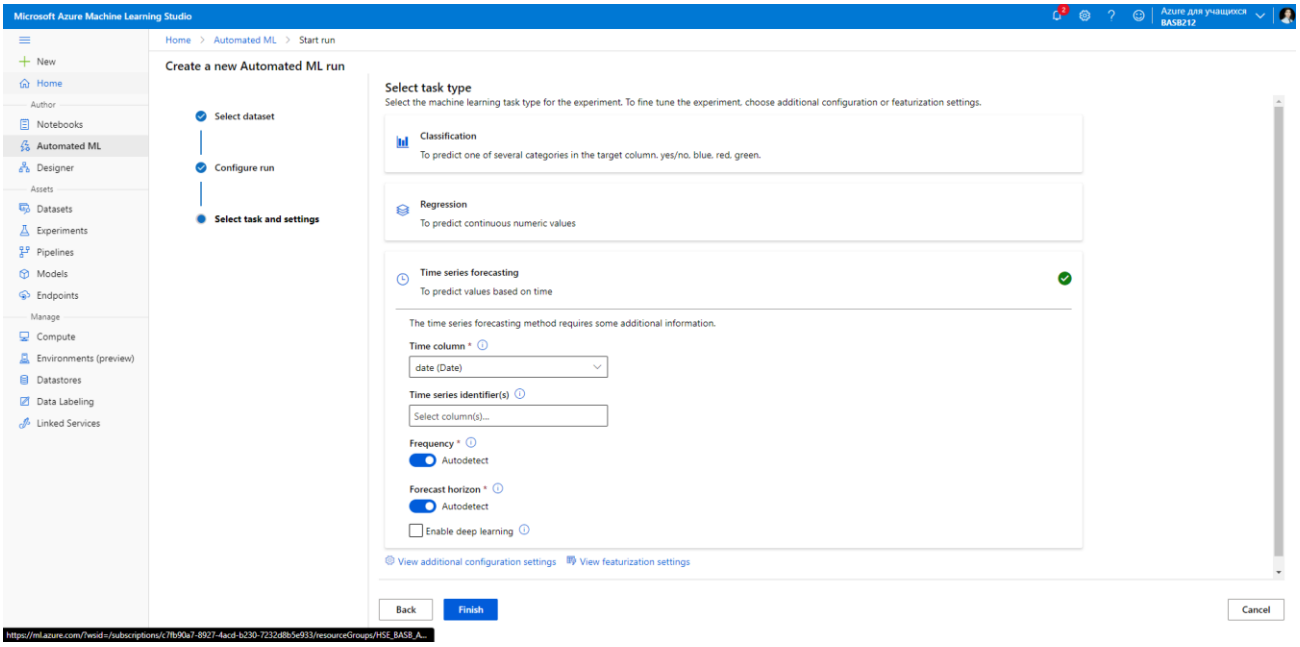


Compute a name “Standard-Vostok”. Then click “Next”

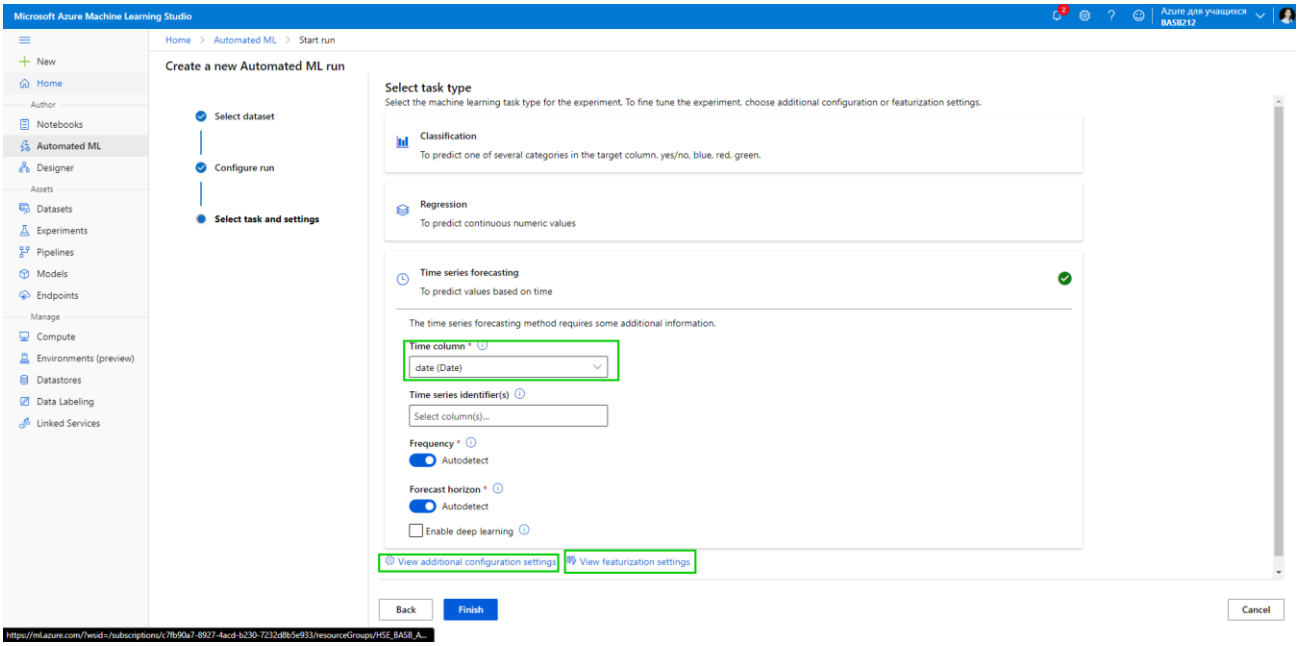


After then, we see this screen. Click just “Next”.

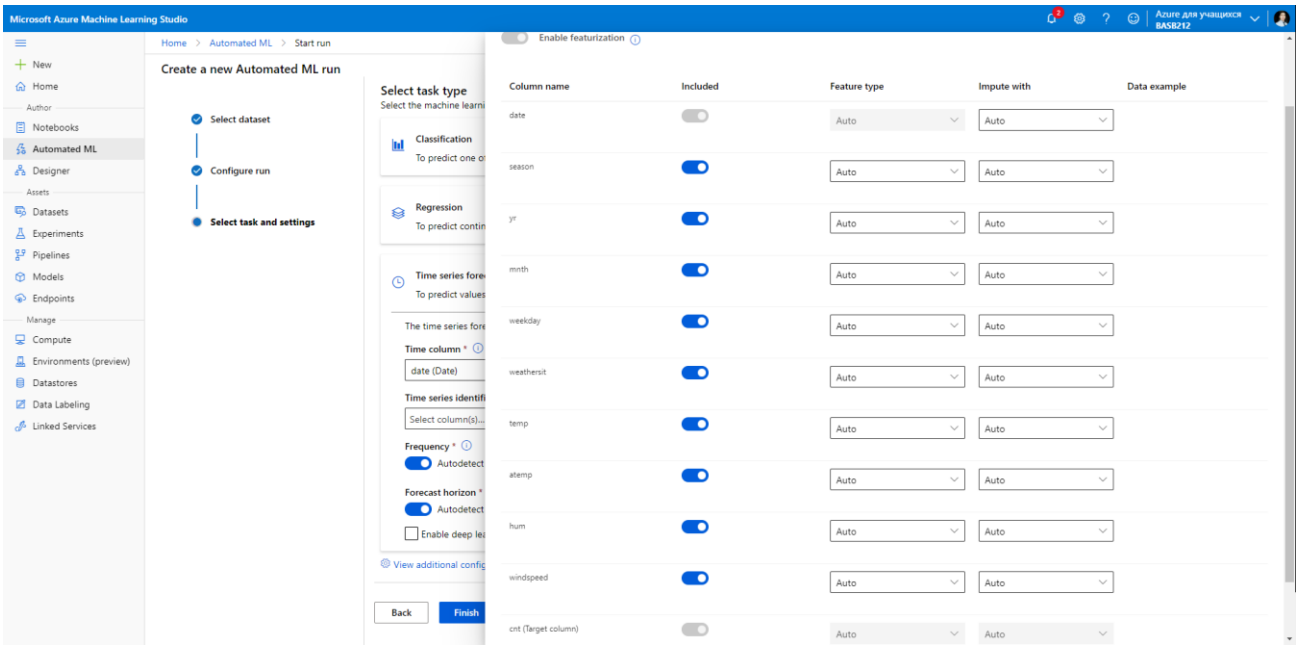




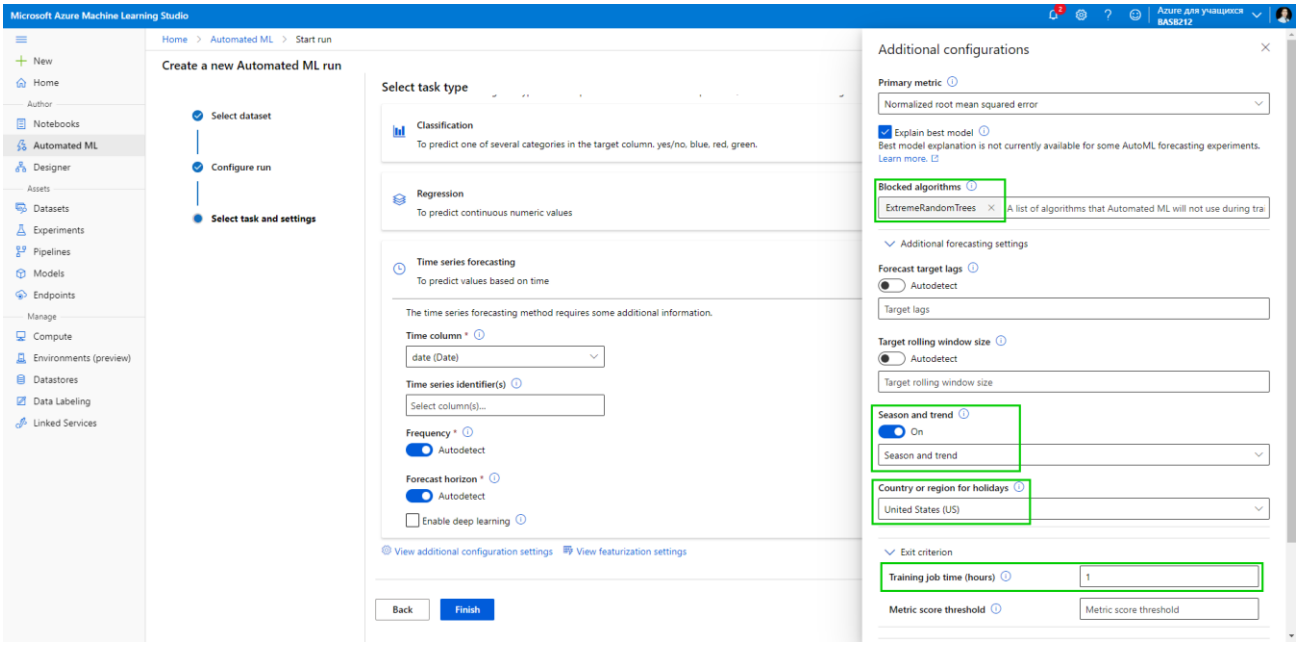
Step 16. Select the machine learning task type for the experiment. We choose “time series forecasting”. Time column “date (Date)”



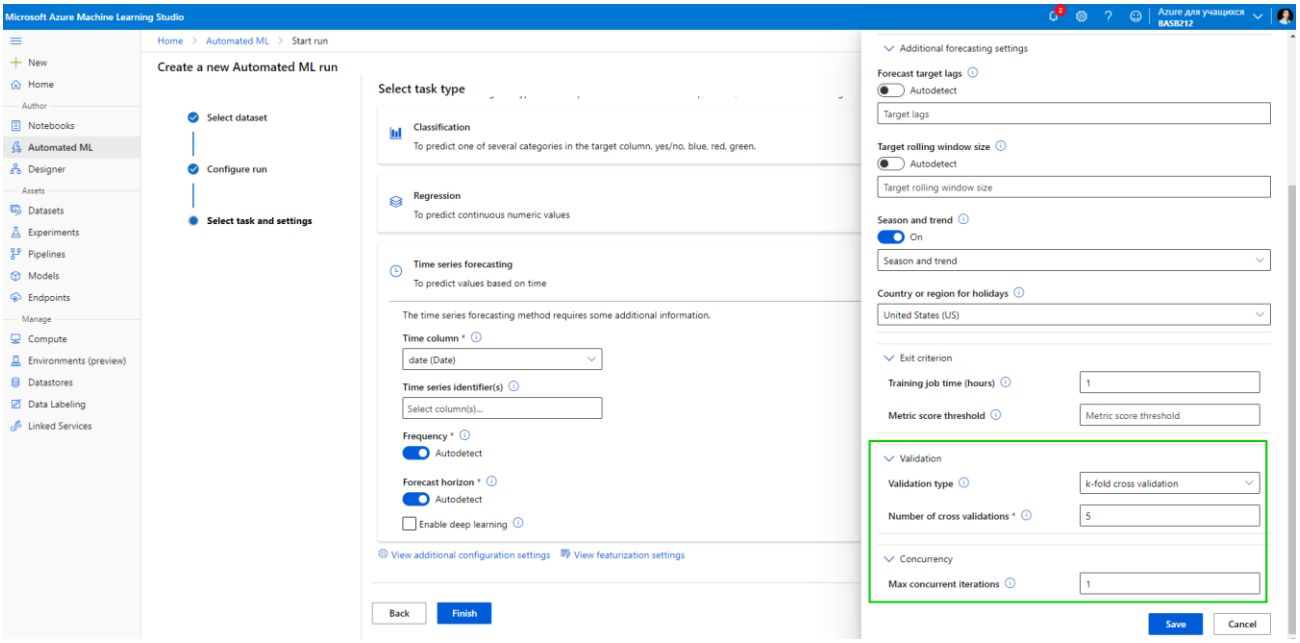
Then we go to “View featurization settings”. If needed we click. In this case we do not change anything. Just we need to persuade that there are not included “date” and “cnt (target column)”. Click “Save”



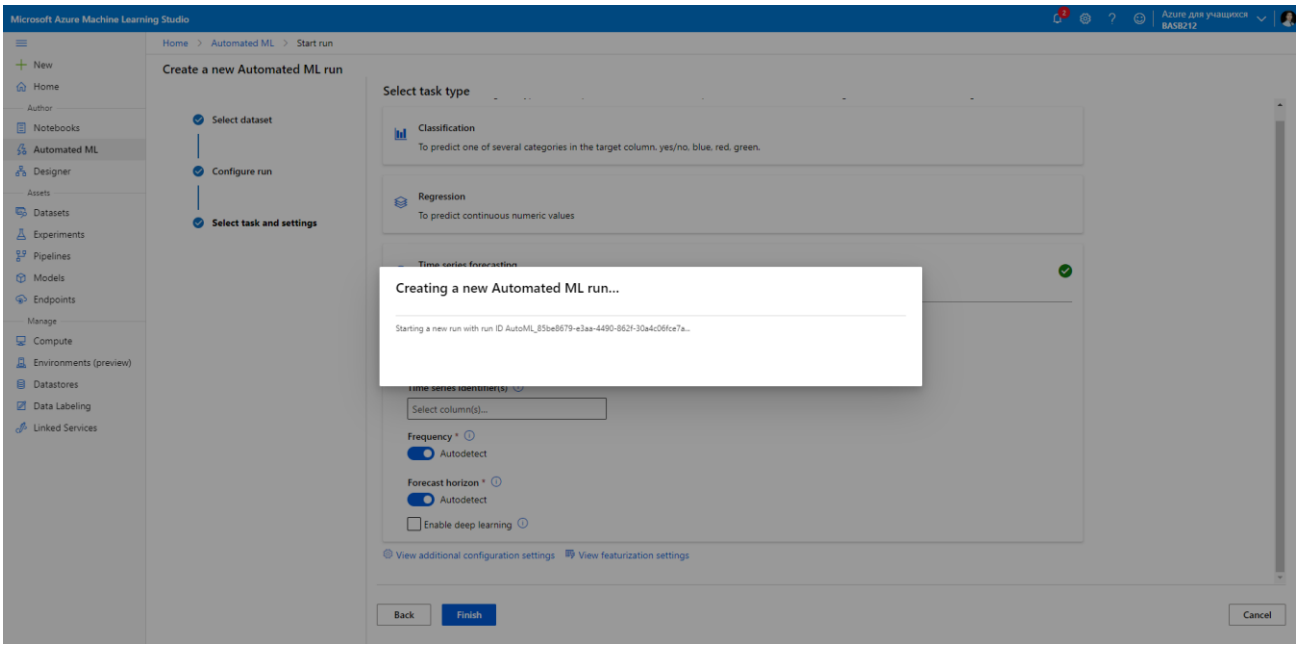
After then, we click on “View additional configuration settings”. There we choose primary metric “Normalized root mean squared error”. Block “ExtremeRandomTrees” algorithm. Turn on “Season and trend” and choose (Season and trend). Counrty or region for holidays “United States (US)”. In the Exit criterion we put 1 for training job time (hours).



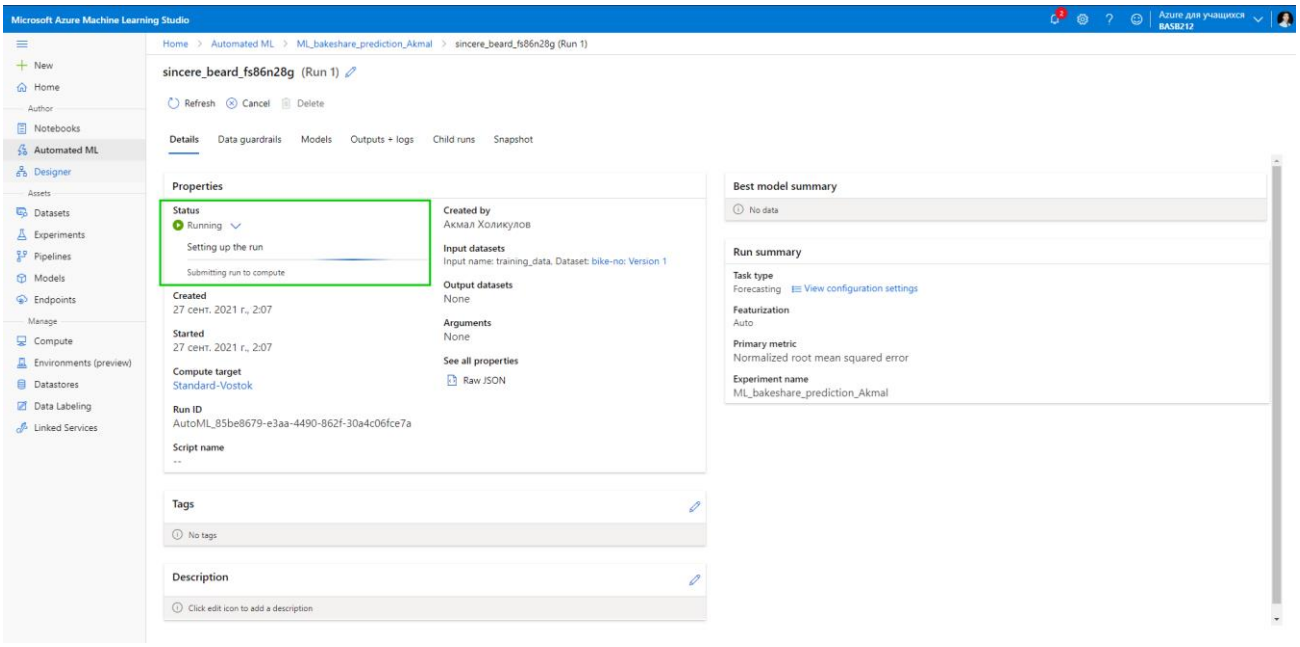
In the “Validation” and “Concurrency” we do not change anything. Click “Save” button



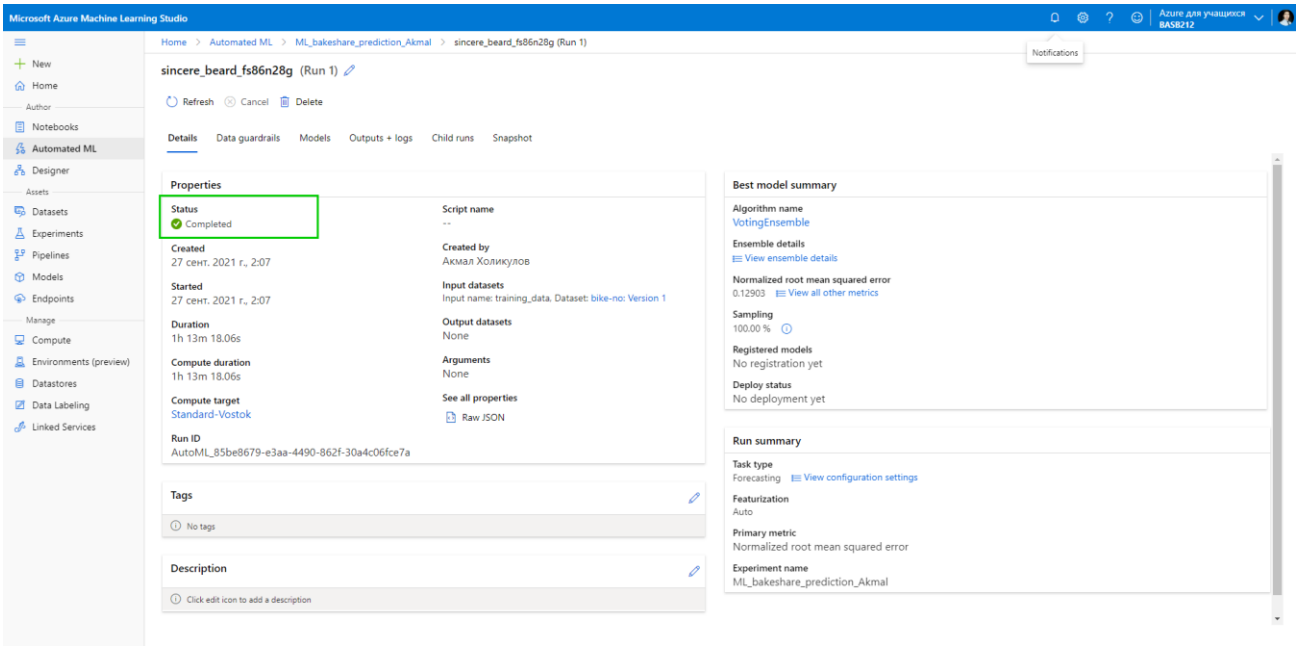
Click “Finish”. There will be creating a new Automated ML run



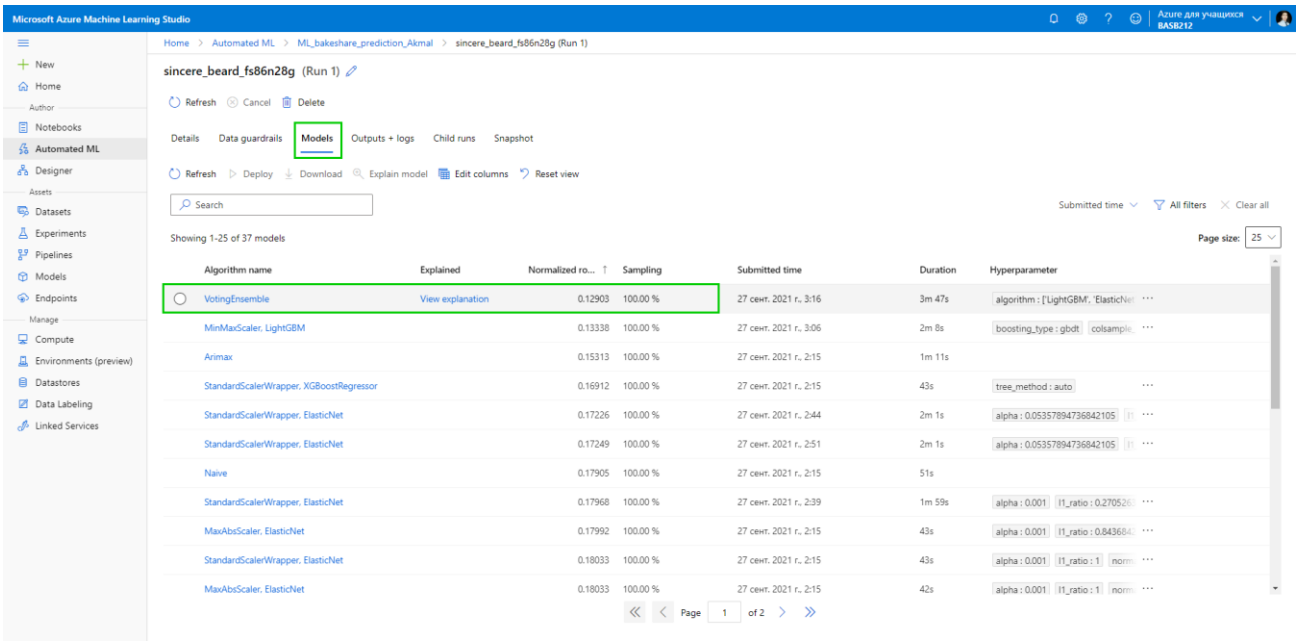
We need to wait while machine is running.



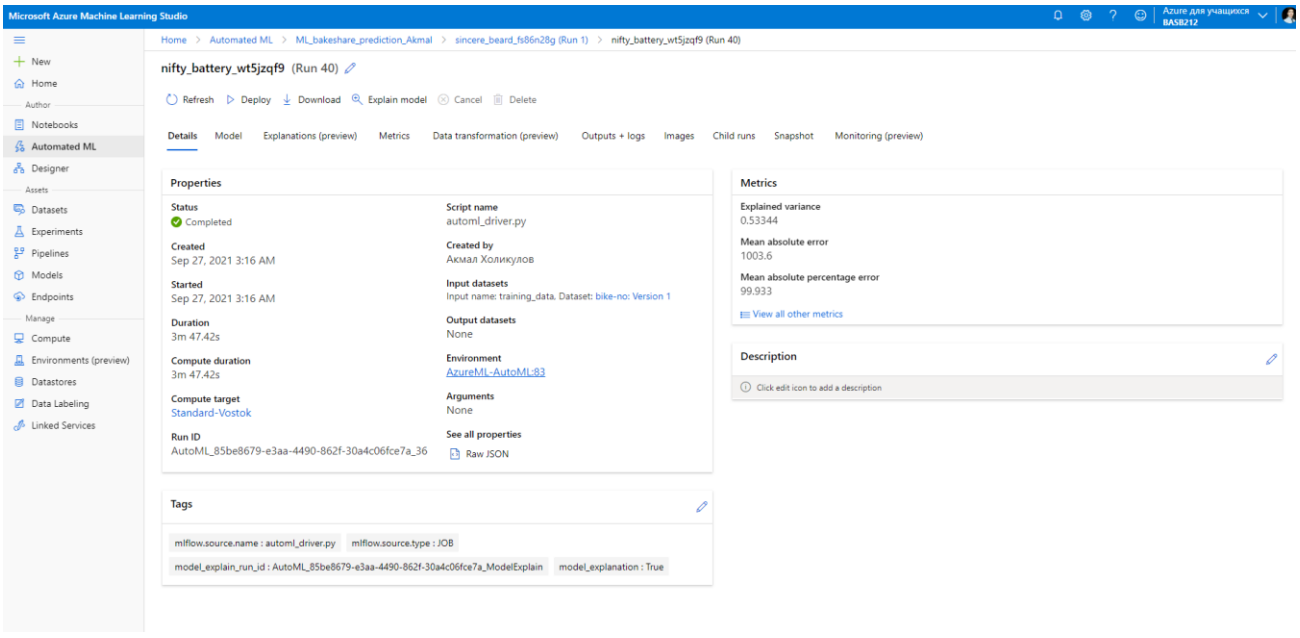
Status: Completed.

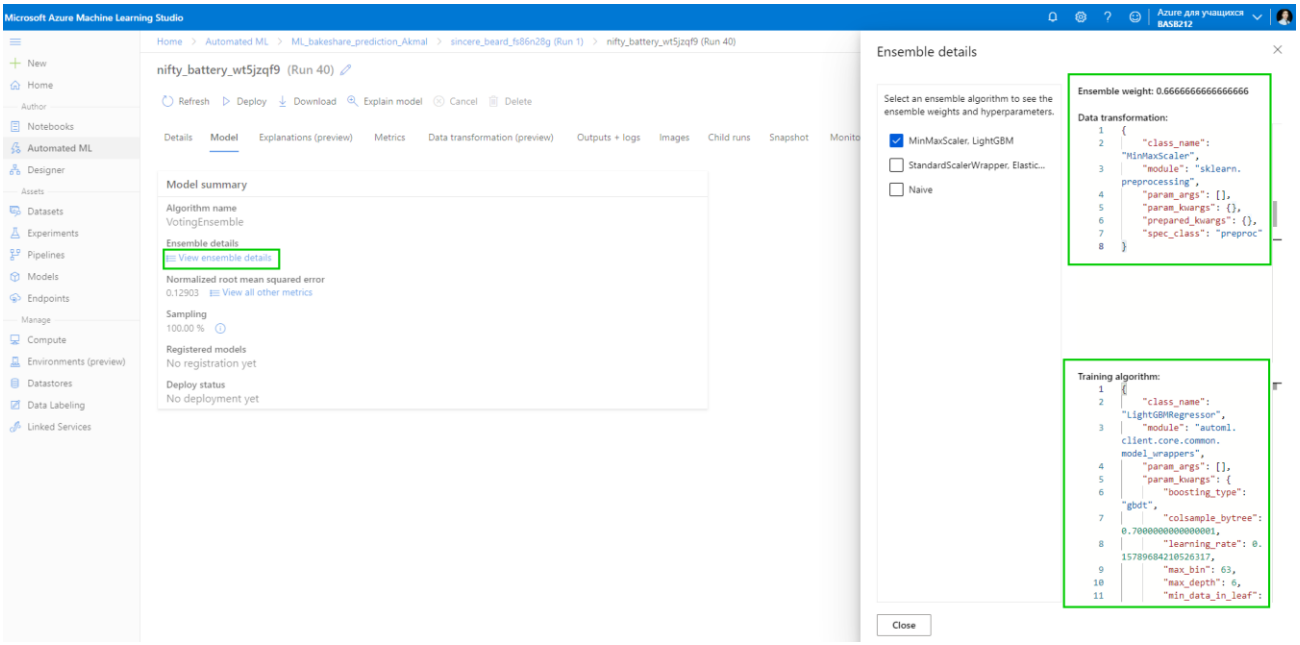


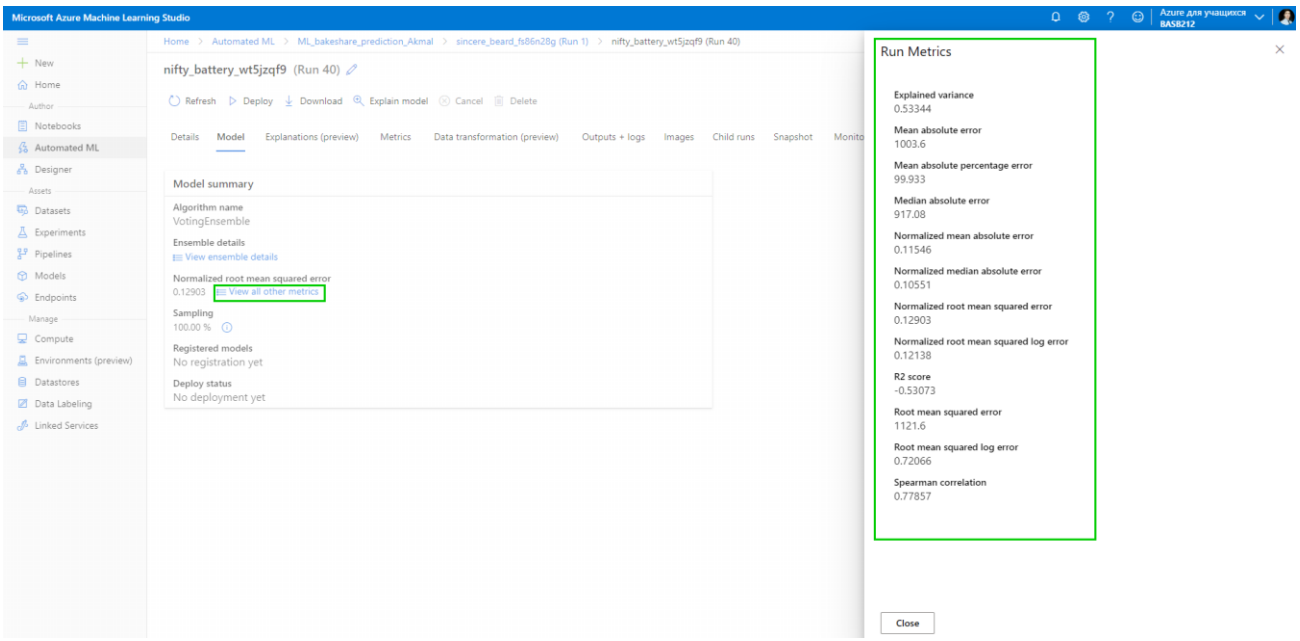
Step 17. We have to go to “Models” in order to see which is the best algorithm. VotingEnsemble is the best ML algorithm.

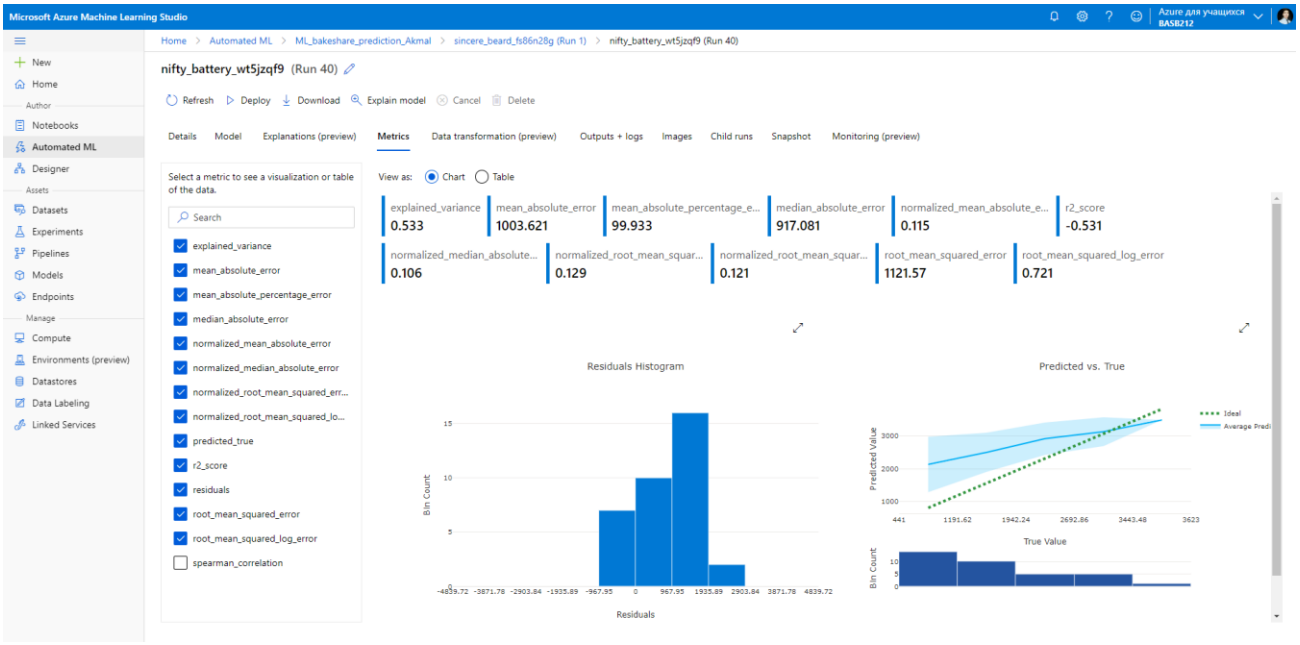


Lets to see more details about algorithm by clicking on “View explanation”

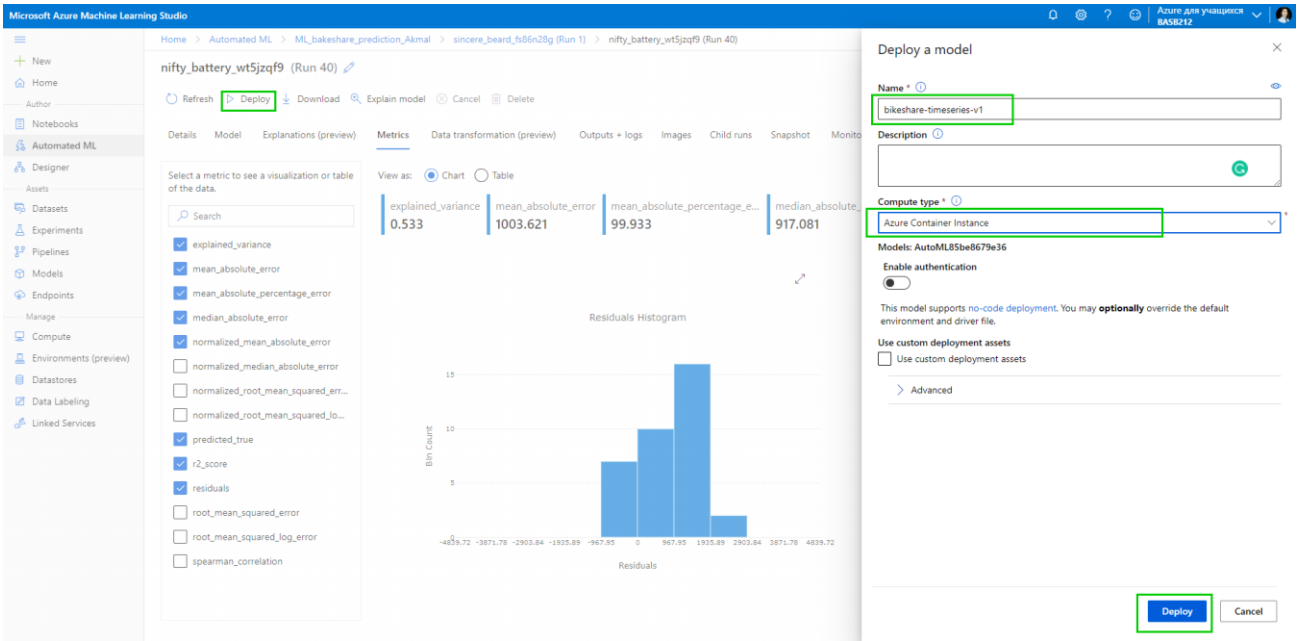



Microsoft Azure Machine Learning Studio
Azure для учащихся
BASB212
Home > Automated ML > ML_bakeshare_prediction_Akmal > sincere_beard_fs86n28g (Run 1) > nifty_battery_wt5jzqf9 (Run 40)
New
Home
Author
Notebooks
Automated ML
Designer
Assets
Datasets
Experiments
Pipelines
Models
Endpoints
Manage
Compute
Environments (preview)
Datastores
Data Labeling
Linked Services
nifty_battery_wt5jzqf9 (Run 40)
Refresh
Deploy
Download
Explain model
Cancel
Delete
Details
Model
Explanations (preview)
Metrics
Data transformation (preview)
Outputs + logs
Images
Child runs
Snapshot
Model summary
Algorithm name
VotingEnsemble
Ensemble details
View ensemble details
Normalized root mean squared error
0.12903
View all other metrics
Sampling
100.00 %
Registered models
No registration yet
Deploy status
No deployment yet
Ensemble details
Select an ensemble algorithm to see the ensemble weights and hyperparameters.
MinMaxScaler, LightGBM
StandardScalerWrapper, Elastic...
Naive
Ensemble weight: 0.6666666666666666
Data transformation:
{
"class_name": "MinMaxScaler",
"module": "sklearn.preprocessing",
"param_args": [],
"param_kwargs": {},
"prepared_kwargs": {},
"spec_class": "preproc"
}
Training algorithm:
{
"class_name": "LightGBMRegressor",
"module": "automl.client.core.common.model_wrappers",
"param_args": [],
"param_kwargs": {
"boosting_type": "gbdt",
"colsample_bytree": 0.7000000000000001,
"learning_rate": 0.1578968421052631 7,
"max_bin": 63,
"max_depth": 6,
"min_data_in_leaf":
Close

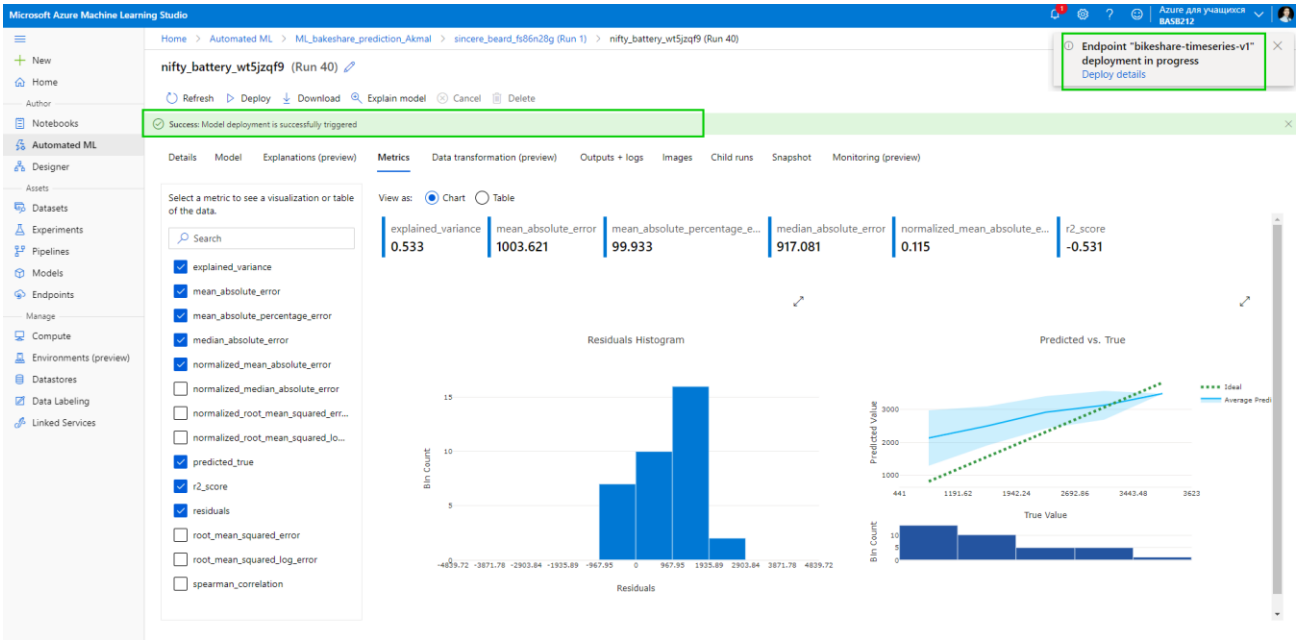

Microsoft Azure Machine Learning Studio
Azure для учащихся
BASB212
Home > Automated ML > ML_bakeshare_prediction_Akmal > sincere_beard_fs86n28g (Run 1) > nifty_battery_wt5jzqf9 (Run 40)
New
Home
Author
Notebooks
Automated ML
Designer
Assets
Datasets
Experiments
Pipelines
Models
Endpoints
Manage
Compute
Environments (preview)
Datastores
Data Labeling
Linked Services
nifty_battery_wt5jzqf9 (Run 40)
Refresh
Deploy
Download
Explain model
Cancel
Delete
Details
Model
Explanations (preview)
Metrics
Data transformation (preview)
Outputs + logs
Images
Child runs
Snapshot
Model summary
Algorithm name
VotingEnsemble
Ensemble details
View ensemble details
Normalized root mean squared error
0.12903
View all other metrics
Sampling
100.00 %
Registered models
No registration yet
Deploy status
No deployment yet
Run Metrics
Explained variance
0.53344
Mean absolute error
1003.6
Mean absolute percentage error
99.933
Median absolute error
917.08
Normalized mean absolute error
0.11546
Normalized median absolute error
0.10551
Normalized root mean squared error
0.12903
Normalized root mean squared log error
0.12138
R2 score
-0.53073
Root mean squared error
1121.6
Root mean squared log error
0.72066
Spearman correlation
0.77857
Close



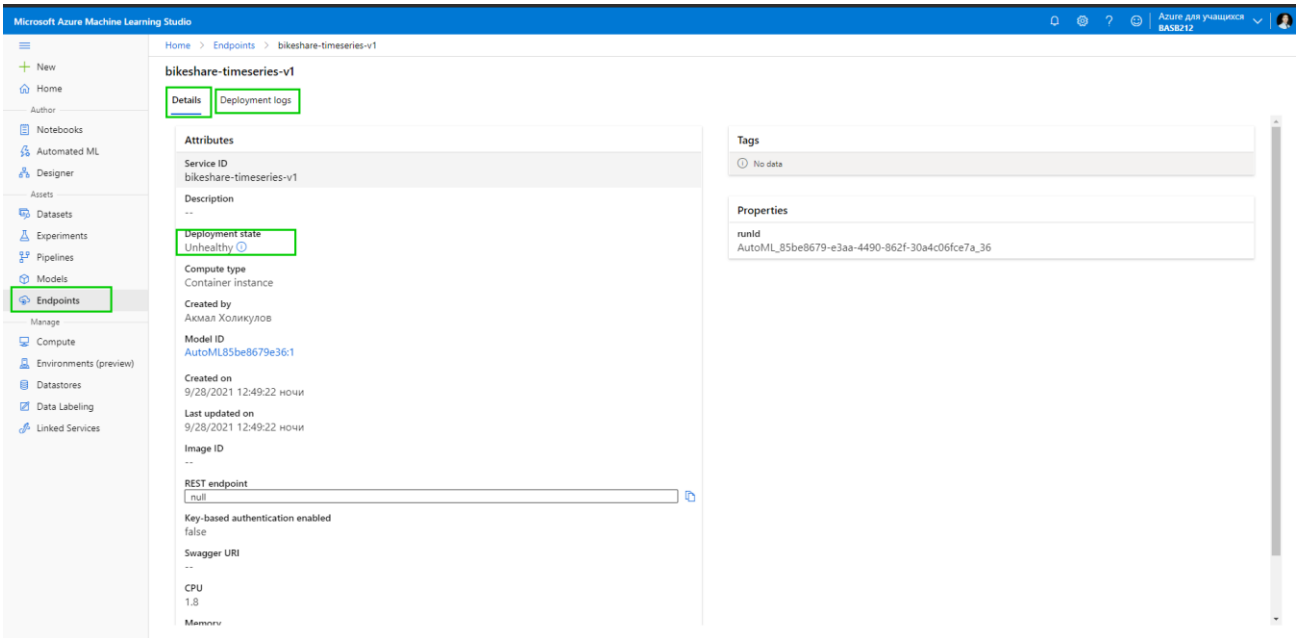
Choose the needed metrics on left side. Click on the “Deploy”. Then on the right side appears “Deploy a model” window. Put a name “bikeshare-timeseries-v1”. Then choose Compute type “Azure Container Instance”. After that click Deploy button.

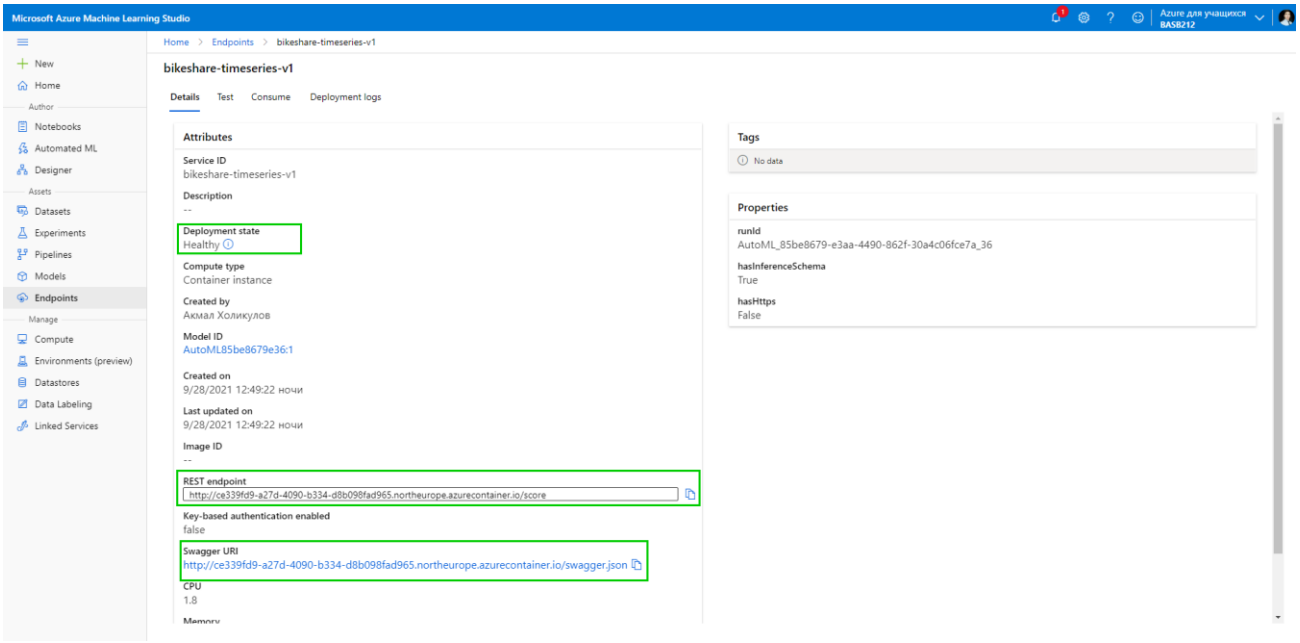


We see that “Model deployment is successfully triggered”. We need to click on Deploy details

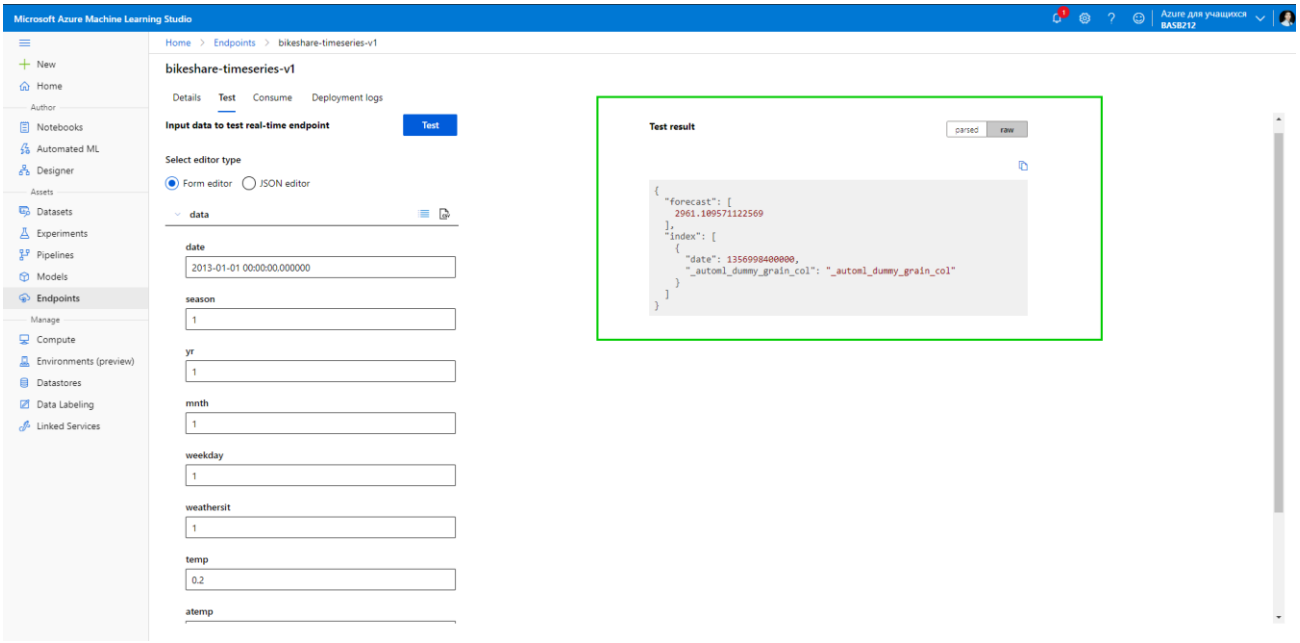


Appears deployment details on the screen





Let’s test it



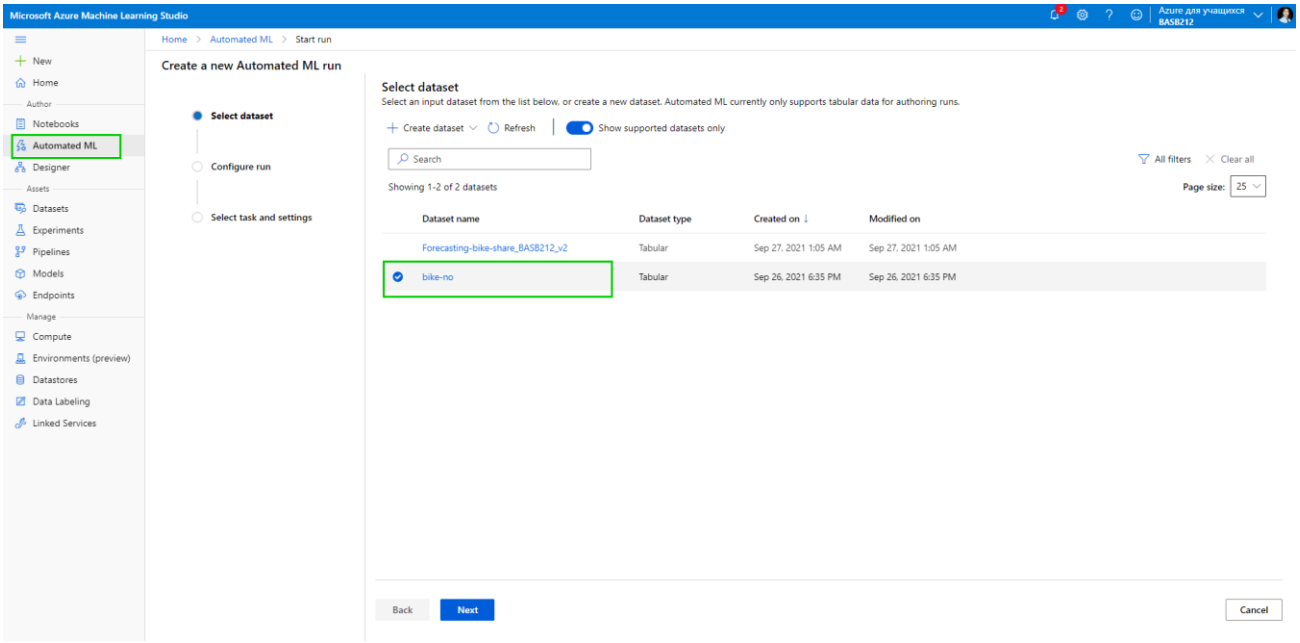
Also in the “Consume” there are the codes in 3 programming languages. Lets choose Python.

```
import urllib.request
import json
import os
import ssl

def allowSelfSignedHttps(allowed):
    # bypass the server certificate verification on client side
```

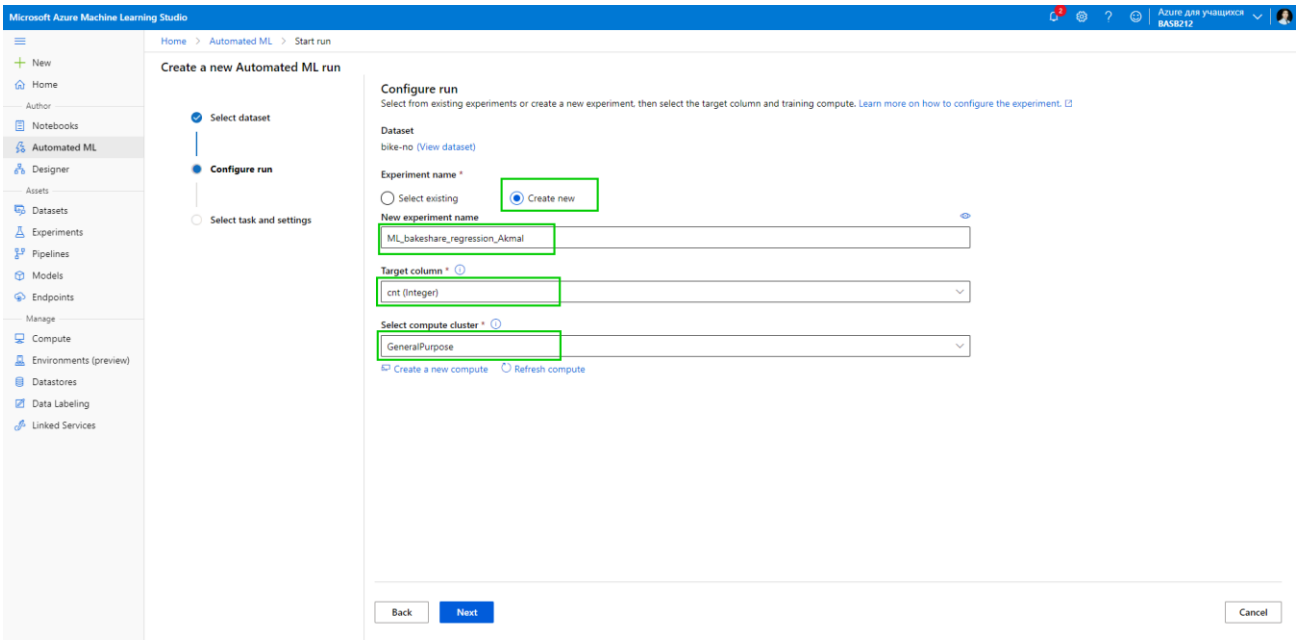
```
    if allowed and not os.environ.get('PYTHONHTTPSVERIFY', '') and getattr(ssl,
'_create_unverified_context', None):
        ssl._create_default_https_context = ssl._create_unverified_context

allowSelfSignedHttps(True) # this line is needed if you use self-signed certificate in your scoring
service.

# Request data goes here
data = {
    "data":
    [
        {
            'date': "2000-01-01 00:00:00,000000",
            'season': "0",
            'yr': "0",
            'mnth': "0",
            'weekday': "0",
            'weathersit': "0",
            'temp': "0",
            'atemp': "0",
            'hum': "0",
            'windspeed': "0",
        },
    ],
}

body = str.encode(json.dumps(data))

url = 'http://ce339fd9-a27d-4090-b334-d8b098fad965.northeurope.azurecontainer.io/score'
api_key = '' # Replace this with the API key for the web service
headers = {'Content-Type':'application/json', 'Authorization':('Bearer '+ api_key)}

req = urllib.request.Request(url, body, headers)

try:
    response = urllib.request.urlopen(req)

    result = response.read()
    print(result)
except urllib.error.HTTPError as error:
    print("The request failed with status code: " + str(error.code))

    # Print the headers - they include the requert ID and the timestamp, which are useful for
debugging the failure
    print(error.info())
                                print(json.loads(error.read().decode("utf8", 'ignore')))
```

Regression (self-work)

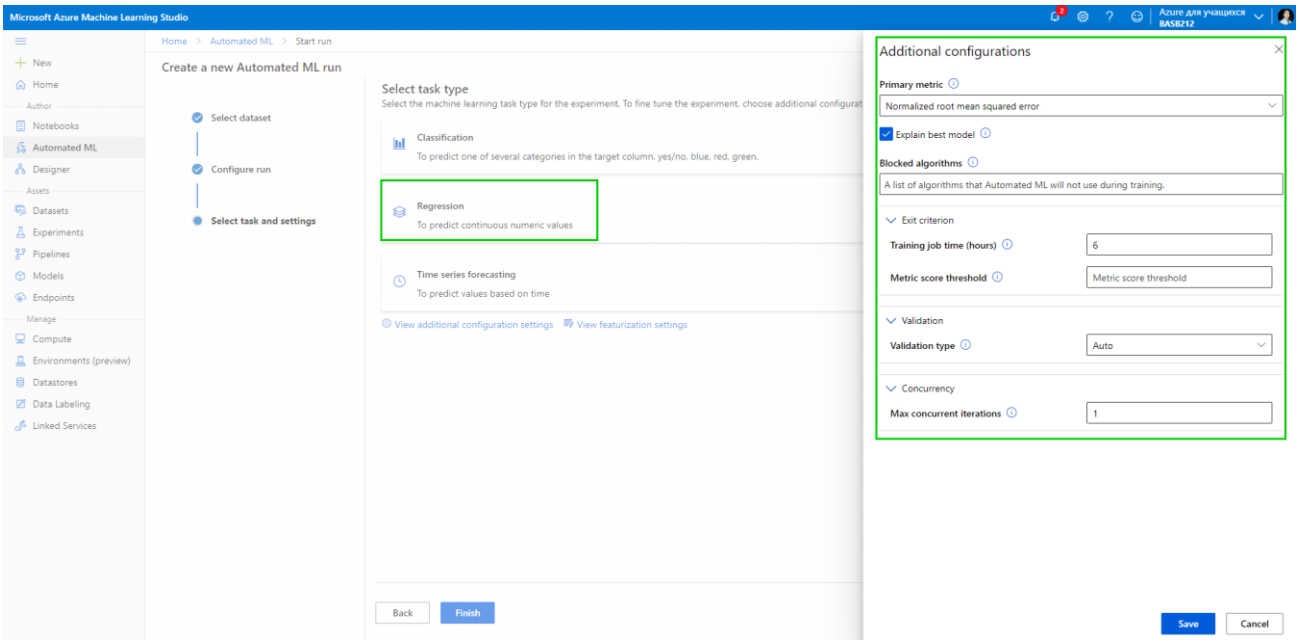
Step 1.

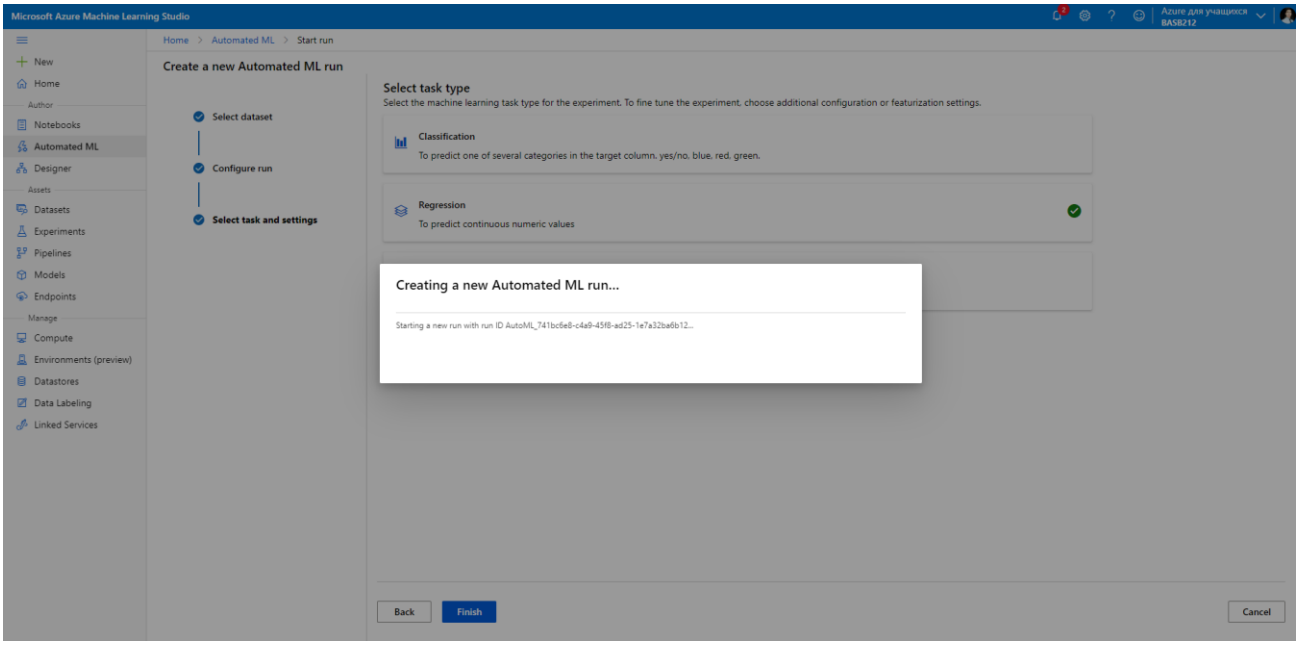


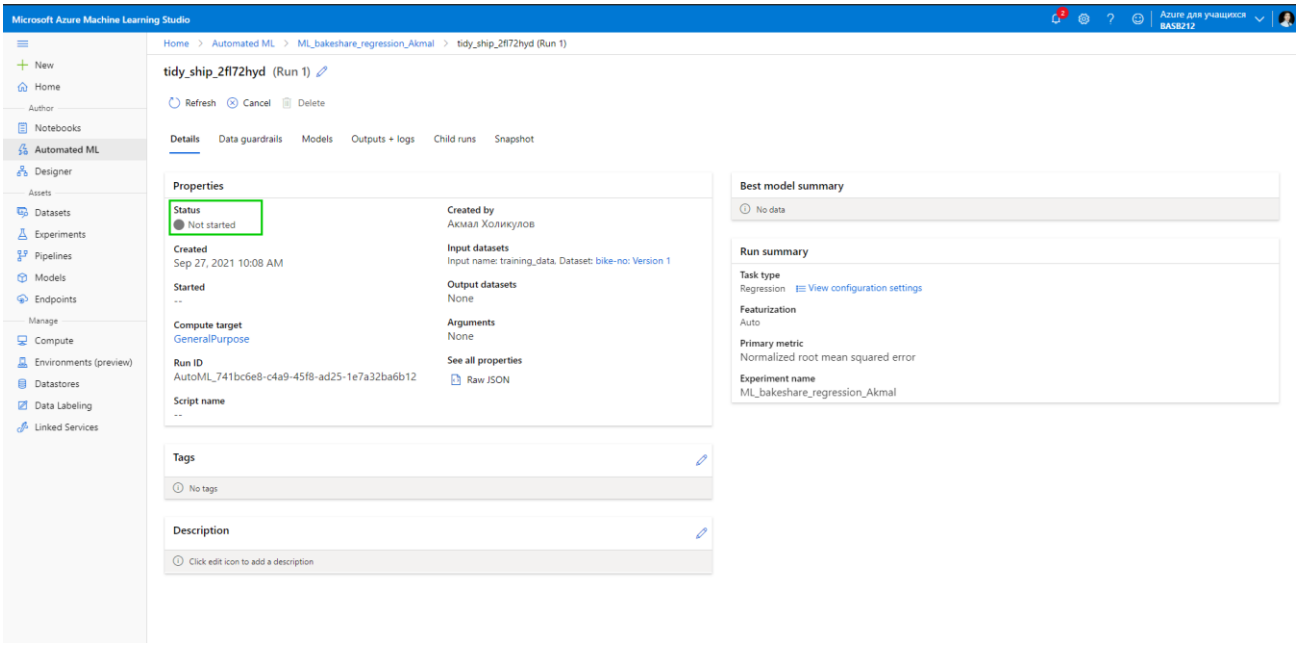
Step 2

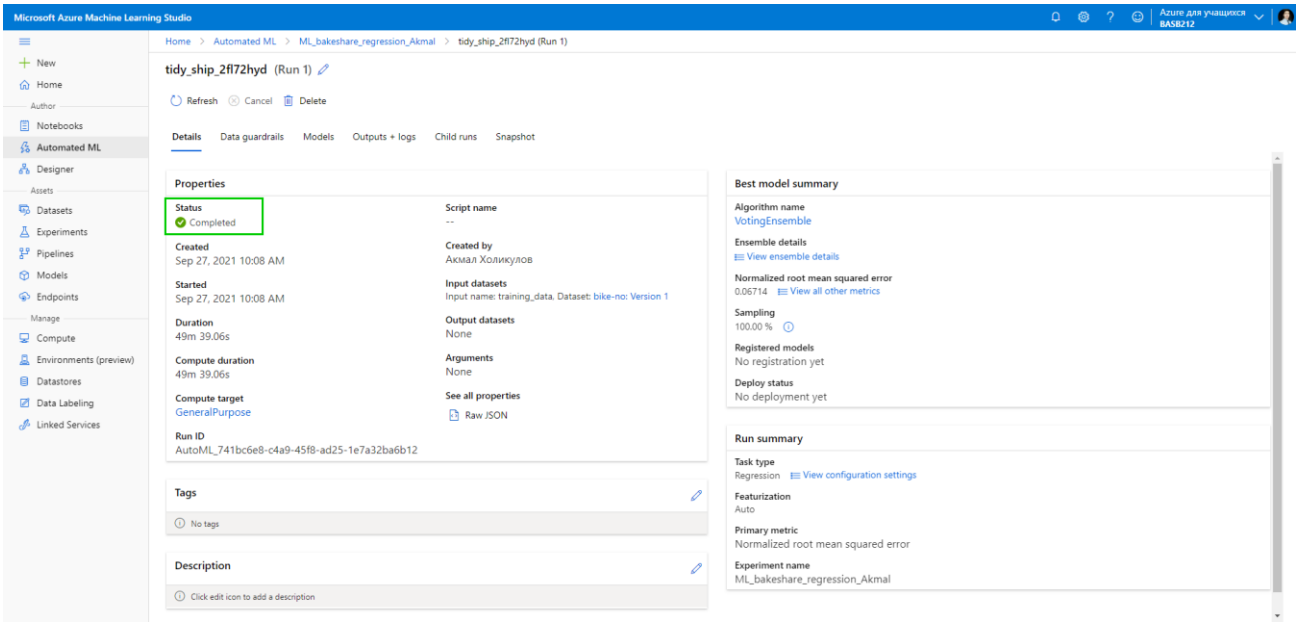


Step 3

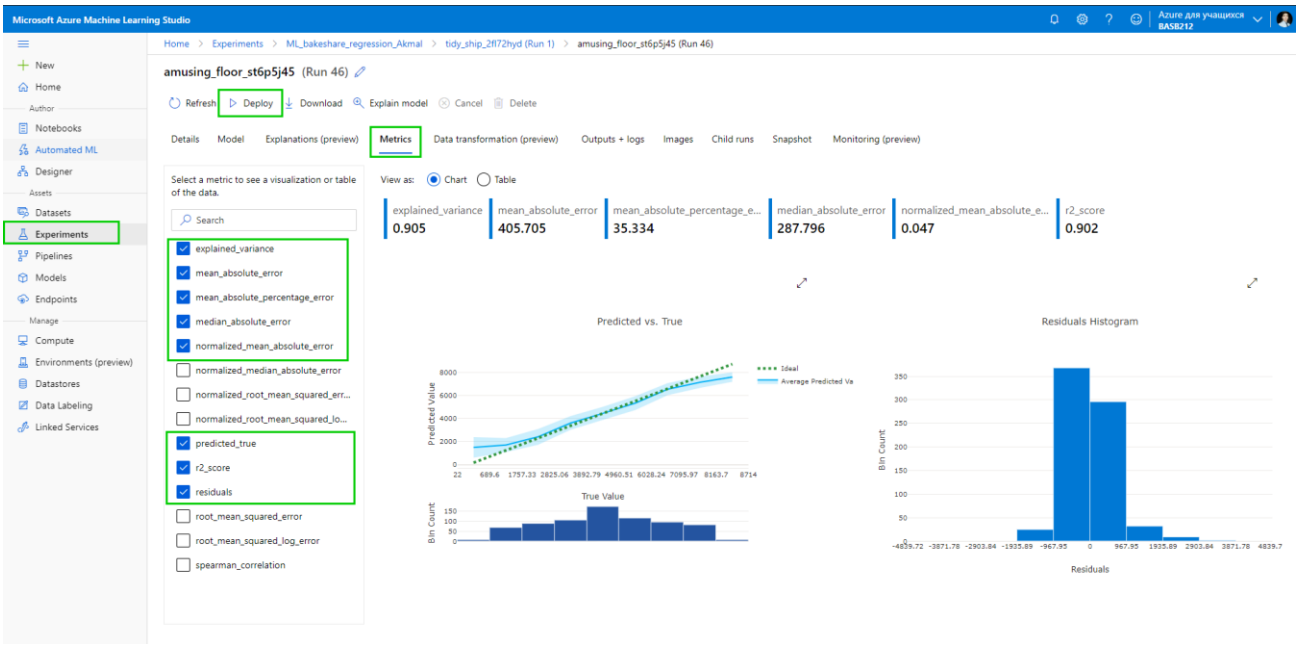

Microsoft Azure Machine Learning Studio
Azure для учащихся
BASB212
Home > Automated ML > Start run
Create a new Automated ML run
New
Home
Author
Notebooks
Automated ML
Designer
Assets
Datasets
Experiments
Pipelines
Models
Endpoints
Manage
Compute
Environments (preview)
Datastores
Data Labeling
Linked Services
Select dataset
Configure run
Select task and settings
Select task type
Select the machine learning task type for the experiment. To fine tune the experiment, choose additional configurat
Classification
To predict one of several categories in the target column. yes/no. blue. red. green.
Regression
To predict continuous numeric values
Time series forecasting
To predict values based on time
View additional configuration settings
View featurization settings
Back
Finish
Additional configurations
Primary metric
Normalized root mean squared error
Explain best model
Blocked algorithms
A list of algorithms that Automated ML will not use during training.
Exit criterion
Training job time (hours)
6
Metric score threshold
Metric score threshold
Validation
Validation type
Auto
Concurrency
Max concurrent iterations
1
Save
Cancel

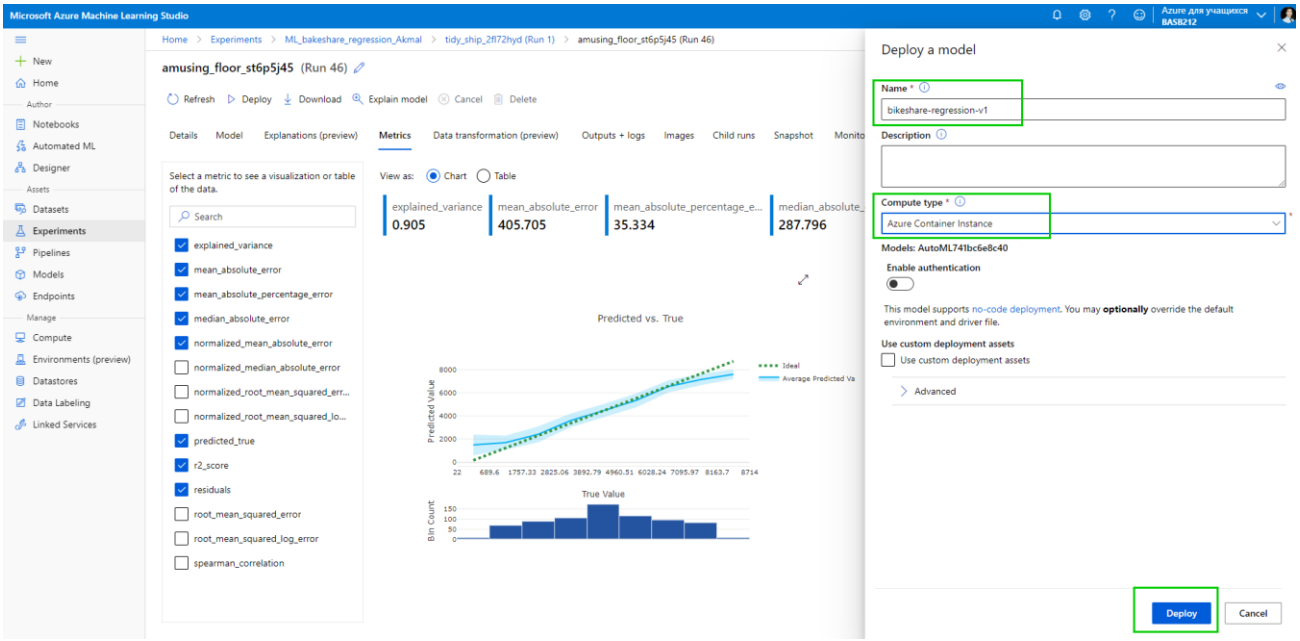

Microsoft Azure Machine Learning Studio
Azure для учащихся
BASB212
Home > Automated ML > Start run
Create a new Automated ML run
New
Home
Author
Notebooks
Automated ML
Designer
Assets
Datasets
Experiments
Pipelines
Models
Endpoints
Manage
Compute
Environments (preview)
Datastores
Data Labeling
Linked Services
Select dataset
Configure run
Select task and settings
Select task type
Select the machine learning task type for the experiment. To fine tune the experiment, choose additional configuration or featurization settings.
Classification
To predict one of several categories in the target column. yes/no. blue. red. green.
Regression
To predict continuous numeric values
Creating a new Automated ML run...
Starting a new run with run ID AutoML_741bc6e8-c4a9-45f8-ad25-1e7a32ba6b12...
Back
Finish
Cancel

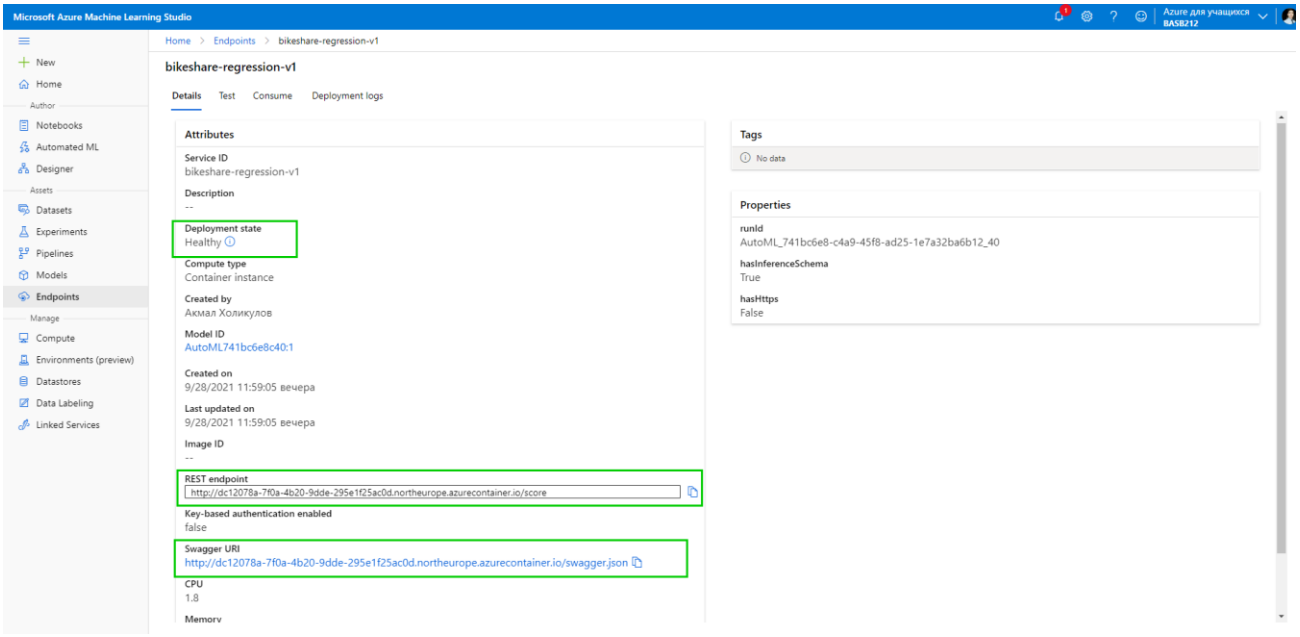

Microsoft Azure Machine Learning Studio
Azure для учащихся
BASB212
Home > Automated ML > ML_bakeshare_regression_Akmal > tidy_ship_2fl72hyd (Run 1)
New
Home
Author
Notebooks
Automated ML
Designer
Assets
Datasets
Experiments
Pipelines
Models
Endpoints
Manage
Compute
Environments (preview)
Datastores
Data Labeling
Linked Services
tidy_ship_2fl72hyd (Run 1)
Refresh
Cancel
Delete
Details
Data guardrails
Models
Outputs + logs
Child runs
Snapshot
Properties
Status
Not started
Created
Sep 27, 2021 10:08 AM
Started
--
Compute target
GeneralPurpose
Run ID
AutoML_741bc6e8-c4a9-45f8-ad25-1e7a32ba6b12
Script name
--
Created by
Акмал Холикулов
Input datasets
Input name: training_data, Dataset: bike-no: Version 1
Output datasets
None
Arguments
None
See all properties
Raw JSON
Best model summary
No data
Run summary
Task type
Regression
View configuration settings
Featurization
Auto
Primary metric
Normalized root mean squared error
Experiment name
ML_bakeshare_regression_Akmal
Tags
No tags
Description
Click edit icon to add a description

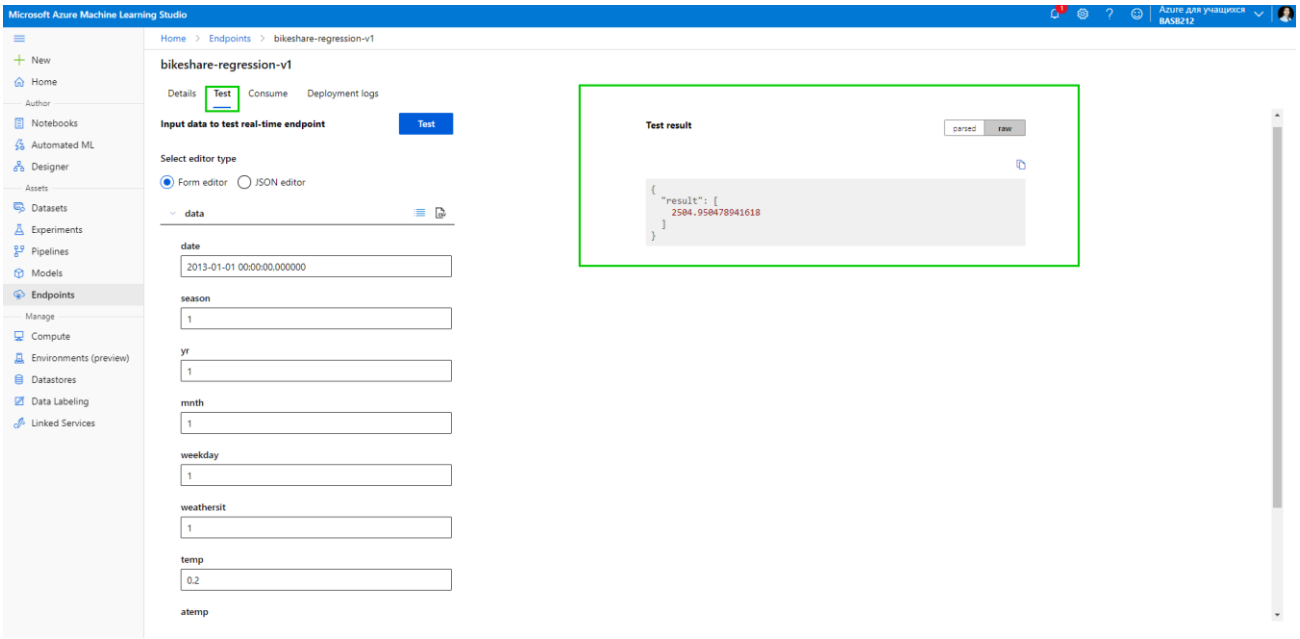

Microsoft Azure Machine Learning Studio
Azure для учащихся
BASB212
Home > Automated ML > ML_bakeshare_regression_Akmal > tidy_ship_2fl72hyd (Run 1)
New
Home
Author
Notebooks
Automated ML
Designer
Assets
Datasets
Experiments
Pipelines
Models
Endpoints
Manage
Compute
Environments (preview)
Datastores
Data Labeling
Linked Services
tidy_ship_2fl72hyd (Run 1)
Refresh
Cancel
Delete
Details
Data guardrails
Models
Outputs + logs
Child runs
Snapshot
Properties
Status
Completed
Created
Sep 27, 2021 10:08 AM
Started
Sep 27, 2021 10:08 AM
Duration
49m 39.06s
Compute duration
49m 39.06s
Compute target
GeneralPurpose
Run ID
AutoML_741bc6e8-c4a9-45f8-ad25-1e7a32ba6b12
Script name
--
Created by
Акмал Холикулов
Input datasets
Input name: training_data, Dataset: bike-no: Version 1
Output datasets
None
Arguments
None
See all properties
Raw JSON
Best model summary
Algorithm name
VotingEnsemble
Ensemble details
View ensemble details
Normalized root mean squared error
0.06714
View all other metrics
Sampling
100.00 %
Registered models
No registration yet
Deploy status
No deployment yet
Run summary
Task type
Regression
View configuration settings
Featurization
Auto
Primary metric
Normalized root mean squared error
Experiment name
ML_bakeshare_regression_Akmal
Tags
No tags
Description
Click edit icon to add a description

Now we need to deploy model. We have to repeat as before.







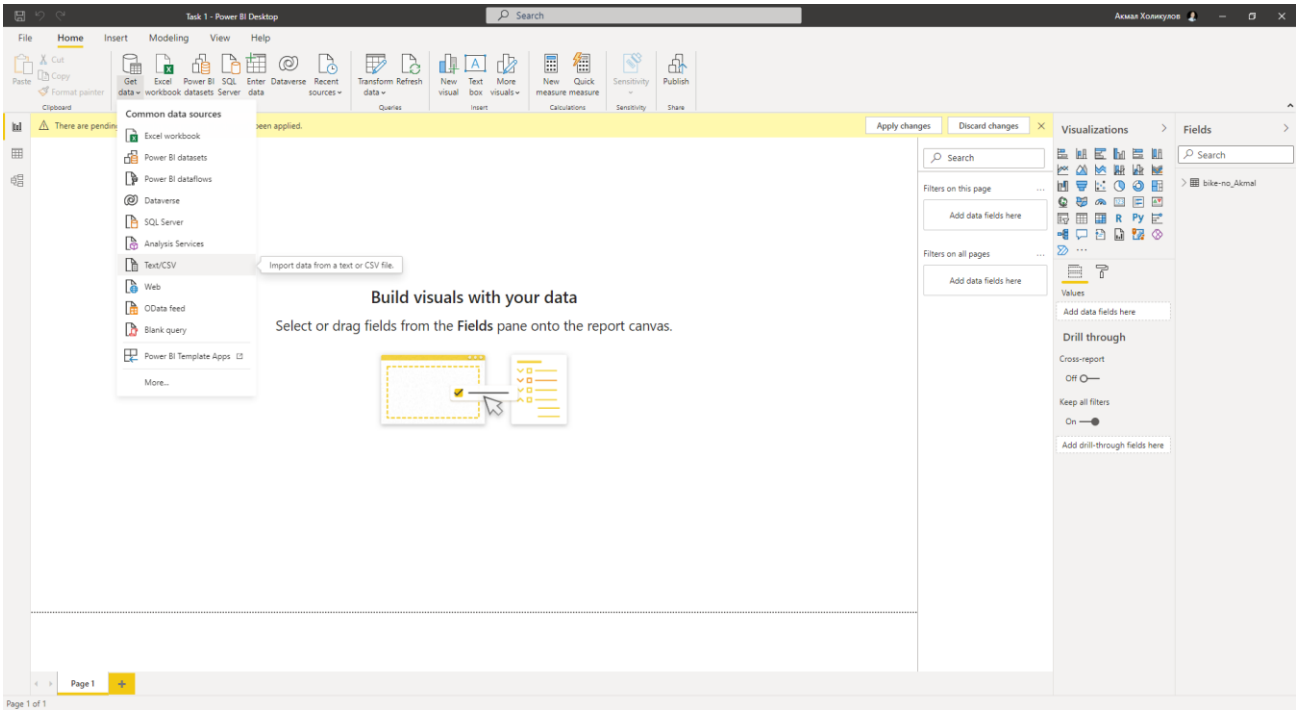


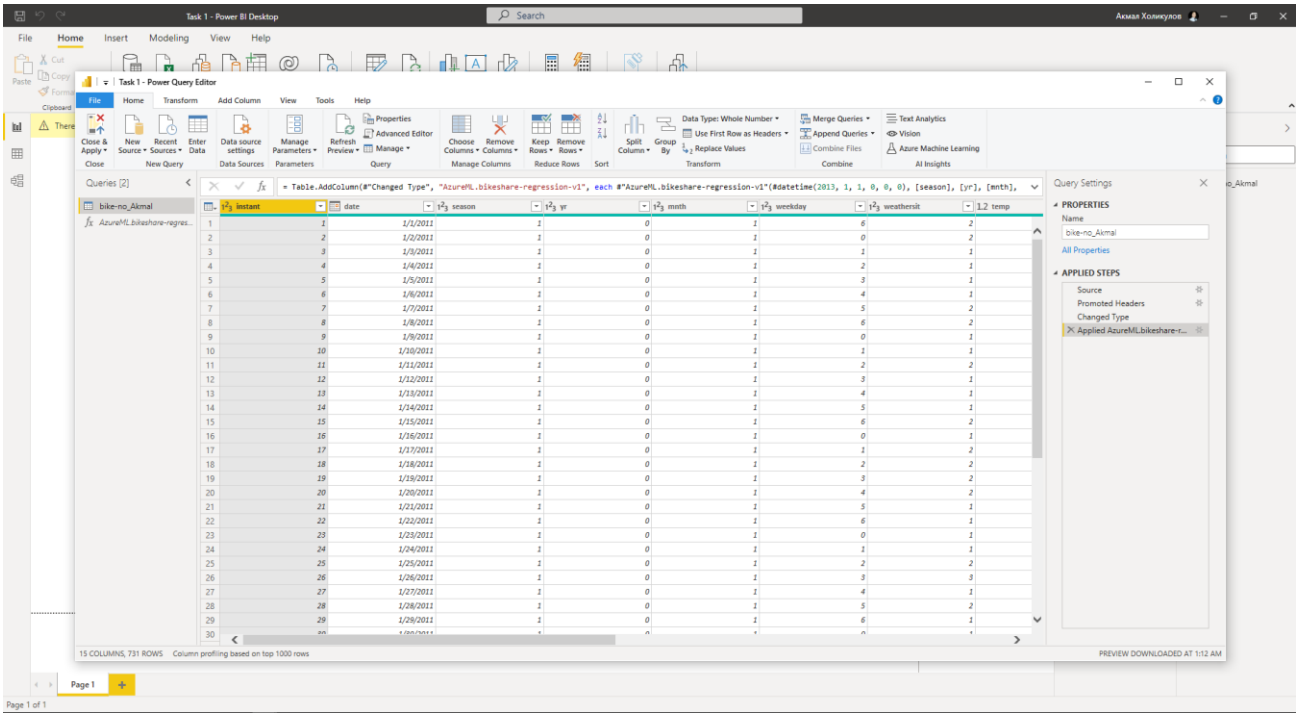

```
import urllib.request
import json
import os
import ssl

def allowSelfSignedHttps(allowed):
    # bypass the server certificate verification on client side
```

```
    if allowed and not os.environ.get('PYTHONHTTPSVERIFY', '') and getattr(ssl, '_c
reate_unverified_context', None):
        ssl._create_default_https_context = ssl._create_unverified_context

allowSelfSignedHttps(True) # this line is needed if you use self-
signed certificate in your scoring service.

# Request data goes here
data = {
    "data":
    [
        {
            'date': "2000-01-01 00:00:00,000000",
            'season': "0",
            'yr': "0",
            'mnth': "0",
            'weekday': "0",
            'weathersit': "0",
            'temp': "0",
            'atemp': "0",
            'hum': "0",
            'windspeed': "0",
        },
    ],
}

body = str.encode(json.dumps(data))

url = 'http://dc12078a-7f0a-4b20-9dde-
295e1f25ac0d.northeurope.azurecontainer.io/score'
api_key = '' # Replace this with the API key for the web service
headers = {'Content-Type':'application/json', 'Authorization':('Bearer '+ api_key)}

req = urllib.request.Request(url, body, headers)

try:
    response = urllib.request.urlopen(req)

    result = response.read()
    print(result)
except urllib.error.HTTPError as error:
    print("The request failed with status code: " + str(error.code))

    # Print the headers - they include the requert ID and the timestamp, which are
useful for debugging the failure
    print(error.info())
    print(json.loads(error.read().decode("utf8", 'ignore')))
```

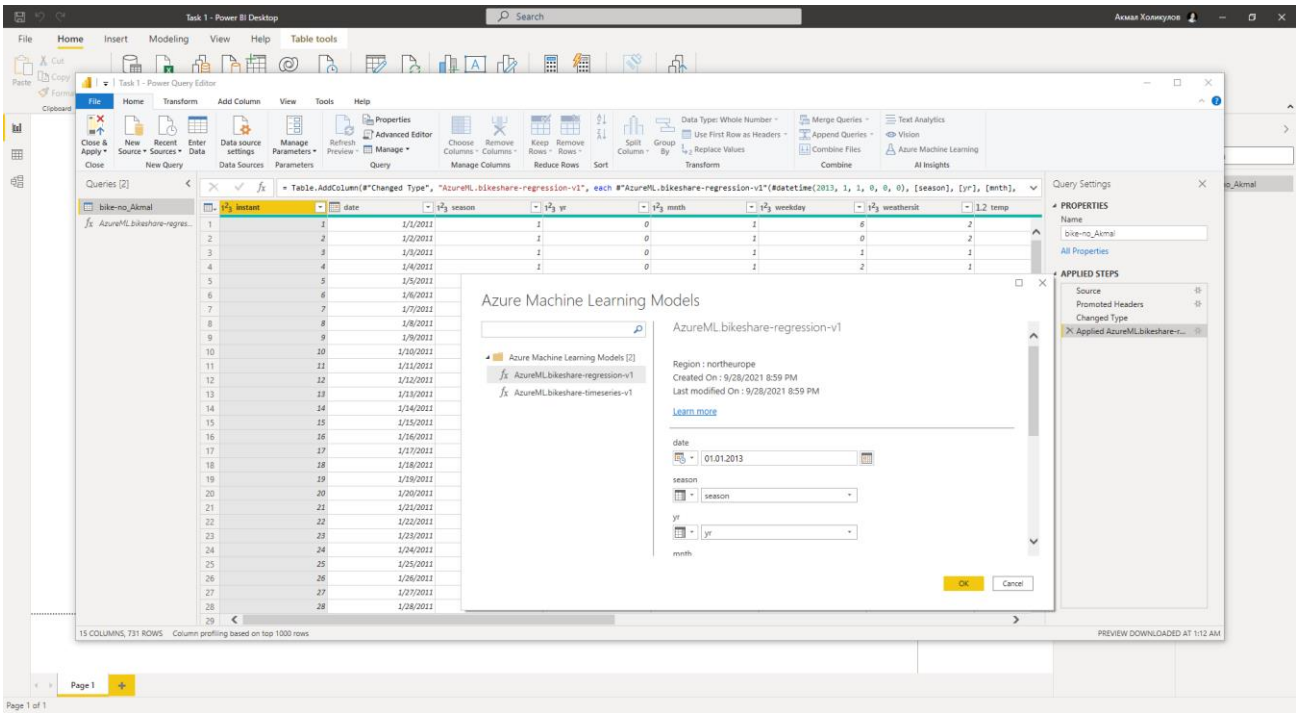
Consume Azure Machine Learning models in Power BI

First, we need to upload a data.

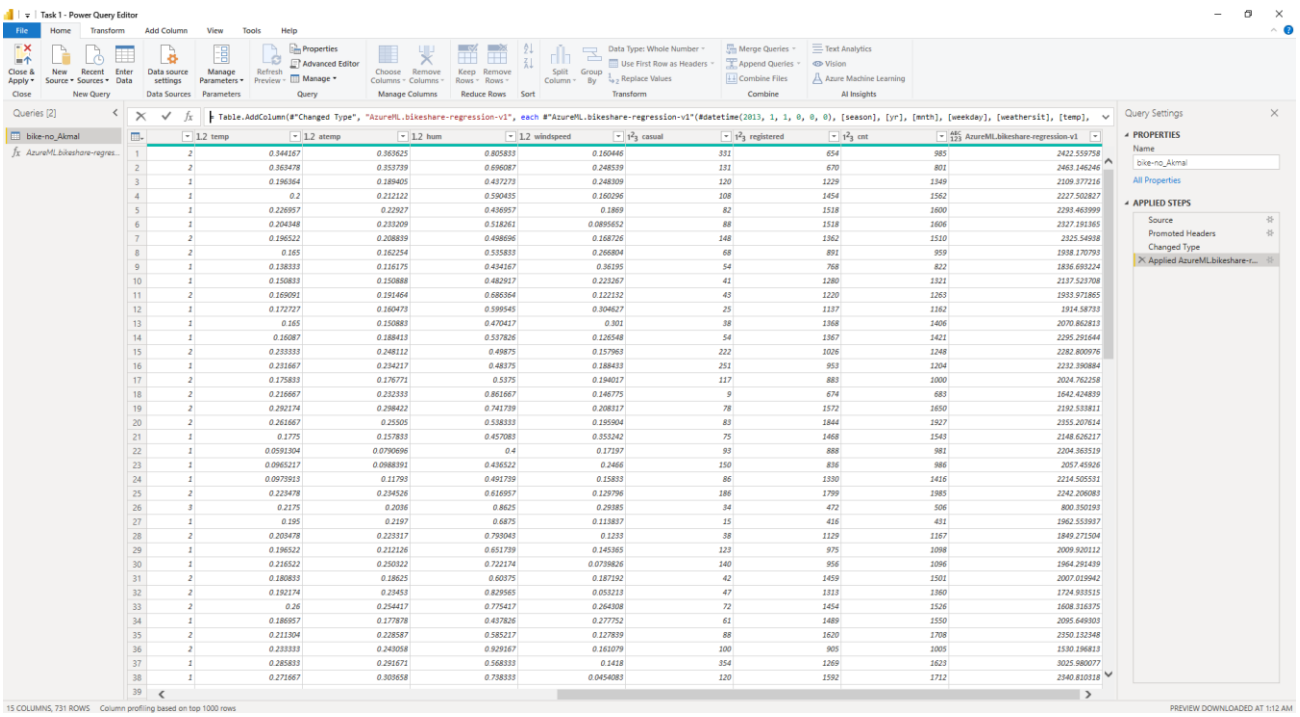




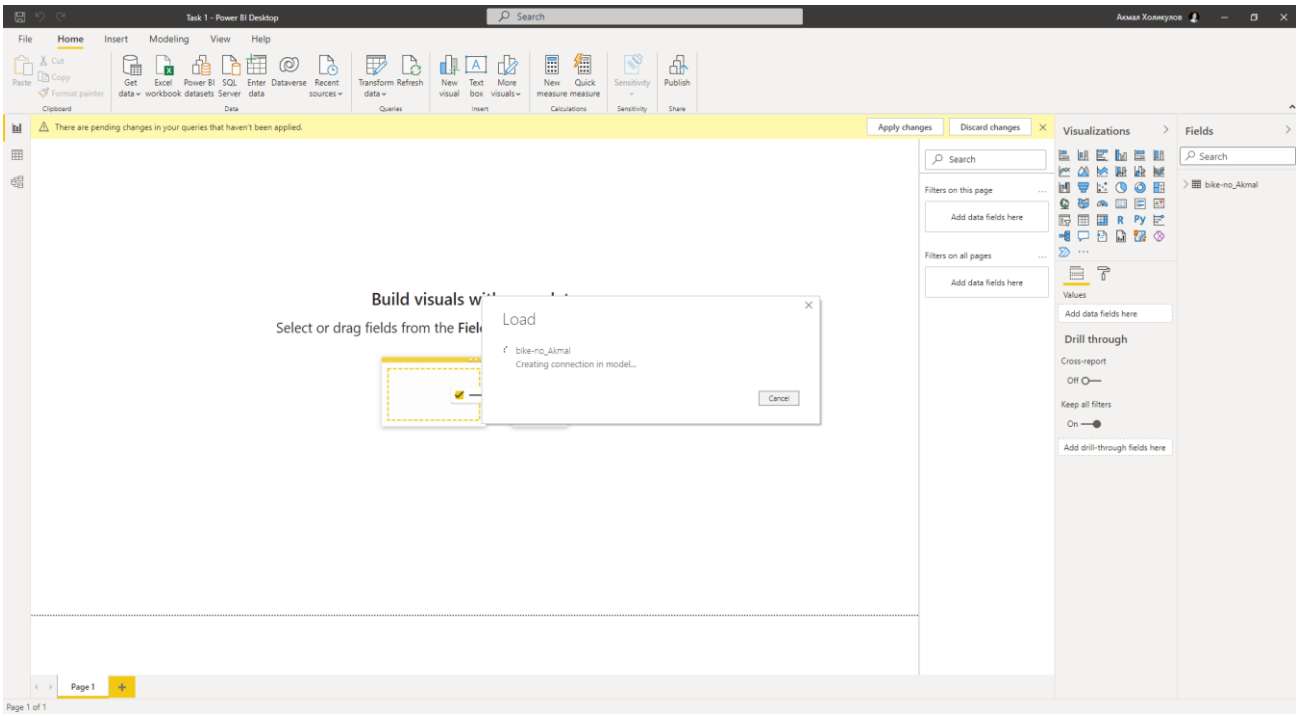
Then open that data by clicking “transform data”. In the window (Power Query Editor) we click on “Azure Machine Learning”. Choose deployed model from Azure ML (bikeshare-regression-v1), then put some data to columns. Click “OK”.



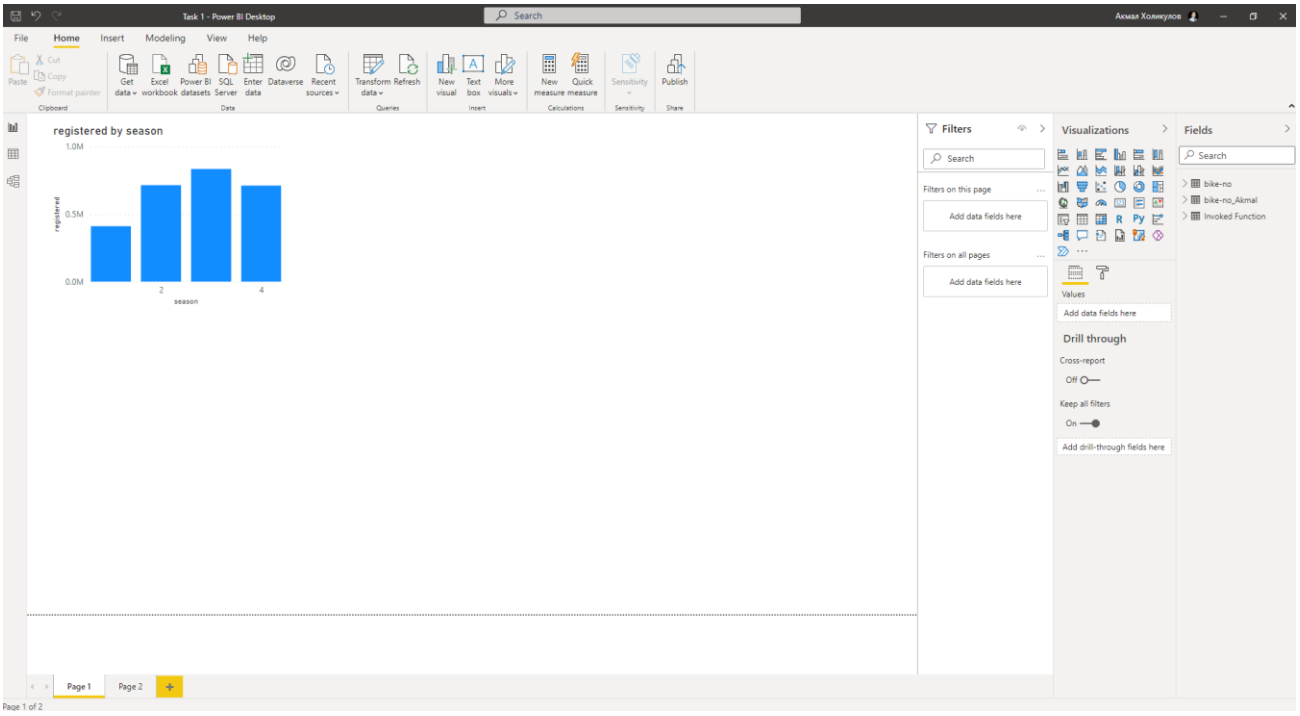
Here is the calculation made by model



Click on “Close and apply”



After that we can visualize the data. For example, we take a “season” on X axis and “registered by season” on Y axis.



What other Data Science projects can be implemented for a bikeshare company? Share your ideas in the report

I think there are two data analysis algorithms that could be implemented for a bike rental company.

First, there is logistic regression. We have a lot of data on weather conditions over the past few years. By applying a logistic regression algorithm, we can use it to predict weather conditions, whether it is snow or rain (binary classification of data points). Bicycle rentals are directly dependent on weather conditions, because customers are not inclined to rent bicycles in bad weather conditions.

Second, linear regression. The number of people renting bicycles has increased since the start of the business. To measure the relationship between two continuous variables, we can apply a linear regression algorithm. For example, the number of rented bicycles on the Y-axis and the number of years on the X-axis. After applying linear regression, we can see the trend and based on that information we will know how many bikes we need to buy in advance for next years.