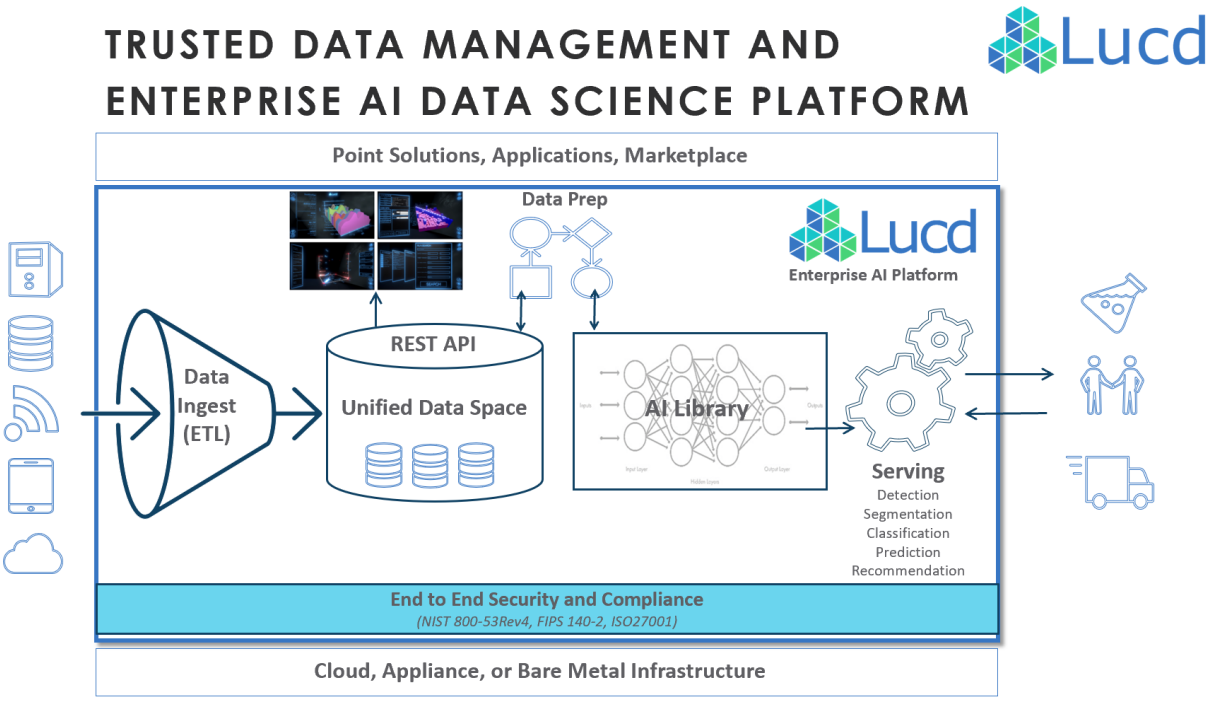
Tutorial: Developing and Training a Classification Model in the Lucd Data Science Platform

The Lucd Enterprise AI Data Science Platform is a highly secure, scalable, open and flexible platform for persisting an fusing large and numerous datasets and training AI models for production against those datasets.

The Lucd platform is an end to end platform that can be deployed in public cloud environments, on premise on bare metal hardware, or the Lucd multi-tenant PaaS can be directly accessed. The platform consists of:

1) A scalable open data ingest capability
2) A petabyte scale unified data space data repository
3) 3-D Visualization and Exploration
4) An Exploratory Data Analysis Rest Service
5) A Kubernetes environment to train PyTorch and TensorFlow models
6) NLP Word Embedding and Explainable AI Assets
7) Model results visualization and exporting to internal or external serving capability



Introduction

The purpose of this tutorial is to get a feel for leveraging the Lucd Enterprise AI Data Science Platform. The Lucd platform if flexible and can perform many things. But, this tutorial will specifically walk step by

step through training a custom developed classification model. While many standard and custom models can be trained on many different datasets, this tutorial will focus on this specific use case. <describe data set> <describe objective> (i.e. what is trying to be achieved in this tutorial.

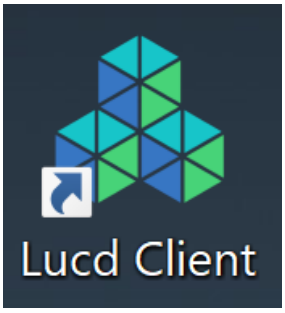
Step 1: Deploy Lucd Platform

<Outside of Scope for this tutorial> <Not required if accessing the Lucd Multitenant PaaS>. This step is out of scope for this tutorial. Refer to:

1) Lucd Infrastructure Requirements: https://community.lucd.ai/hc/en-us/articles/360037762592-Infrastructure-Requirements-v6-2-0
2) Lucd Deployment Guide: https://community.lucd.ai/hc/en-us/articles/360037762792-Deployment-Guide-v6-2-0
3) Create Security and Access Framework, Create User Account(s) <link>

Step 2: Load Lucd Client UI

The Lucd 3D Client UI is the main interface to Lucd.



Obtaining the Lucd Client UI and Accounts is outside of scope of this tutorial. Refer to: <need to provide an external link> this link is internal https://community.lucd.ai/hc/en-us/articles/360037995531-Lucd-Client-STEAM-Installation-Guide-Lucd-Internal-

Step 3: Ingest Data

Setting up NiFi and Ingesting data is outside the scope of this tutorial. Refer to:
https://community.lucd.ai/hc/en-us/articles/360038129271-NiFi-Configuration-v6-2-0

Step 4: Login to Platform, Visualize and Explore Data

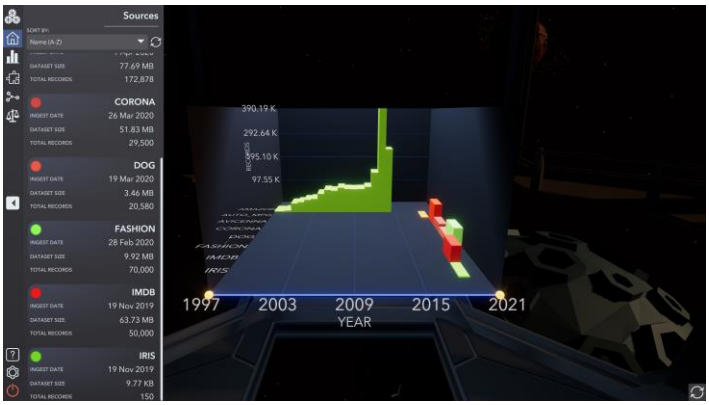
At the end of step 2, there should be a Lucd Client Icon loaded on your Desktop.

- 4.1: Double Click on that Icon, the following screen should launch:

- Step 4.2: Enter the Username, Password, and Domain that you obtained at Step 2. Then click LOGIN. The following screen should appear:

- Step 4.3: Click to refresh data sources

- A screen similar to the following should appear



From here, there are many visualization, EDA and AI training operations that can be done. This tutorial is focusing on a specific flow, so documenting every capability is outside the scope of this tutorial. Refer here for a full capability description:

<Refer to user guide: https://community.lucd.ai/hc/en-us/articles/360022853292-Lucd-Client-User-Guide

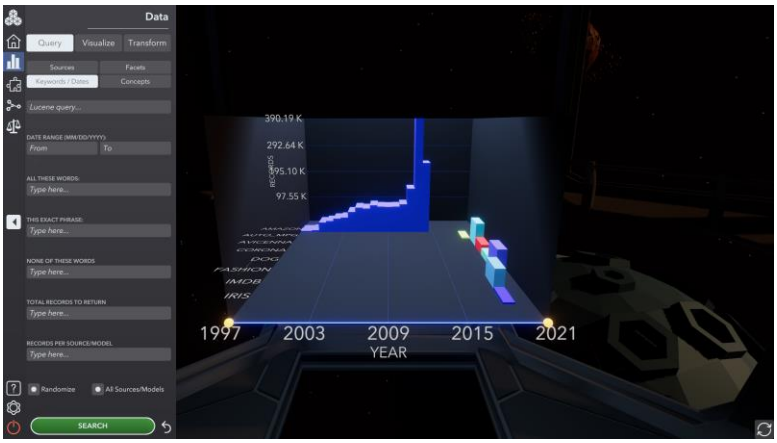
For this tutorial, at this point we will proceed to Step 5.

Step 5: Create Virtual Data Set (VDS).

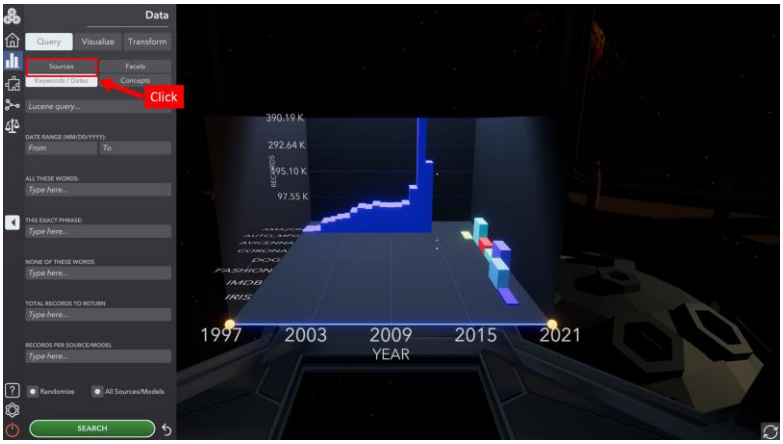
In this section we will create a VDS of the IRIS Dataset. There are many different ways to search on data in the Lucd Unified Data Space, fuze and merge multiple datasets in the UI, and perform EDA operations, but for purposes of this tutorial we will just focus on creating a VDS from the IRIS Dataset.

- Step 5.1, from the Home Screen, select Data:

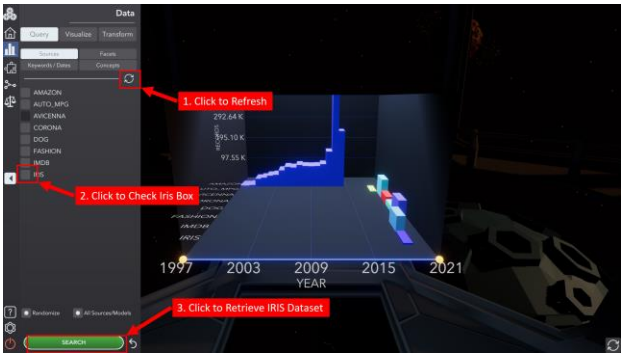
- The following view should result:



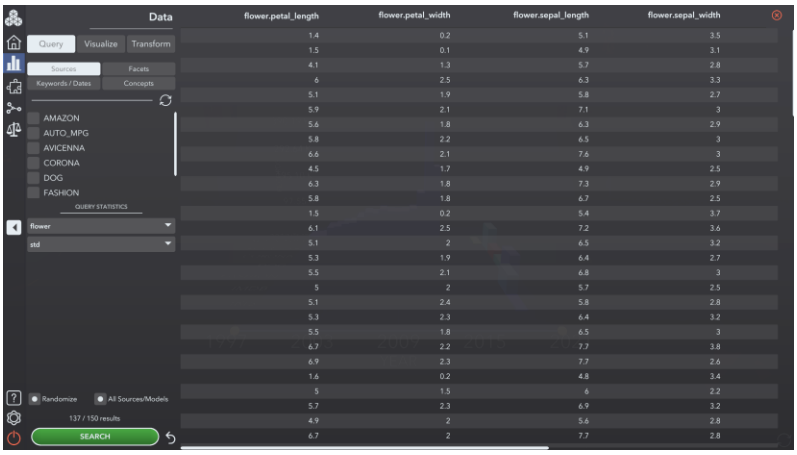
- Click on Sources:



- Step 5.2, on the following screen, refresh the sources, click on Iris, and click search per the following:

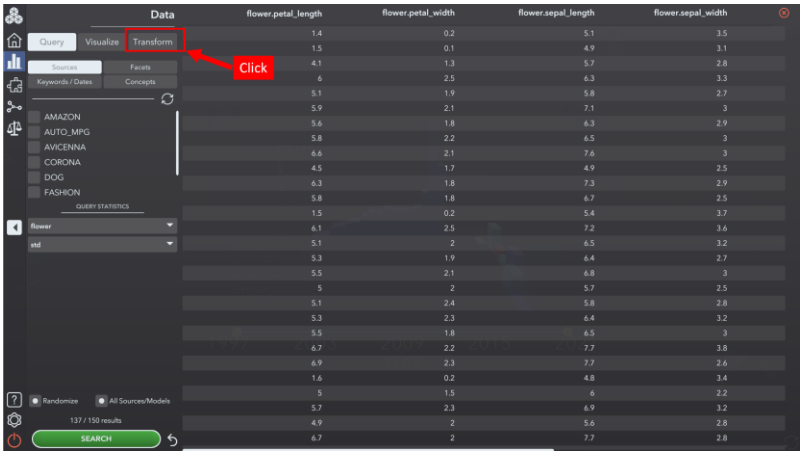


The following should result:

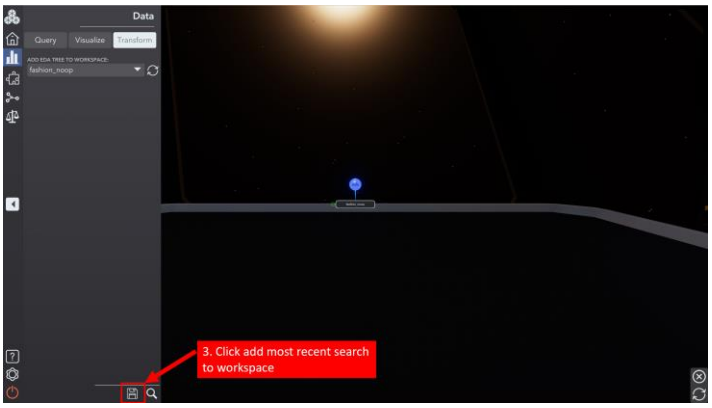


Per the above, there are many operations that can be performed to Visualize and Transform the results of a UDS search. However for this tutorial, we will just create a VDS out of the search results.

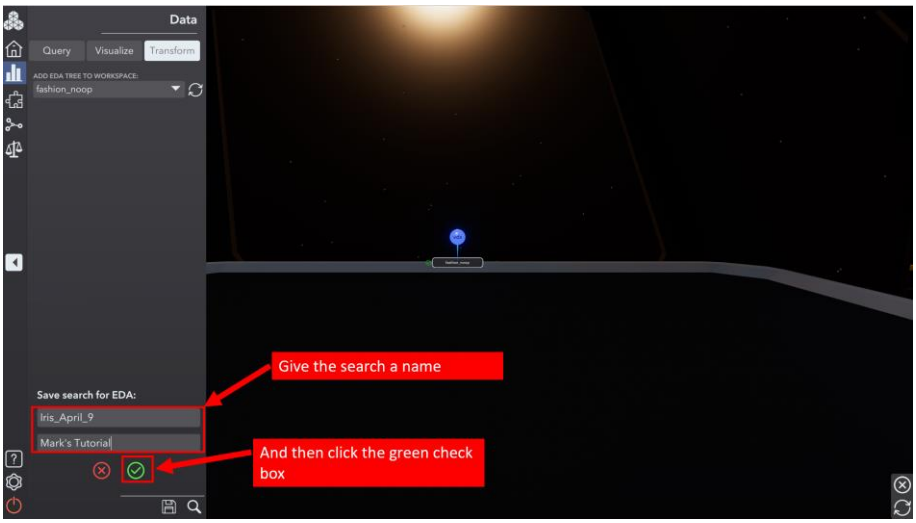
- Step 5.3 Click the Transform Button:



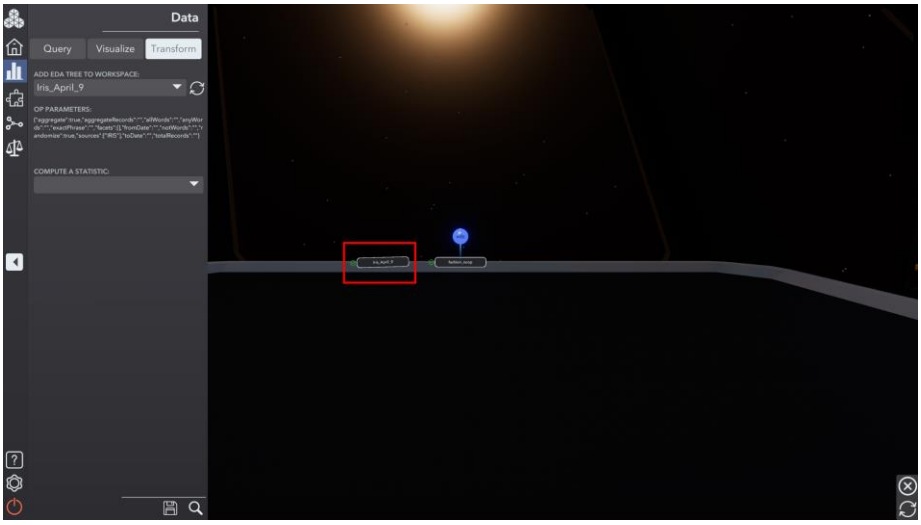
- Step 5.4 Click add most recent search to workspace



- Give the search a name, and a description, then hit the green check mark

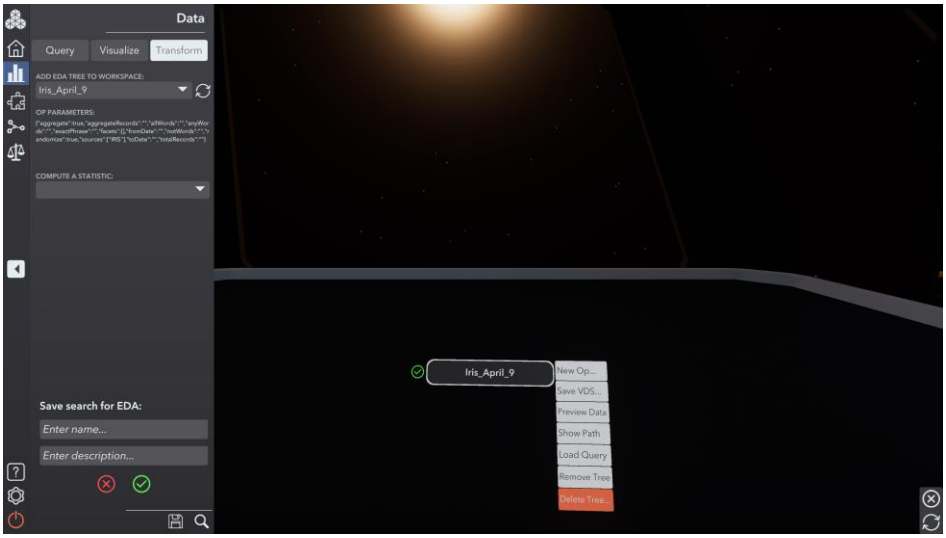


- The search name should now appear on the workspace

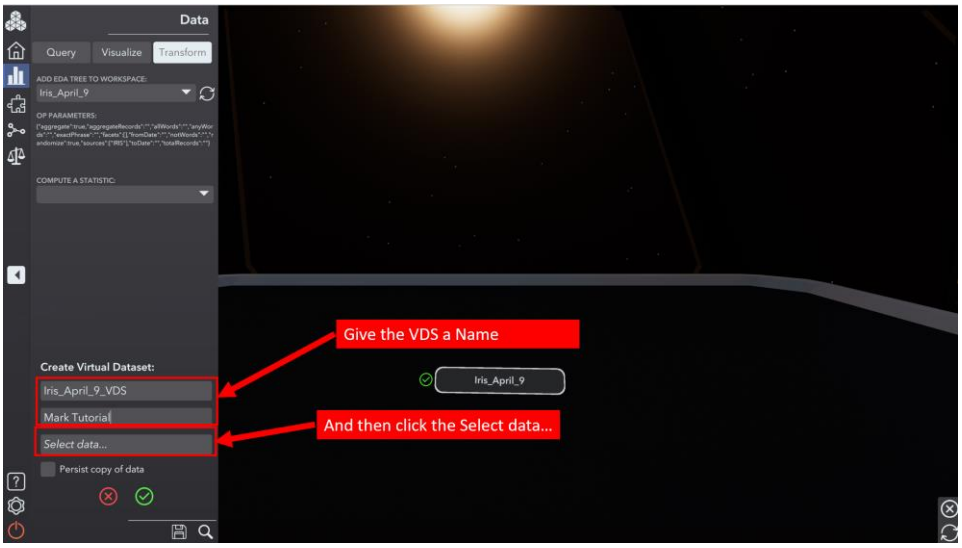


The Lucd 3D Client supports touch screens and you should be able to zoom, rotate (or pinch your touchscreen) to make the search result larger. You can experiment with that. For now, if you click and hold the search name and pull forward it will come forward larger (you can hit the refresh button in the bottom right corner of the screen if you want to go back to the original view

- Step 5.5, click the green check mark on the left next to the search result and various operations appear to the right:

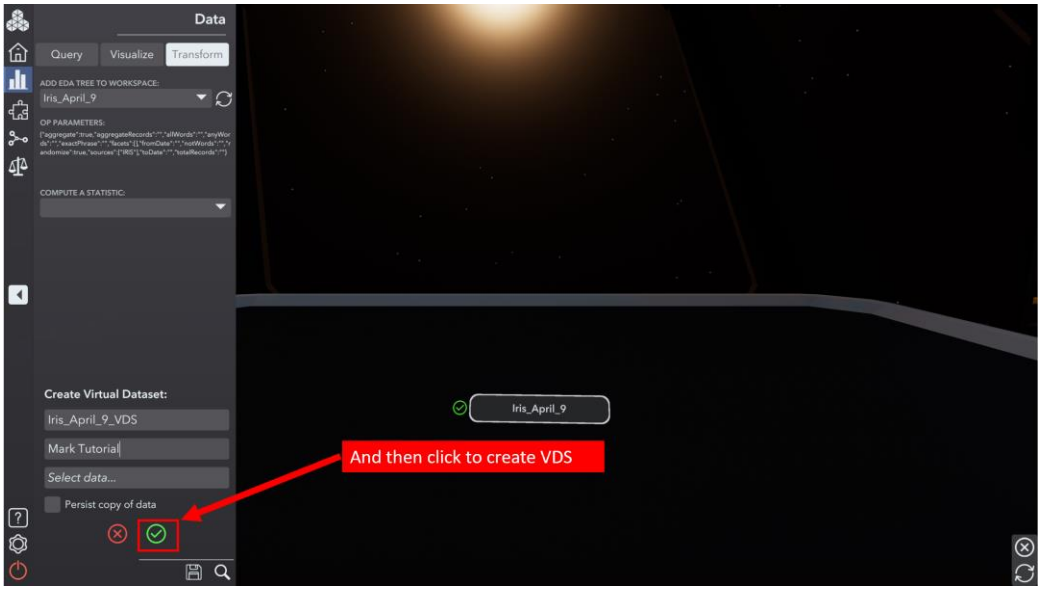


- Per the above, there are many operations that can be performed on the search result before creating a Virtual Data Set (VDS). But for now, we will create a VDS directly from the search result by clicking on the Save VDS
- After clicking on the Save VDS, the following will appear, Give the VDS a name and description and the click Select data

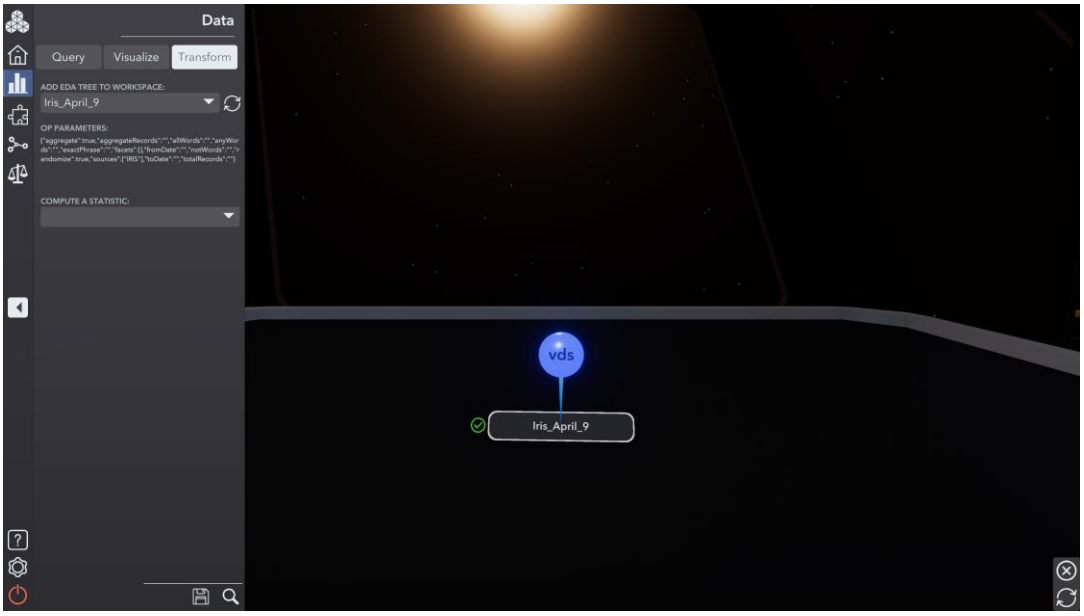


- The following screen will appear. If we were to start training at this time, we would select the data for the features and the data as the label (we could do this right now to start training). But, for the purposes of this Tutorial, we are going to create a custom data operation to add a new field first, so for now, make sure all are selected and hit the green check mark

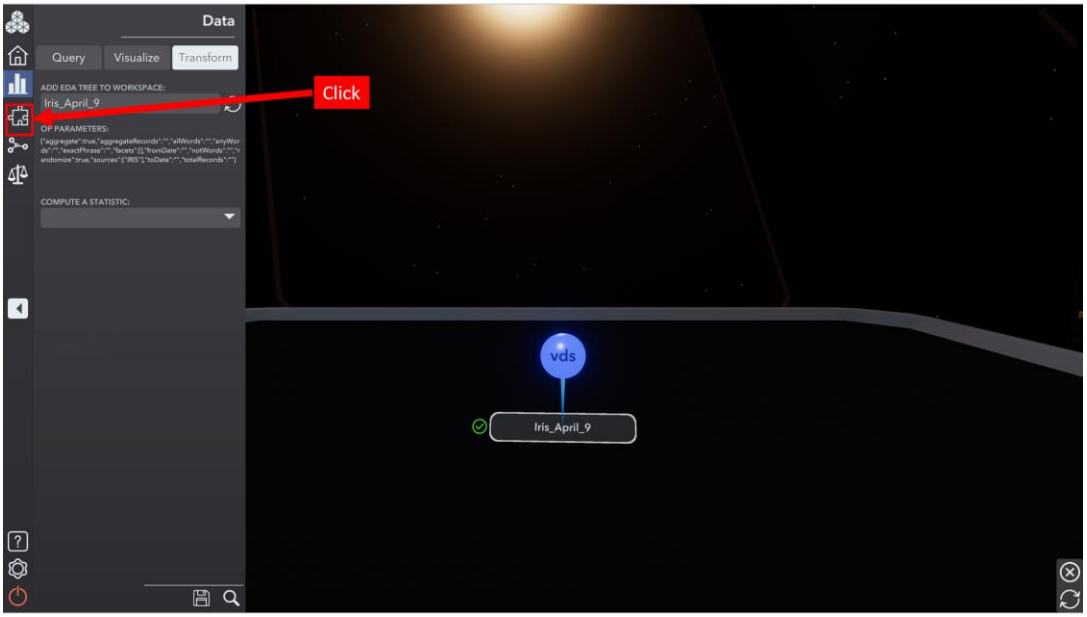
- After that, the previous screen will appear, hit the green check mark.



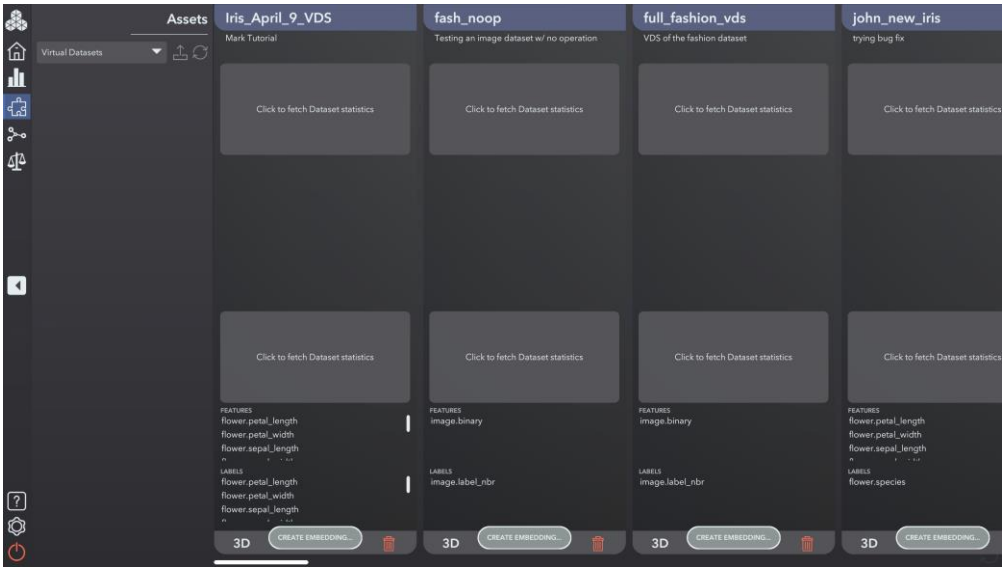
- A blue “VDS” will the appear above the search result in the workspace:



- You can check that the VDS is present by clicking on the assets button



- Click the refresh button and the Virtual Data Set that you created should appear:



Now that we have created a VDS, we will create a custom operation in a Jupyter Notebook in Step 6.

Step 6: Use Jupyter Notebook to create Custom EDA Operation

For Step 6, a Jupyter Notebook is required. Recommend leveraging a notebook from Anaconda. If not previously done, you can install Anaconda from: https://www.anaconda.com/distribution/

Once Anaconda is installed on your machine, it is recommended that you create a new environment either via that Anaconda Prompt to via Anaconda Navigator. You can read Anaconda documentation for how to do this. You can name the environment as whatever you want, for the purposes of this tutorial we will assume that the environment name is “Lucd-python”. Add or ensure the following packages are installed on “Lucd-python” (you can follow the conda install steps from Anaconda documentation):

- Dask
- Dask Distributed
- Dask Machine Learning
- Tensorflow
- PyTorch
- lxml
- cryptography (should already be installed)

Step 6.1: Obtain Lucd-python client package (in the future, will be able to pip or conda install), from your Account Manager, unzip the contents into a folder on your machine, not the location:

lucd-python-package.zip

Step 6.2: In either Anaconda Navigator or Anaconda command line, (while in the Lucd-env), navigate to that folder in the directory structure and run:

- python setup.py

This will install the Lucd-python package

Step 6.3: Within the Lucd environment, launch a Jupyter Notebook

- In the first cell import the following:

from lucd import LucdClient, log

from eda.int import asset

from eda.int import vds

from eda.int import uds

from eda.lib import lucd_uds

```
from lucd import LucdClient, log
from eda.int import asset
from eda.int import vds
from eda.int import uds
from eda.lib import lucd_uds
```

- In the next cell enter your login credentials:

client = lucd.LucdClient(domain="<your domain>",

username="<your username>",

password="<your password>",

login_domain="<your domain>"

)

```
client = lucd.LucdClient(domain="<your domain>",
                         username="<your username>",
                         password="<your password>",
                         login_domain="<your domain>"
                         )
```

Now you should be able to see all your VDSs.

all_vds, http = vds.read({"uid": "<your username>"})

```
all_vds, http = vds.read({"uid": "<your username>"})
```

Then entering: all_vds into the next cell

Or just enter:

for my_dict_list in all_vds:

print(all_vds[my_dict_list]['name'] + " is from key: " + my_dict_list)

```
for my_dict_list in all_vds:
    print(all_vds[my_dict_list]['name'] + " is from key: " + my_dict_list)
```

You can see the VDS that we created, highlight the following (note you will see your login name vs demo

```
for my_dict_list in all_vds:
    print(all_vds[my_dict_list]['name'] + " is from key: " + my_dict_list)
```

```
Iris_April_9_v3 is from key: demo_9223370450392300426
Iris_April_9_VDS_v2 is from key: demo_9223370450403986295
Iris_April_9_VDS is from key: demo_9223370450406517386
```

Paste that info into the following cell:

df = lucd_uds.get_dataframe("demo_9223370450406517386", limit=100)

```
df = lucd_uds.get_dataframe("demo_9223370450406517386", limit=100)
```

```
2020-04-10 09:34:29,618 | root | INFO | dask.py:28 | Creating Dask LocalCluster: http://localhost:60000/status
```

Error

Step 7: Upload custom model to Lucd Platform

<Provide Steps>

Step 8: Train custom model in Lucd Platform

<Provide Steps>

Step 9: Export trained custom model to internal or external serving.

Outside the scope of this document

<Provide link>

Summary and Additional References

<Provide link>