PRINCE DR K VASUDEVAN COLLEGE OF ENGINEERING AND TECNOLOGY
(Mambakkam - Medavakkam Main Rd, Ponmar, Chennai, Tamil Nadu 600127)
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING.

Machine Learning based Vehicle Performance Analyzer

Team ID : PNT2022TMID38136

Team Size : 4

Team Leader : SANJAY V (sanjay1332002@gmail.com)

Team member : JOSHUA EBINESH B (joshuaebinesh50@gmail.com)

Team member : DOREN SAM JOSEPH B (dorenbd2001@gmail.com)

Team member : DHANUSH P (p.dhanushdhanu2001@gmail.com)

SI EMAIL : 411619104020@smartinternz.com

SI PASSWORD : PNTIBMNw10

WEBMAIL : https://sg2plmcpnl492529.prod.sin2.secureserver.net:2096/

GIT USRERNAME

SANJAY V : sanjay13-jai

JOSHUA EBINESH B : joshuaebinesh

DOREN SAM JOSEPH B : Doren-Sam-Joseph04

DHANUSH P : Dhanush-21

ASSIGNMENT DETAILS:

ASSIGNMENT 1: 1. Split this string, *2. Use.format() to print the following string, 3.In this nest dictionary grab the word "hello", 4.1 Create an array of 10 zeros, 4.2 Create an array of 10 fives, 5. Create an array of all the even integers from 20 to 35, 6. Create a 3x3 matrix with values ranging from 0 to 8, 7. Concatenate a and b, 8. Create a dataframe with 3 rows and 2 columns, 9. Generate the series of dates from 1st Jan, 2023 to 10th Feb, 2023, 10. Create 2D list to DataFrame***.**

ASSIGNMENT 2: 1. Download the dataset: Dataset 2. Load the dataset. 3. Perform Below Visualizations.Univariate Analysis. Bi - Variate Analysis. Multi - Variate Analysis 4. Perform descriptive statistics on the dataset. 5. Handle the Missing values. 6. Find the outliers and replace the outliers 7. Check for Categorical columns and perform encoding. 8. Split the data into dependent and independent variables. 9. Scale the independent variables 10. Split the data into training and testing

ASSIGNMENT 3: 1. Download the dataset: Dataset. 2. Load the dataset into the tool. 3. Perform Below Visualizations. • Univariate Analysis. • Bi-Variate Analysis. • Multi-Variate Analysis. 4. Perform descriptive statistics on the dataset. 5. Check for Missing values and deal with them. 6. Find the outliers and replace them outliers. 7. Check for Categorical columns and perform encoding. 8. Split the data into dependent and independent variables. 9. Scale the independent variables. 10. Split the data into training and testing. 11. Build the Model. 12. Train the Model. 13. Test the Model. 14. Measure the performance using Metrics.

ASSIGNMENT 4: 1. Download the dataset: Dataset. 2. Load the dataset into the tool. 3. Perform Below Visualizations. • Univariate Analysis. • Bi- Variate Analysis. • Multi-Variate Analysis. 4. Perform descriptive statistics on the dataset. 5. Check for Missing values and deal with them. 6. Find the outliers and replace them outliers. 7. Check for Categorical columns and perform encoding. 8. Scaling the data.9. Perform any of the clustering algorithms. 10. Add the cluster data with the primary dataset. 11. Split the data into dependent and independent variables. 12. Split the data into training and testing. 13. Build the Model. 14. Train the Model. 15. Test the Model. 16. Measure the performance using Evaluation Metrics.

DATE OF FINISHING

S.NO	NAME	ASSIGNMENT 1	ASSIGNMENT 2	ASSIGNMENT 3	ASSIGNMENT 4
1	SANJAY V	26-09-2022	27-09-2022	04-11-2022	05-11-2022
2	JOSHUA EBINESH B	26-09-2022	28-09-2022	04-11-2022	05-11-2022
3	DOREN SAM JOSEPH B	27-09-2022	28-09-2022	04-11-2022	05-11-2022
4	DHANUSH P	27-09-2022	27-09-2022	04-11-2022	05-11-2022

TASK ASSIGNED:

PROJECT OBJECTIVES:

Predicting the performance level of cars is an important and interesting problem. The main goal is to predict the performance of the car to improve certain behaviours of the vehicle. This can significantly help to improve the system's fuel consumption and increase efficiency.

The performance analysis of the car is based on the engine type, no of engine cylinders, fuel type, horsepower, etc. These are the factors on which the health of the car can be predicted. It is an ongoing process of obtaining, researching, analyzing, and recording health based on the above three factors. The performance objectives like mileage, dependability, flexibility and cost can be grouped together to play a vital role in the prediction engine and engine management system. This approach is a very important step towards understanding the vehicle's performance.

PRE REQUISITES:

1. INSTALL ANACONDA (SANJAY V & JOSHUA EBINESH B)
2. INSTALL PYTHON PACKAGES (DOREN SAM JOSEPH B)
3. PRIOR KNOWLEDGE (DHANUSH.P)

PRE-REQUISITES

1.INSTALL ANACONDA(JOSHUA EBINESH B, SANJAY V) :

1. Download the Anaconda installer.
2. Go to your Downloads folder and double-click the installer to launch. To prevent permission errors, do not launch the installer from the Favorites folder.

Note

If you encounter issues during installation, temporarily disable your anti-virus software during install, then re-enable it after the installation concludes. If you installed for all users, uninstall 3.Anaconda and re-install it for your user only.

Click **Next**.

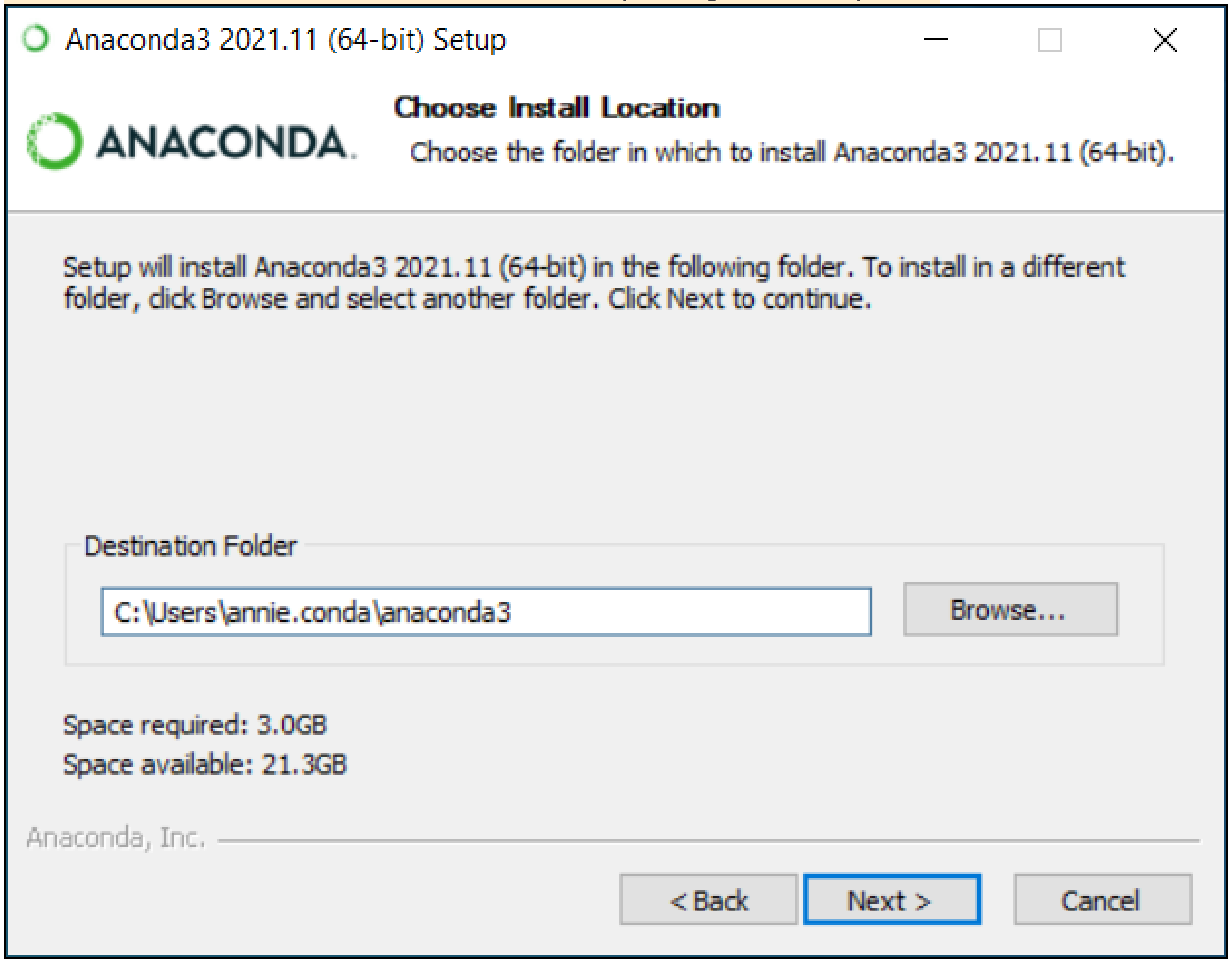
3. Read the licensing terms and click **I Agree**.
4. It is recommended that you install for **Just Me**, which will install Anaconda

Distribution to just the current user account. Only select an install for **All Users** if you need to install for all users' accounts on the computer (which requires Windows Administrator privileges).

4.Select a destination folder to install Anaconda and click **Next**. Install Anaconda to a directory path that does not contain spaces or unicode characters. For more information on destination folders.

Caution

Do not install as Administrator unless admin privileges are required.



Anaconda3 2021.11 (64-bit) Setup

Choose Install Location

Choose the folder in which to install Anaconda3 2021.11 (64-bit).

Setup will install Anaconda3 2021.11 (64-bit) in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder

C:\Users\annie.conda\anaconda3

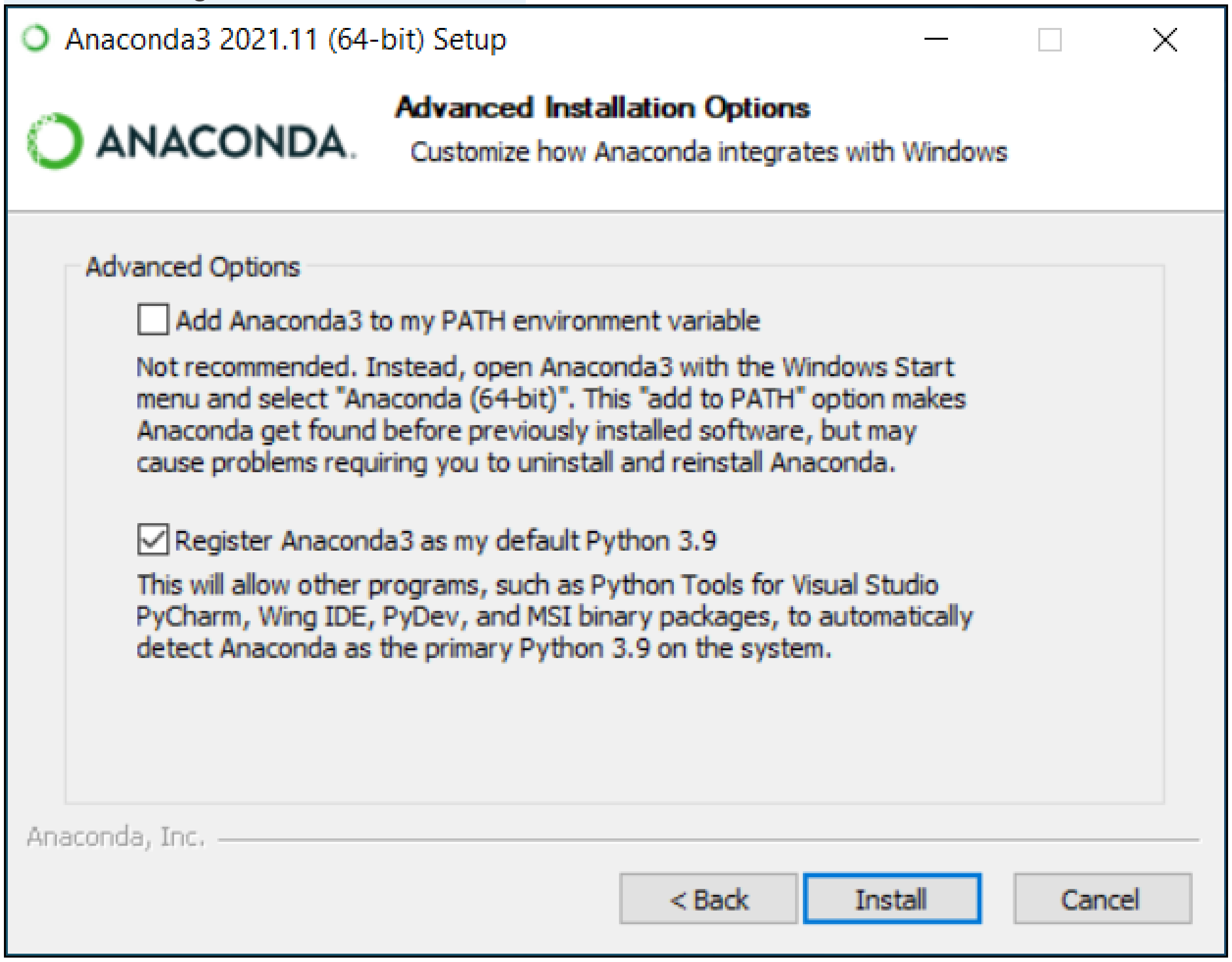
Space required: 3.0GB
Space available: 21.3GB

5. Choose whether to add Anaconda to your PATH environment variable or register Anaconda as your default Python. We **don't recommend** adding Anaconda to your PATH environment variable, since this can interfere with other software. Unless you plan on installing and running multiple versions of Anaconda or multiple versions of Python, accept the default and leave this box checked.

Instead, use Anaconda software by opening Anaconda Navigator or the Anaconda Prompt from the Start Menu.

Note

As of `Anaconda Distribution 2022.05`, the option to add Anaconda to the PATH environment variable during an **All Users** installation has been disabled. This was done to address a security exploit. You can still add Anaconda to the PATH environment variable during a **Just Me** installation.


Anaconda3 2021.11 (64-bit) Setup
ANACONDA.
Advanced Installation Options
Customize how Anaconda integrates with Windows
Advanced Options
Anaconda, Inc.
< Back
Install
Cancel

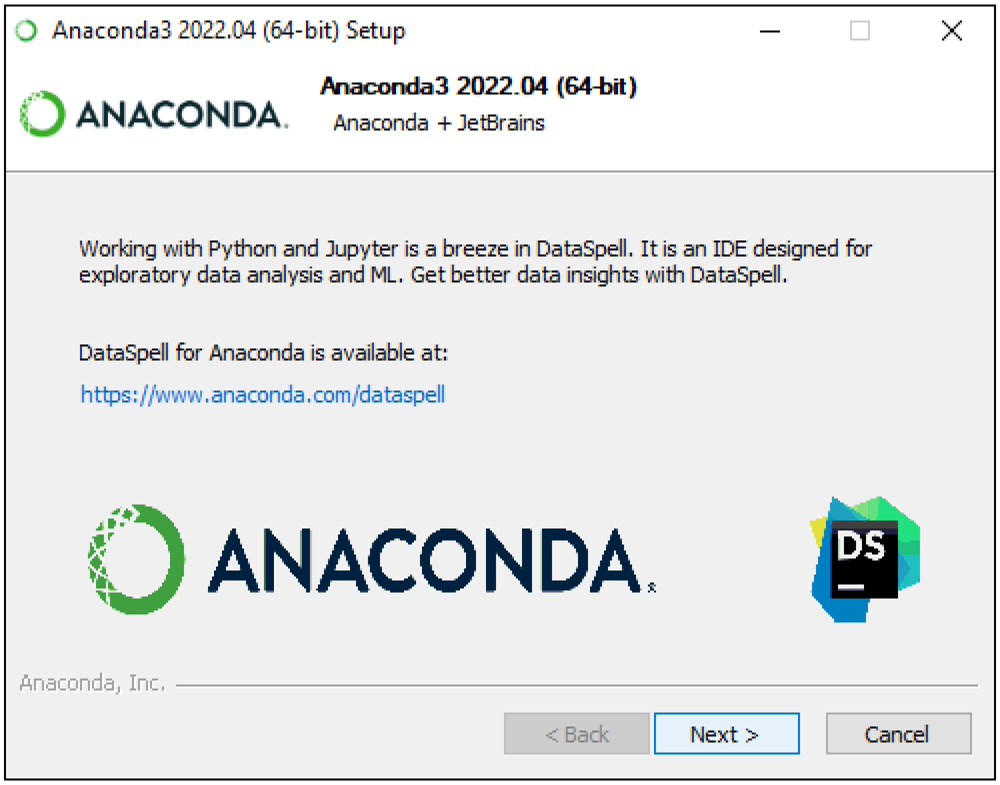
☐ Add Anaconda3 to my PATH environment variable

Not recommended. Instead, open Anaconda3 with the Windows Start menu and select "Anaconda (64-bit)". This "add to PATH" option makes Anaconda get found before previously installed software, but may cause problems requiring you to uninstall and reinstall Anaconda.

☑ Register Anaconda3 as my default Python 3.9

This will allow other programs, such as Python Tools for Visual Studio PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically detect Anaconda as the primary Python 3.9 on the system.

6. Click **Install**. If you want to watch the packages Anaconda is installing, click Show Details.
7. Click **Next**.
8. Optional: To install Dataspell for Anaconda, click https://www.anaconda.com/dataspell.


Anaconda3 2022.04 (64-bit) Setup

Anaconda3 2022.04 (64-bit)
Anaconda + JetBrains

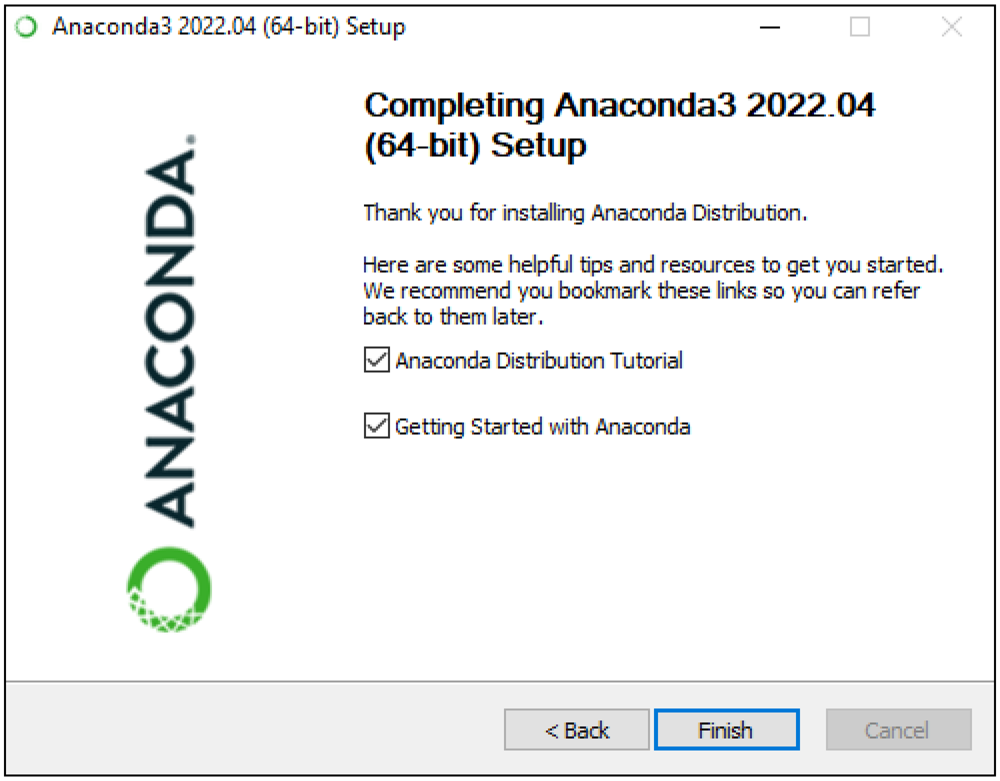
Working with Python and Jupyter is a breeze in DataSpell. It is an IDE designed for exploratory data analysis and ML. Get better data insights with DataSpell.

DataSpell for Anaconda is available at:

https://www.anaconda.com/dataspell

Anaconda, Inc.

9. Or to continue without Dataspell, click **Next**.
10. After a successful installation you will see the “Thanks for installing Anaconda” dialog box:


Anaconda3 2022.04 (64-bit) Setup

Completing Anaconda3 2022.04 (64-bit) Setup

Thank you for installing Anaconda Distribution.

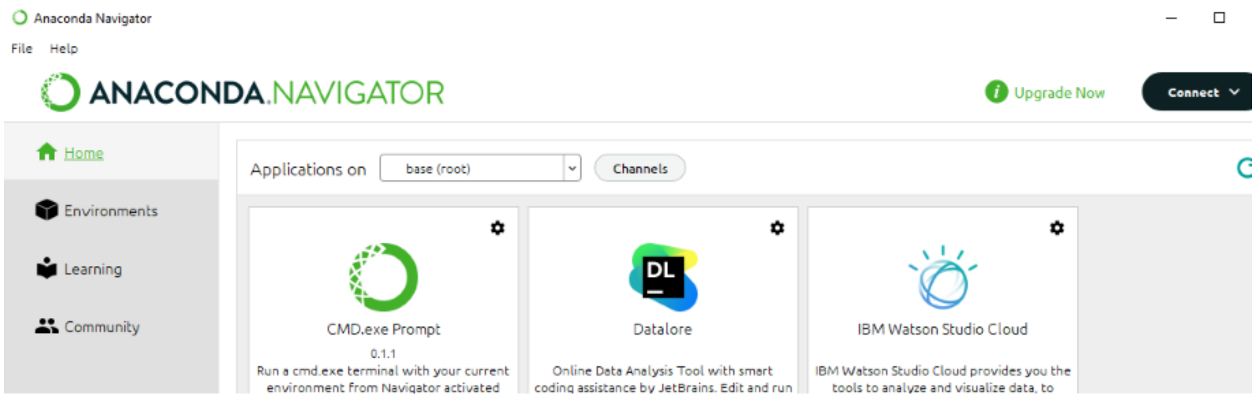
Here are some helpful tips and resources to get you started. We recommend you bookmark these links so you can refer back to them later.

- [✓] Anaconda Distribution Tutorial
- [✓] Getting Started with Anaconda

11. If you wish to read more about Anaconda.org and how to get started with Anaconda, check the boxes “Anaconda Distribution Tutorial” and “Learn more about Anaconda”. Click the **Finish** button.
12. Verify your installation.

2. INSTALL PYTHON PACKAGES (DOREN SAM JOSEPH B)

Step 1: Open your Anaconda navigator



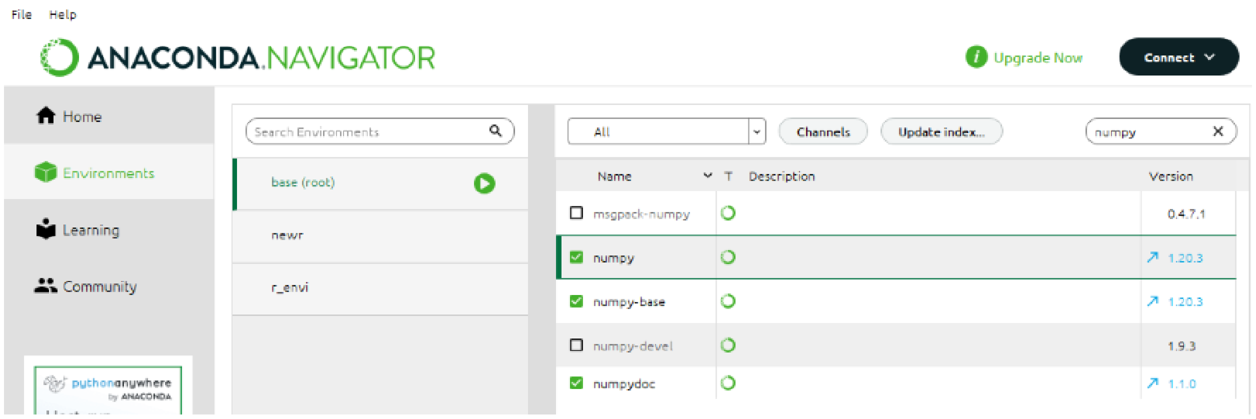
Run a cmd.exe terminal with your current environment from Navigator activated

Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run

IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to

Step 2:

- Go to the Environment tab.
- Search your package in the upper right search bar.
- Check the package you want to install.



	Name	T	Description	Version
☐	msgpack-numpy			0.4.7.1
✓	numpy			↗ 1.20.3
✓	numpy-base			↗ 1.20.3
☐	numpy-devel			1.9.3
✓	numpydoc			↗ 1.1.0

Step 3: Here, you can click on **apply** to install or update your packages.

3. PRIOR KNOWLEDGE (DHANUSH.P)

Here at Anaconda, our mission has always been to make the art of data science accessible to all. We strive to empower people to overcome technical obstacles to data analysis so they can focus on asking better questions of their data and solving actual, real-world problems. With this goal in mind, we’re excited to announce our recently revamped Anaconda Training program!

Drawing from our own in-house experiences, we’ve distilled the principle components of a professional data science career into a carefully crafted Data Science Learning Path. The path is ideal for both practicing professionals who need to learn Python, and people who already have some Python training but want to further sharpen their skills.

Our Data Science Learning Path begins with an introduction to the Anaconda Ecosystem, because, well, data scientists usually need to share their data analysis workflows. Both the open source Anaconda Distribution and commercial Anaconda Enterprise incorporate conda, our fully open source, cross-platform environment and package manager. We’ll show you how conda helps easily capture, deploy, and share reproducible computational environments including data and code.

Next we'll take you through the fundamentals of data import and export, manipulation, and analysis. As data scientists, we sometimes assert that 80% of our jobs are spent "wrangling" or "munging" our data; the courses in our Learning Path will teach you how to make this process as painless, straightforward, and effective as possible. Prior to cleaning data, even loading data requires familiarity with the right technologies. We'll show you the best tools and techniques for this as well as for tasks such as correcting errors, compensating for missing values, identifying outliers, merging datasets, and reshaping datasets.

Our Data Science Learning Path will then cover how to use the Anaconda Ecosystem to apply key skills to statistical analysis, machine learning, and visualization. While the theoretical foundations of machine learning can be daunting, our experts will help you see that the essential elements as applied in industry are not difficult to understand or implement, particularly thanks to vibrant open source communities. A data scientist's deliverable to clients often bundles visualizations and statistics to convey the results of machine learning as succinct data-driven stories. Our training program will give you a familiarity with appropriate graphics packages and, more importantly, a certain graphical literacy.

Finally, it's impossible to be a data scientist these days without knowing how to work with big data. The proliferation of commodity hardware and cloud technologies enables individuals to harness enormous amounts of data easily; a good data scientist must know when such technologies are necessary (and when they aren't!) and how to engage them. Our experts will help you get started with Anaconda-friendly parallel computing technologies with minimal fuss (for instance, to process data sets out-of-core).

DATA COLLECTION

1.DOWNLOAD DATASET(DOREN SAM JOSEPH B)

Our team download the dataset from collect datasets from different open sources like kaggle.com, data.gov, UCI machine learningrepository, etc.

DATA PREPROCESSING

1.Reading The Dataset(Joshua Ebinesh)

Starting a notebook is always easy, you just start a couple of cells which often just contain a `df.head()`. However, as the project grows (and in industry they always do), you will need to organize your folders. You will need a folder for the data, and another folder for notebooks. As the EDA progresses, you will need more folders representing different subsections of the main analysis. On top of that, your project should be reproducible, so that your peers can download the code, run the script, and it will work as intended, hopefully yielding the same results you had :)

So, if you have a `read_csv(relative_path_to_data)` on your notebook, moving it from one folder to another will require a change in the code. This is undesirable, we would like it to work regardless of its location. You could solve this by using `read_csv(absolute_path_to_data)`, but this is even worse: you will deal with paths lengthier than they need to be, and your code will probably break if you try to run it on another machine.

Let's say you have your working directory on `/system_name/project`, from which you run `jupyter lab` or `jupyter notebook`. The data directory is located at `/system_name/project/data`, and your notebooks are in `system_name/project/notebooks`

We propose two ways to solve this problem:

1. Using a environment variable inside a notebook
2. Using a data module

Environment Variable

With this approach, we inform the system the location of the data directory through the usage of an environment variable. Before starting a jupyter server, we will set the variable by doing

```
export DATA_DIR=system_name/project/data
```

If you are on the `/system_name/project` folder, you can do:

```
export DATA_DIR=$(pwd)/data
```

to achieve the same effect. Now, this variable is accessible to all child processes you start from your bash terminal. In your notebooks, you can now do:

Now the only thing your notebook needs to know is the `file_name` of the dataset. Sounds fair, right?

Another thing you can try to do is changing the working directory of the notebook itself by doing this:

This works, but I prefer the former, as the latter makes the notebook work in a directory that it is not, feels slightly shady :).

Finally, it might be a bit boring to set the environment variable every time you start a jupyter server. You can automate this process using python-dotenv. It will search for a `.env` file, first in the local directory, and then in all it's parents. Once it does, it will load the variables defined there. Check the project documentation if you like the idea!

Data Module

We used an environment variable to hold information about the project configuration, and exposed this to the notebook. But what about moving this responsibility somewhere else? We can create a module whose responsibility is to know the data directory, and where the datasets are. I prefer this approach, as it will make datasets explicit symbols inside the code.

We will need a `project_package` folder to represent, well, the project's package. Inside it, we will create a `project_data.py` module:

We use the `__file__` dunder method which returns the current file path, and the built-in `Path` class to navigate through the directory. We make this package installable by creating the `setup.py` inside the `project` folder:

We are almost there! Now, we install the package we just created in development mode, so that changes to the package won't require a reinstall:

```
python -m pip install -e .
```

This should install the `project_package` package, which can be accessed from the notebook:

This way, any notebook in any environment and location will access the data using the same method. If the data location changes, there's just one location we need to change: the `project_data.py` module.