Assignment: Example of implementing AI to predict Air Pollution Index based on meteorological data and specific time of the year

1. Introduction.

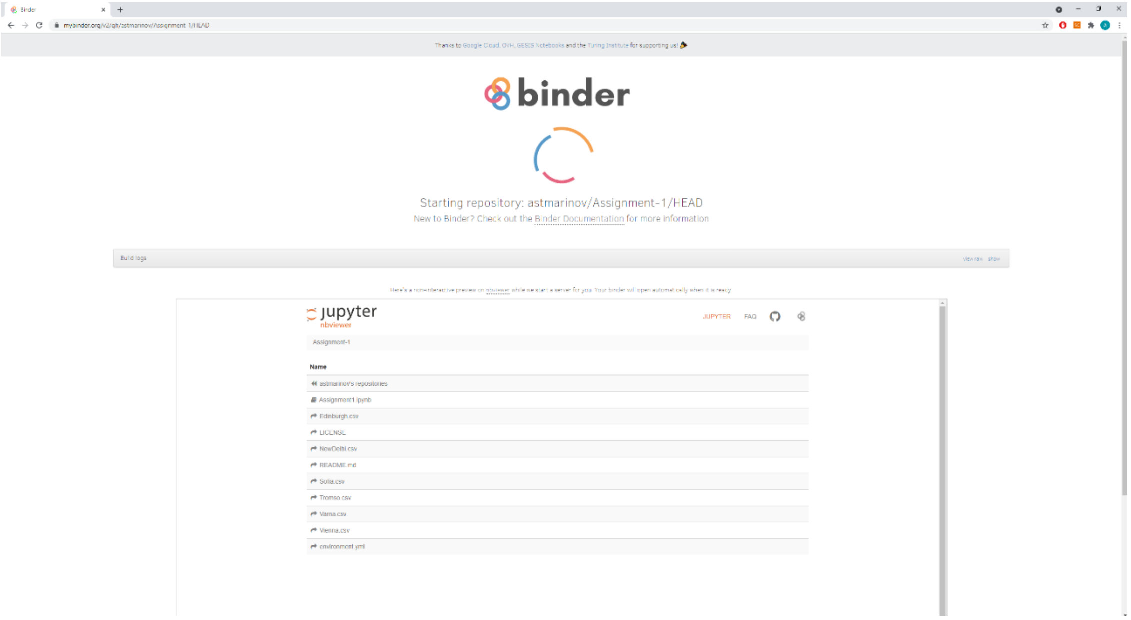
This assignment aims to demonstrate the use of AI in forming a simple application that predicts the air Quality index (AQI) for a specific city location. AQI is predicted per day based on weather information, the specific day of the week and the specific month. The classifier is based on the Random Forest algorithm. The application is build using the Python programing language and the Jupyter Notebook [3] environment. In order to avoid complex installations of Python environments and modules, the Notebook is running online from a binder server. The classifier is based on the Random Forest algorithm.

The aim of the assignment is to test key steps in using AI and draw conclusion related to training and usage of data.

Detailed descriptions of the application are provided in the notebook file.

2. Start-up instructions.

In order to start the notebook use the following link:
https://mybinder.org/v2/gh/astmarinov/Assignment-1/HEAD



Thanks to Google Cloud, OVH, GESIS Notebooks and the Turing Institute for supporting us!

Starting repository: astmarinov/Assignment-1/HEAD
New to Binder? Check out the Binder Documentation for more information

Here's a non-interactive preview on nbviewer while we start a server for you. Your binder will open automatically when it is ready.

Assignment-1

Name
astmarinov's repositories
Assignment1.ipynb
Edinburgh.csv
LICENSE
NewDelhi.csv
README.md
Sofia.csv
Tromso.csv
Varna.csv
Vienna.csv
environment.yml

Figure 1: Loading the notebook

This link will lead to a loading screen where the jupyter notebook server will be loaded (Figure 1). This may take different amount of time depending on internet connection and PC specifics. It takes more time when run for a first time.

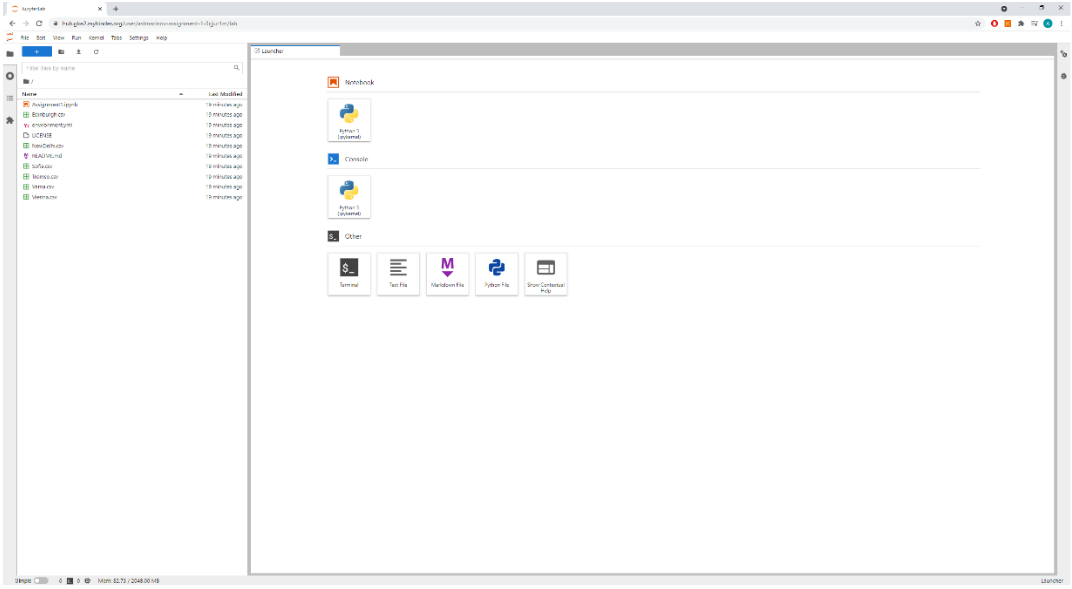
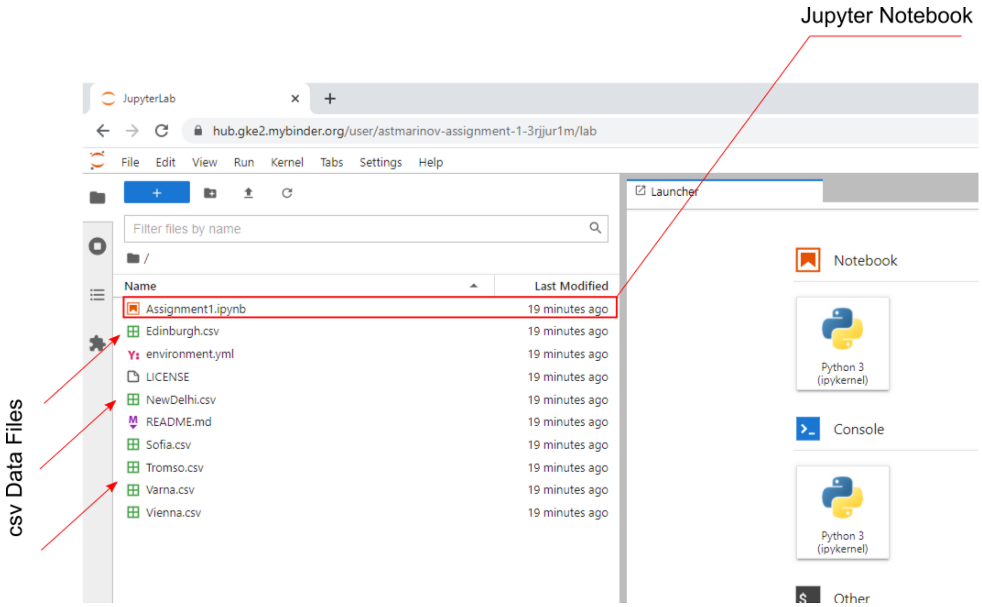


Figure 2: Jupyter notebook environment loaded

Once loaded the explorer of the environment will be presented (Figure 2). Form this window the notebook containing the exercise can be started or the data files can be viewed (data is in csv format) – refer to Figure 3.



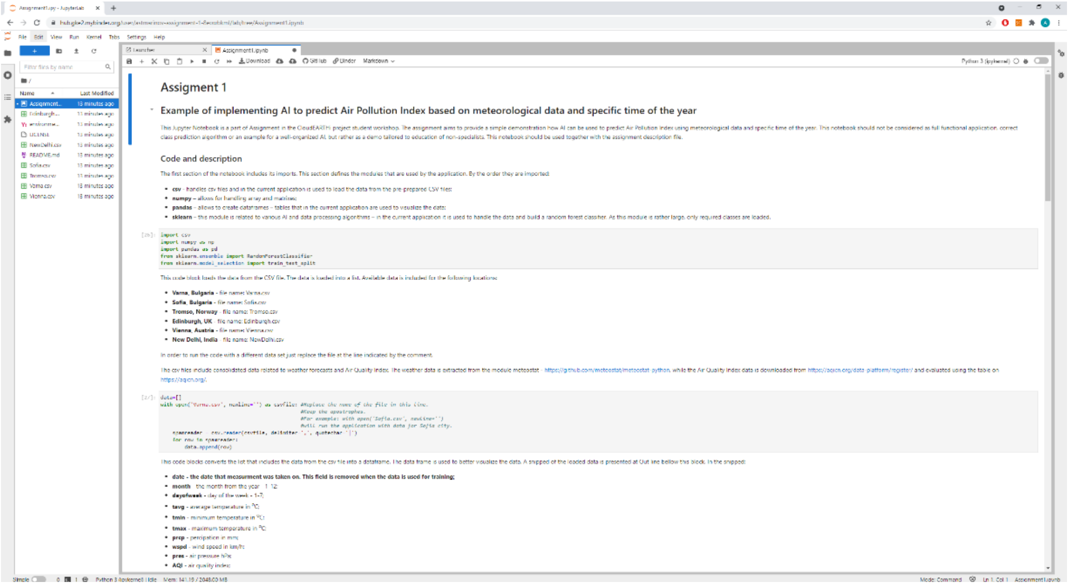
Name	Last Modified
Assignment1.ipynb	19 minutes ago
Edinburgh.csv	19 minutes ago
environment.yml	19 minutes ago
LICENSE	19 minutes ago
NewDelhi.csv	19 minutes ago
README.md	19 minutes ago
Sofia.csv	19 minutes ago
Tromso.csv	19 minutes ago
Varna.csv	19 minutes ago
Vienna.csv	19 minutes ago

Figure 3: Selecting to view data or run the notebook

If the notebook is started the view from Figure 4 should be present. The notebook can be executed either cell by cell or altogether. First time users should use the restart and run button (execution cell by cell can lead to errors if cells are executed out of order). Restart and run takes some time until the hole notebook is processed.

The notebook has 3 main fields:

- **Markdown** – those cells provide information on the notebook and should not be modified (Figure 5)
- **Program code cells** – those cells contain the program code of the application. In the ideal case they should be modified only based on the instructions in the markdown text (Figure 5).
- **Output cells** – those cells yield results from the execution of the program code cells (Figure 6).



Assigment 1

Example of implementing AI to predict Air Pollution Index based on meteorological data and specific time of the year

This Jupyter Notebook is a part of Assignment in the CloudEARTHi project student workshop. The assignment aims to provide a simple demonstration how AI can be used to predict Air Pollution Index using meteorological data and specific time of the year. This notebook should not be considered as full functional application, correct class prediction algorithm or an example for a well-organized AI, but rather as a demo tailored to education of non-specialists. This notebook should be used together with the assignment description file.

Code and description

The first section of the notebook includes its imports. This section defines the modules that are used by the application. By the order they are imported:

- **csv** - handles csv files and in the current application is used to load the data from the pre-prepared CSV files;
- **numpy** – allows for handling array and matrixes;
- **pandas** – allows to create dataframes – tables that in the current application are used to visualize the data;
- **sklearn** – this module is related to various AI and data processing algorithms – in the current application it is used to handle the data and build a random forest classifier. As this module is rather large, only required classes are loaded.

```
[26]: import csv
      import numpy as np
      import pandas as pd
      from sklearn.ensemble import RandomForestClassifier
      from sklearn.model_selection import train_test_split
```

This code block loads the data from the CSV file. The data is loaded into a list. Available data is included for the following locations:

- **Varna, Bulgaria** - file name: Varna.csv
- **Sofia, Bulgaria** - file name: Sofia.csv
- **Tromso, Norway** - file name: Tromso.csv
- **Edinburgh, UK** - file name: Edinburgh.csv
- **Vienna, Austria** - file name: Vienna.csv
- **New Delhi, India** - file name: NewDelhi.csv

In order to run the code with a different data set just replace the file at the line indicated by the comment.

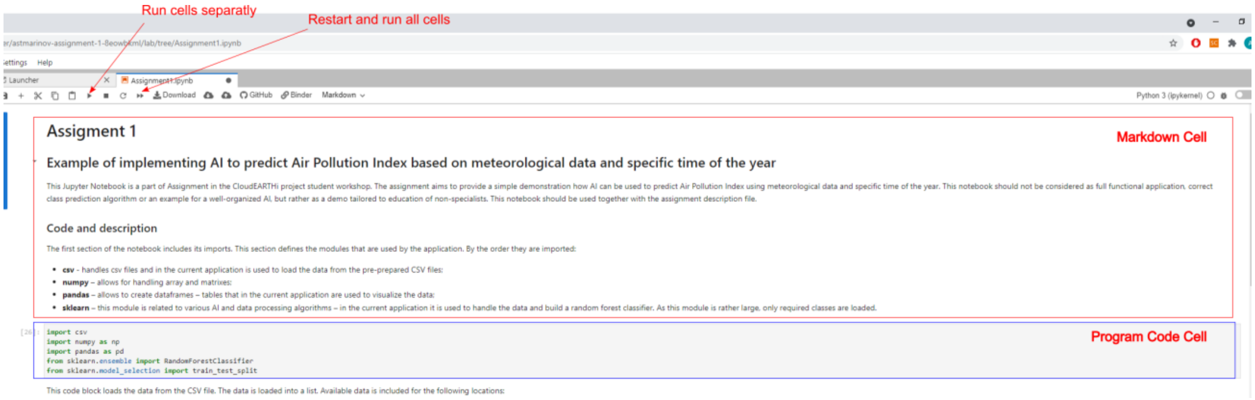
The csv files include consolidated data related to weather forecasts and Air Quality Index. The weather data is extracted from the module meteostat - https://github.com/meteostat/meteostat-python, while the Air Quality Index data is downloaded from https://aqicn.org/data-platform/register/ and evaluated using the table on https://aqicn.org/.

```
[27]: data=[]
      with open('Varna.csv', newline='') as csvfile: #Replace the name of the file in this line.
                                                     #Keep the apostrophes.
                                                     #For example: with open('Sofia.csv', newline='')
                                                     #will run the application with data for Sofia city.
          spamreader = csv.reader(csvfile, delimiter=',', quotechar='|')
          for row in spamreader:
              data.append(row)
```

This code blocks converts the list that includes the data from the csv file into a dataframe. The data frame is used to better visualize the data. A snipped of the loaded data is presented at Out line bellow this block. In the snipped:

- **date - the date that measurment was taken on. This field is removed when the data is used for training;**
- **month** - the month from the year - 1-12;
- **dayofweek** - day of the week - 1-7;
- **tavg** - average temperature in °C;
- **tmin** - minimum temperature in °C;
- **tmax** - maximum temperature in °C;
- **prcp** - percipation in mm;
- **wspd** - wind speed in km/h;
- **pres** - air pressure hPa;
- **AQI** - air quality index;

Figure 4: Notebook started


Run cells separatly

Restart and run all cells

Markdown Cell

Program Code Cell

Assigment 1

Example of implementing AI to predict Air Pollution Index based on meteorological data and specific time of the year

This Jupyter Notebook is a part of Assignment in the CloudEARTHi project student workshop. The assignment aims to provide a simple demonstration how AI can be used to predict Air Pollution Index using meteorological data and specific time of the year. This notebook should not be considered as full functional application, correct class prediction algorithm or an example for a well-organized AI, but rather as a demo tailored to education of non-specialists. This notebook should be used together with the assignment description file.

Code and description

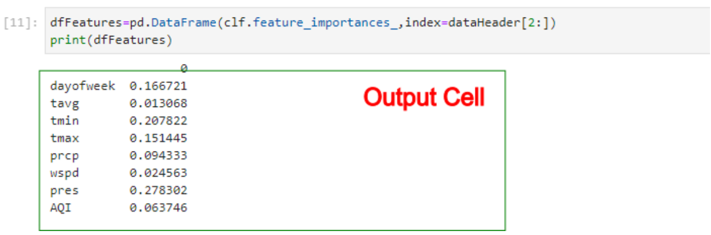
The first section of the notebook includes its imports. This section defines the modules that are used by the application. By the order they are imported:

- **csv** - handles csv files and in the current application is used to load the data from the pre-prepared CSV files;
- **numpy** – allows for handling array and matrixes;
- **pandas** – allows to create dataframes – tables that in the current application are used to visualize the data;
- **sklearn** – this module is related to various AI and data processing algorithms – in the current application it is used to handle the data and build a random forest classifier. As this module is rather large, only required classes are loaded.

```
[26]: import csv
      import numpy as np
      import pandas as pd
      from sklearn.ensemble import RandomForestClassifier
      from sklearn.model_selection import train_test_split
```

This code block loads the data from the CSV file. The data is loaded into a list. Available data is included for the following locations:

Figure 5: Notebook controls, markdown fields and program code cells



```
[11]: dfFeatures=pd.DataFrame(clf.feature_importances_,index=dataHeader[2:])
      print(dfFeatures)
```

```
                  0
dayofweek  0.166721
tavg       0.013068
tmin       0.207822
tmax       0.151445
prcp       0.094333
wspd       0.024563
pres       0.278302
AQI        0.063746
```

Figure 6: Output cells

3. Tasks

Task 1:

View the csv data files and the sources of the data and answer the following question: Is this application based on Big Data.

Answer:

Task 2:

Run the notebook for all the provided cities and record OOB accuracy and feature importance

Table 1. Classification results

Varna	**OOB accuracy**						
	Feature importance						
	dayofweek	tavg	tmin	tmax	prcp	wspd	pres
Sofia	**OOB accuracy**						
	Feature importance						
	dayofweek	tavg	tmin	tmax	prcp	wspd	pres
Edinburgh	**OOB accuracy**						
	Feature importance						
	dayofweek	tavg	tmin	tmax	prcp	wspd	pres
New Delhi	**OOB accuracy**						
	Feature importance						

	dayofweek	tavg	tmin	tmax	prcp	wspd	pres
Tromso	**OOB accuracy**						
	Feature importance						
	dayofweek	tavg	tmin	tmax	prcp	wspd	pres
Vienna	**OOB accuracy**						
	Feature importance						
	dayofweek	tavg	tmin	tmax	prcp	wspd	pres

Task 3:

Try to provide explanation for the difference in accuracy between the different cities (same algorithm, same settings). Hint: observe the data AQI.

Answer:

Task 4:

Try rationalize the feature importance. How the different features affect he air pollution.

For example: the month feature has high feature importance as the winter months are related to higher pollution due to emissions from heating installations…

Answer:

Task 5:

Provide your thoughts on the application. Do you see possibility to develop it further? What else should be included? Can you think of example for commercialization?

Answer:

4. Guidance

If you require guidance, consultation, suggestions or have difficulty completing the assignment please contact prof. Angel Marinov – Technical University of Varna, Bulgaria at a.marinov@tu-varna.bg

Videos, links and sources

Python:

https://www.youtube.com/watch?v=kqtD5dpn9C8&ab_channel=ProgrammingwithMosh

Jupyter Notebook:

https://jupyter.org/

https://www.youtube.com/watch?v=HW29067qVWk&ab_channel=CoreySchafer

Binder:

https://www.youtube.com/watch?v=owSGVOov9pQ&ab_channel=SerenaBonaretti

Air Quality Index data used in the assignment:

https://aqicn.org/data-platform/register/

Meteorological data used in the assignment:

https://github.com/meteostat/meteostat-python

Random Forest Algorithm:

https://www.youtube.com/watch?v=J4Wdy0Wc_xQ&t=5s&ab_channel=StatQuestwithJoshStarmer