AFRICAN LEADERSHIP UNIVERSITY RWANDA

AI ENVIRONMENT CREATION AND TESTING 23/1/2021

[Department: Computer Science]

[Facilitator: Kudakwashe Dandajena]

PEER MEMBERS

JOYCE NJERI	jmiiri17@alustudent.com
FATMATA ALICE KOROMA	f.koroma@alustudent.com
GILBERT SIBOMANA	g.sibomana@alustudent.com
AYO-FISHER OLUWAPAMILERIN	o.ayofishe@alustudent.com
CHRISTINE WASIKE	c.wasike@alustudent.com

TABLE OF CONTENTS

1.0 Introduction

1.1 Purpose

The document is meant to show how the group created a functional or running environment for implementing AI experiments. In this document it highlights the different features the group tested for in their AI environment.

1.2 Overall Objective

The AI implementation environment is to be created on a single node with the required libraries and tools that will simulate an on-premise hardware environment which will be our machine. This document will show how the group created the AI environment and their testing for the environment with a specific data set from an African context.

1.3 Document Conventions

This document follows Latex format.

1.4 Intended Audience And Reading Suggestions

This document is worth reading by an audience that is interested in the groups' AI environment and would like to know how they created and tested the environment.

2.0 Project Overview

In this step-by-step guide we will:

1. Download and install all the required libraries and get the most useful packages for machine learning in Python.
2. Load a dataset from an African context and understand its structure using statistical summaries and data visualization.
3. Implement simple data preprocessing into a form suitable for training.

The best way to come to terms with a new platform or tool is by working through a machine learning project, end-to-end, and covering the key steps from loading data, summarizing data, evaluating algorithms and making some predictions. For this project, we are going to create our AI environment and cover the precluding steps to completing a full Machine Learning project.

Below is an overview of what we are going to cover:

- Installing the required tools and libraries.
- Choosing a dataset
- Summarizing the dataset
- Visualizing the dataset
- Data Preprocessing

Our AI environment should be running specifically on the Linux platform, so this is where we started with creating our AI environment. Some of us used virtual machines to be able to have Linux on our PCs, others used dual boot, and others were already using Linux on their PC.

How to Create a Linux Virtual Machine For Machine Learning With Python 3

3.0 Downloading, Installing and Starting AI Environment

3.1 Latest Anaconda installation

Here are steps we followed to install Anaconda on Windows Ubuntu Terminal: the majority Linux platform being used by group members.

1. We installed all necessary packages (eg *libgl1-mesa-glx, etc…*)\
2. Downloaded Anaconda installation script
3. Verified data integrity of data script
4. Run the script we downloaded($ *bash script.sh)*
5. Updated path to be
6. And run it with ***anaconda-navigator*** command in terminal

How to Setup Anaconda on Ubuntu Terminal

3.2 Latest version of Python

To be able to use Jupyter notebook and other libraries with anaconda we also needed to have latest python version so we installed it through these steps:

1. Updated our packages (*sudo apt update*)
2. Added ppa to our system list
3. Installed python to our system(*sudo apt install python3.8*)

How to Setup a Python Environment for Machine Learning with Anaconda

3.3 Jupyter Notebook

Most of the time when you install Anaconda well the Jupyter Notebook is already installed, but when it's not installed you just have to open your Anaconda(*anaconda-navigator*) through the terminal and scroll to Jupyter

Notebook then click install. From there you can open it by clicking launch. After opening it you can create a new notebook by choosing Python 3.

How To Set Up Jupyter Notebook with Python 3 on Ubuntu

3.4 TensorFlow and Keras

Tensorflow is an open source library that was developed by Google, it is used to develop and train models using python and other languages. The library was developed for deep learning application and machine learning.

Keras is a neural network library that was developed in python. It works on developing and evaluating deep learning model, and it helps in defining and training neural network model

Here is how to install Tensorflow and Keras;

1. Check if you have the latest version of pip
2. Install Tensorflow(*pip install tensorflow*)

NB: when you install **tensorflow** library it comes with other libraries including **Keras**

Install TensorFlow on Ubuntu

3.5 NumPy

NumPy is a numeric python library, it contains mult-dimension array and matrix data structure, it can be used to perform different mathematical operations on arrays.

If you have installed your anaconda well, the NumPy will be there without needing to install. In case the numpy is not installed you can still install it through below steps:

1. Make sure you have python installed
2. And have latest pip version
3. Then install NumPy(*sudo apt install python-numpy*)

Install Numpy on Ubuntu

3.6 SciPy

Scipy is a scientific python library that works with NumPy to run. It helps to perform mathematical operations.

Installation steps

1. Update packages(*sudo apt-get update -y*)
2. Install SciPy (*sudo apt-get update -y python-scipy*)

Install SciPy on Ubuntu

3.7 Matplotlib

Is a plotting python library, this library helps to create statics and animated visualizations it works more closely with NumPy.
When you have installed Anaconda well on your system, the matplotlib comes with it. In case it is not there just follow below steps

Installation steps

1. Update package
2. Install Matplotlib

Install Matplotlib on Ubuntu

3.8 Pandas

Pandas library was written for python programming and it helps with data manipulation, and data analysis

When you have installed Anaconda well on your system, the pandas comes with it. In case it is not there just follow below steps

Installation steps

1. Update package
2. Install Pandas

Install Pandas on Ubuntu

3.9 Scikit-Learn

Scikit-Learn is a machine learning library that was developed for python, it works with Numpy, Pandas and Matplotlib.

Installation steps:

1. Update package
2. Install scikit-Learn(*pip install -U scikit-learn*)

Install Scikit-Learn on Ubuntu

4.0 Dataset Selection

Due to the current humanitarian crisis we are in, we decided to explore the increase in covid deaths across the African continent to analyze its impact, and attempt to predict future trends that may impact online learning, long-term, at ALU, and around the world.

The dataset we chose was downloaded in the form of an excel file from 'The Humanitarian Data Exchange' platform. This is a quickly accessible and reliable data source as it is the centre for humanitarian data, aggregating up-to-date data on humanitarian issues around the world.

4.1 Importing libraries

We imported all of the modules, functions and objects we are going to use. Everything should load without error. Incase of an error, we had to stop because we needed a working AI environment before continuing. If this happens, refer to the above notes and links about setting up your AI environment.

5.0 Testing AI Environment

5.1 Summarizing the dataset

In this step we understood the data a few different ways using Pandas Dataframe. Pandas, in particular, offers data structures and operations for manipulating numerical tables. Some of its useful properties such as shape, describe, and dtypes were used in understanding our dataset while simultaneously validating testing of our AI environment as discussed below:

5.1.1 Dimensions of the dataset.

Using the shape property, we were able to get a quick idea of how many instances (rows) and how many attributes (columns) the data contains. Here we saw 52 instances and 316 attributes.

5.1.2 Peek at the data itself.

Using the head() property, we were able to eyeball the first 15 rows of our dataset.

5.1.3 Statistical summary of all attributes.

To further understand our data's summary of each attribute, we investigated statistical figures such as the count, mean, the min and max values as well as some percentiles.

The statistical functions' scipy.stats module particularly contains a growing library of statistical functions including mode, max and min values, mean, variance, skewness, and kurtosis.

5.2 Visualizing the dataset

Now that we had a basic idea about the data, we extended that with some visualizations. For these, we used matplotlib to visualize two types of plots:

5.2.1 Histograms - to get an idea of the distribution.

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. Therefore, we also tested the numpy library that adds support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

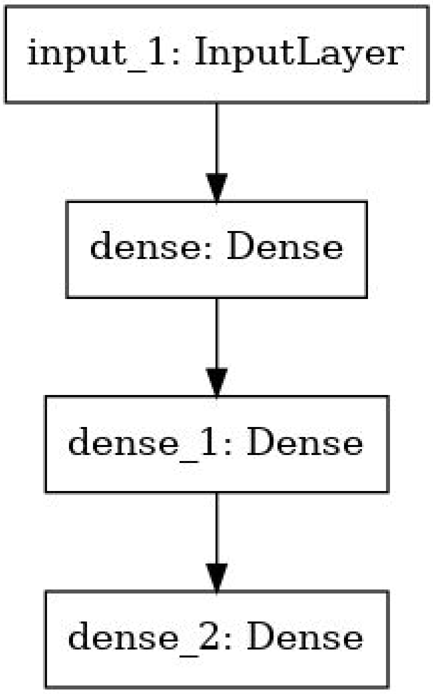
5.2.2 Bar charts to better understand each attribute.

Here, we used arange(), an inbuilt numpy function that returns an ndarray object containing evenly spaced values, to define intervals for plotting our Bar chart.

5.2.3 The Functional API

Before we dive into data preprocessing, let's visit an interesting way of building graphs of layers in AI using Keras functional API. Keras, in itself, is a neural network library that provides only high-level APIs. The main idea is that a deep learning model is usually a directed acyclic graph (DAG) of layers. So the functional API is a way to build graphs of layers.

We used Keras in a very simple artificial neural network simulation using our dataframe's shape property to provide a user-friendly API which makes it easy to quickly prototype deep learning models. Below is our very simple output that validates testing of the Keras library:



5.3 Data Preprocessing

Preprocessing data is one of the few important steps that need to be taken before starting to train models. Here, we also use numpy's np.array to create an array from our dataset. There were two ways we performed data preprocessing on our data: preprocessing with tensorflow and with sklearn.

5.3.1 Preprocessing with tensorflow

Assuming the nominal task for our dataset is to predict covid deaths from the other measurements, our objective was a simple way to train a model using our excel data. We first separated the features and labels from our dataset for training. We then made a regression model, and since there was only a single input tensor, a keras sequential model was sufficient here. To train that model, we passed our features and labels to Model.fit.

5.3.2 Preprocessing with sklearn

There are various steps when preprocessing data with sklearn. For this project, we covered the first step which is checking for missing values. Handling missing values is an essential preprocessing task that can drastically deteriorate your model when not done with sufficient care.

MissingIndicator from sklearn.impute came in handy in this step and for filling up missing values with common strategies such as mean, most_frequent, median and constant, sklearn provides a SimpleImputer.

6.0 Conclusion

With the data set of our choice with an African context, the 9 libraries listed have been used to properly manipulate the data, perform operation and procedures as a way of demonstrating the functionality of the created environment. The group achieved the following results:

1. Created the AI environment
2. Used the data set to test the different functionalities of the environment
3. Manipulated the data, performed operations and procedures with the data.

Appendix A: Glossary

Term	**Definition**
AI	Artificial Intelligence
PPA	Predictive and Prescriptive Analytics
Anaconda	Anaconda is a distribution of the Python and R programming languages for scientific computing, that aims to simplify package management and deployment.
Python	Python is an interpreted, high-level and general-purpose programming language.
Jupyter Notebook	Project Jupyter is a nonprofit organization created to "develop open-source software, open-standards, and services for interactive computing across dozens of programming languages"
TensorFlow	TensorFlow is a free and open-source software library for machine learning.
Keras	Keras is an open-source software library that provides a Python interface for artificial neural networks
Numpy (Numerical Python)	NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
Scipy (Scientific Python)	SciPy is a free and open-source Python library used for scientific computing and technical computing.
Matplotlib	Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy
Pandas	pandas is a software library written for the Python programming language for data manipulation and analysis.
Scikit-Learn	Scikit-learn is a free software machine learning library for the Python programming language

Appendix B: Reference

a. Anaconda. 2021. Anaconda | Individual Edition. [online] Available at: <https://www.anaconda.com/products/individual> [Accessed 23 January 2021].

b. Python.org. 2021. *Welcome To Python.Org*. [online] Available at: <https://www.python.org/> [Accessed 23 January 2021].

c. Jupyter.org. 2021. *Project Jupyter*. [online] Available at: <https://jupyter.org/> [Accessed 23 January 2021].

d. TensorFlow. 2021. *Tensorflow*. [online] Available at: <https://www.tensorflow.org/> [Accessed 23 January 2021].

e. Team, K., 2021. *Keras: The Python Deep Learning API*. [online] Keras.io. Available at: <https://keras.io/> [Accessed 23 January 2021].

f. Numpy.org. 2021. *Numpy*. [online] Available at: <https://numpy.org/> [Accessed 23 January 2021].

g. Scipy.org. 2021. *Scipy.Org — Scipy.Org*. [online] Available at: <https://www.scipy.org/> [Accessed 23 January 2021].

h. Matplotlib.org. 2021. *Matplotlib: Python Plotting — Matplotlib 3.3.3 Documentation*. [online] Available at: <https://matplotlib.org/> [Accessed 23 January 2021].

i. Pandas.pydata.org. 2021. *Pandas - Python Data Analysis Library*. [online] Available at: <https://pandas.pydata.org/> [Accessed 23 January 2021].

j. Scikit-learn.org. 2021. *Scikit-Learn: Machine Learning In Python — Scikit-Learn 0.16.1 Documentation*. [online] Available at: <https://scikit-learn.org/> [Accessed 23 January 2021].