Configuration Manual

MSc Research Project
Masters in Artificial Intelligence

Hudson Paul Rajesh
Student ID: X22181920@student.ncirl.ie

School of Computing
National College of Ireland

Supervisor: Dr. Muslim Jameel Syed

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Hudson Paul Rajesh
Student ID:	X22181920
Programme:	Masters in Artificial Intelligence **Year:** 2023-2024
Module:	MSc Research Project
Lecturer:	Muslim Jameel Syed
Submission Due Date:	14th December 2023
Project Title:	Cybersecurity Fortification through MachineLearning: Predictive Models for Malware Detection in Network Environments
Word Count:	**890** **Page Count: 9**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Hudson Paul Rajesh

Date: 14th December 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Hudson Paul Rajesh
Student ID: x22181920

1 Introduction

Welcome to the Cybersecurity Fortification through Machine Learning project's configuration manual. This ground-breaking initiative aims to improve network environments' malware detection capabilities. This extensive manual is intended to assist users in setting up, configuring, and using the predictive models created to strengthen cybersecurity defences.

2 System Specification

- Supported, Metal GPUFamily Apple 7
- RAM; 16 GB
- SSD; 1TB + 500 Gb
- System Type: 64-bit Operating Systems
- Operating System: Apple MacBook Pro

3 Section 3

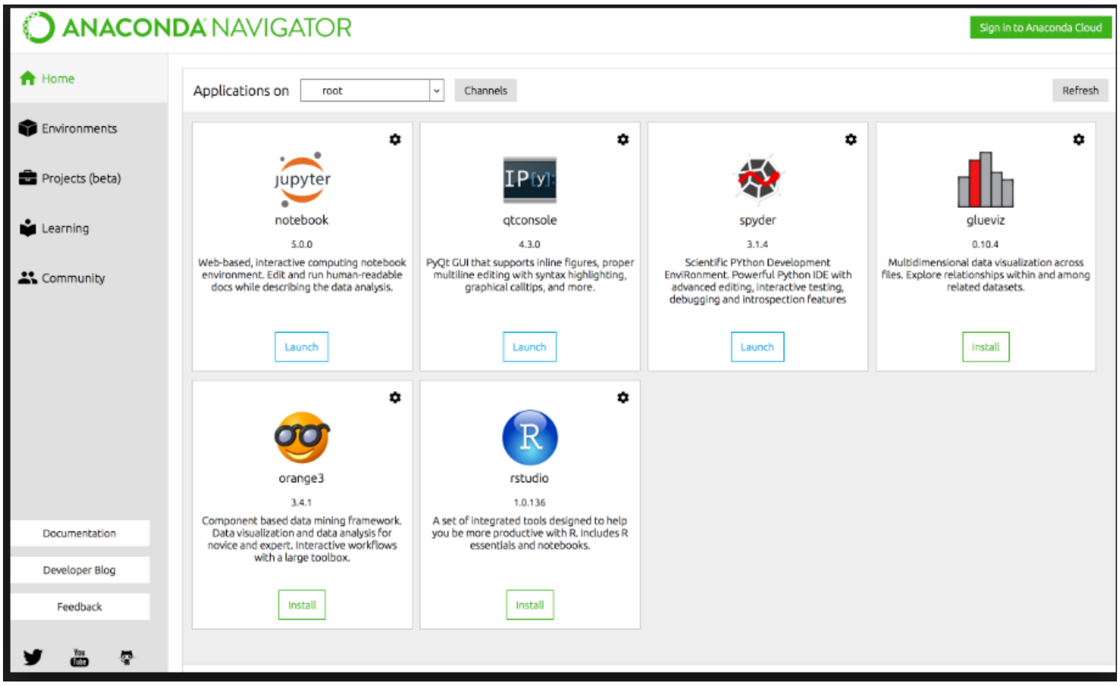
The Anaconda prompt and the Python programming language form the basis for putting this idea into practice. To guarantee accurate and neat results, the project has been integrated with the following libraries and packages:.

- NumPy
- Pandas
- Matplotlib
- Seaborn
- LightGBM
- Scikit-learn (KFold)
- Plotly
- Keras
- TensorFlow (Adam optimizer)

4 Steps for Configuration of Machine Learning:

1. To download and install Anaconda3 (Anconda, 2023)
2. extract the 'ml_env.rar' folder and paste it to the anaconda3's envs folder
3. Extract the 'HIDS_Fullcode 1' folder
4. Run the Anaconda Navigator

5. Open in the anaconda3 prompt and in prompt used the command cd/d to change the directory to HIDS Fullcode 1.
6. (cd) Navigate to ‘HIDS_Fullcode 1’ folder.
7. Now run command: conda activate ml_env
8. Run command: python output_nsl_kdd.py
9. Run command : python output edge_iiot.py



jupyter notebook 5.0.0
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
Launch

qtconsole 4.3.0
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
Launch

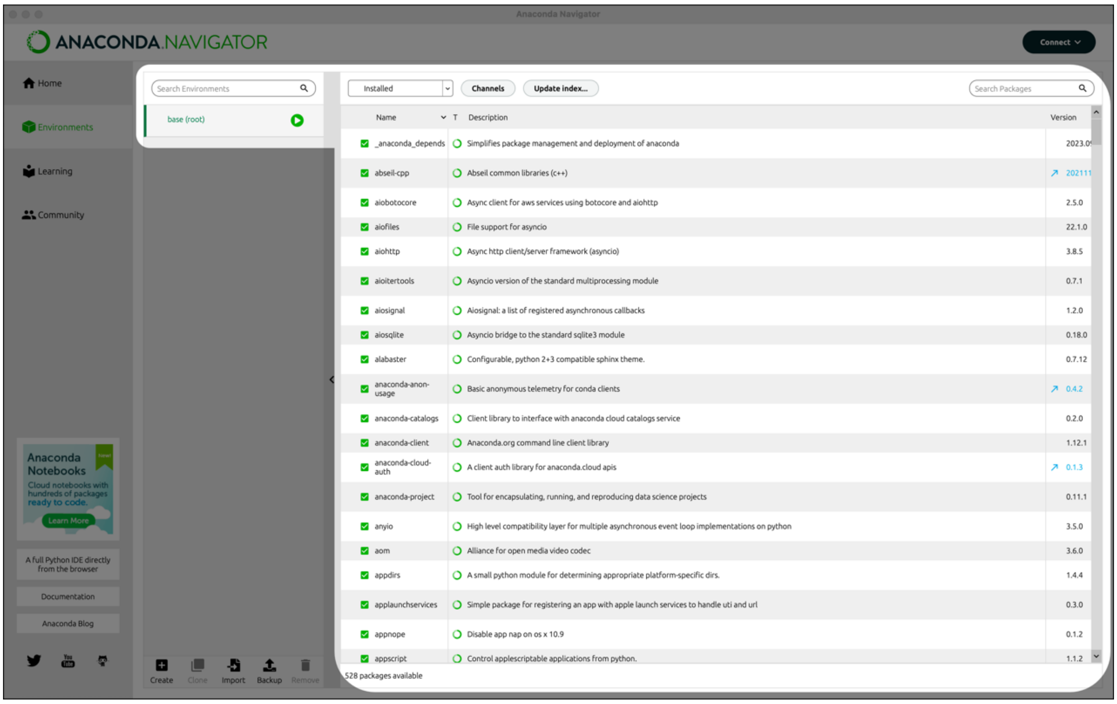
spyder 3.1.4
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features
Launch

glueviz 0.10.4
Multidimensional data visualization across files. Explore relationships within and among related datasets.
Install

orange3 3.4.1
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.
Install

rstudio 1.0.136
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.
Install

Figure 1: Anaconda Navigator



Name	T	Description	Version
☑ _anaconda_depends	○	Simplifies package management and deployment of anaconda	2023.0
☑ abseil-cpp	○	Abseil common libraries (c++)	↗ 202111
☑ aiobotocore	○	Async client for aws services using botocore and aiohttp	2.5.0
☑ aiofiles	○	File support for asyncio	22.1.0
☑ aiohttp	○	Async http client/server framework (asyncio)	3.8.5
☑ aioitertools	○	Asyncio version of the standard multiprocessing module	0.7.1
☑ aiosignal	○	Aiosignal: a list of registered asynchronous callbacks	1.2.0
☑ aiosqlite	○	Asyncio bridge to the standard sqlite3 module	0.18.0
☑ alabaster	○	Configurable, python 2+3 compatible sphinx theme.	0.7.12
☑ anaconda-anon-usage	○	Basic anonymous telemetry for conda clients	↗ 0.4.2
☑ anaconda-catalogs	○	Client library to interface with anaconda cloud catalogs service	0.2.0
☑ anaconda-client	○	Anaconda.org command line client library	1.12.1
☑ anaconda-cloud-auth	○	A client auth library for anaconda.cloud apis	↗ 0.1.3
☑ anaconda-project	○	Tool for encapsulating, running, and reproducing data science projects	0.11.1
☑ anyio	○	High level compatibility layer for multiple asynchronous event loop implementations on python	3.5.0
☑ aom	○	Alliance for open media video codec	3.6.0
☑ appdirs	○	A small python module for determining appropriate platform-specific dirs.	1.4.4
☑ applaunchservices	○	Simple package for registering an app with apple launch services to handle uti and url	0.3.0
☑ appnope	○	Disable app nap on os x 10.9	0.1.2
☑ appscript	○	Control applescriptable applications from python.	1.1.2

528 packages available

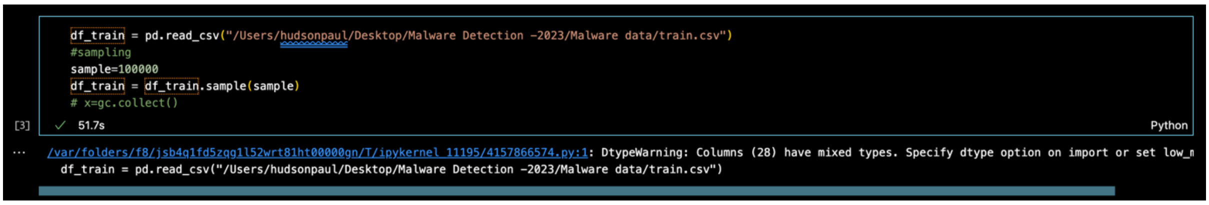
Figure 2: ml_env.rar’

5 Steps for Configuration of Visual Studio Code

1. Download and Install Visual Studio Code:
2. Visit the official Visual Studio Code website (https://code.visualstudio.com/) and download the installer for your operating system.
3. Run the installer and follow the on-screen instructions to install Visual Studio Code on your machine.
4. Open Visual Studio Code:
5. Once the installation is complete, open Visual Studio Code.
6. Extensions and Plugins: Explore and install extensions and plugins based on your requirements. Common extensions include Python, Git, and various language support extensions.
7. Theme and Color Scheme: Choose a theme and color scheme that suits your preference. You can customize the appearance of Visual Studio Code through the "Preferences" menu.
8. Configure Settings: Adjust settings according to your preferences. You can access settings through the gear icon on the bottom left corner and selecting "Settings."
9. Integrated Terminal: Visual Studio Code comes with an integrated terminal. Familiarize yourself with its features, and customize the shell and appearance as needed.
10. Version Control Integration: If you're using version control (e.g., Git), integrate it with Visual Studio Code. Initialize a Git repository in your project folder and connect it to Visual Studio Code.
11. Workspace Setup: Set up your workspace by opening the desired project folder. You can customize your workspace settings to include specific folders and files.
12. Debugger Configuration: If you're working with a programming language that requires debugging, configure the debugger settings. Visual Studio Code supports various debugging configurations.
13. Explore Features: Take the time to explore additional features such as IntelliSense, code navigation, and integrated terminal functionalities.
14. Stay Updated: Regularly check for updates and new extensions to keep Visual Studio Code up to date with the latest features and improvements.

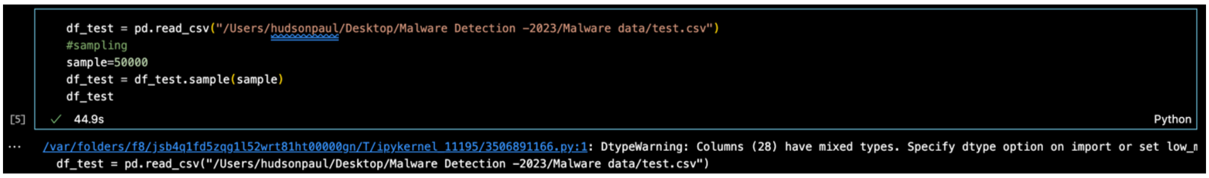
6 Procedure for Machine Learning

6.1 Pre-Processing the data



```
df_train = pd.read_csv("/Users/hudsonpaul/Desktop/Malware Detection -2023/Malware data/train.csv")
#sampling
sample=100000
df_train = df_train.sample(sample)
# x=gc.collect()
```

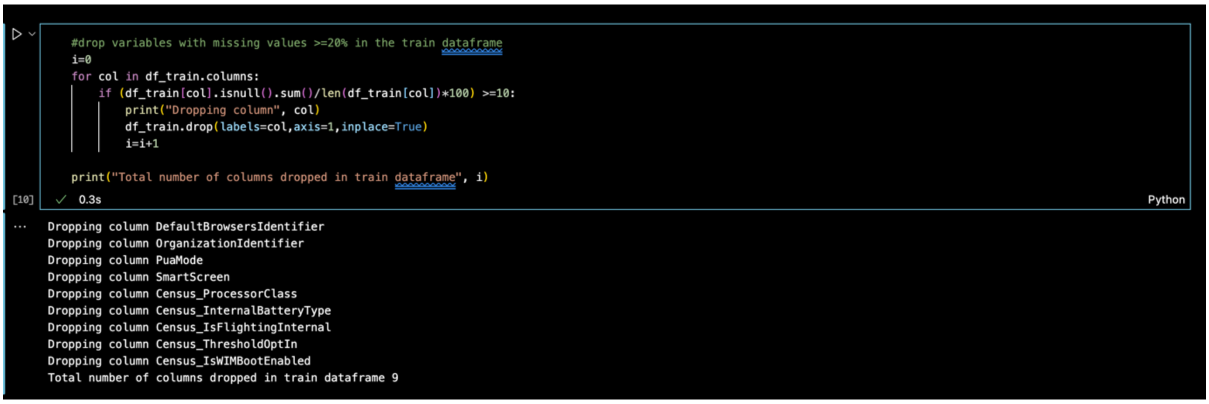
```
[3]  ✓ 51.7s                                                                                     Python
...  /var/folders/f8/jsb4q1fd5zqg1l52wrt81ht00000gn/T/ipykernel_11195/4157866574.py:1: DtypeWarning: Columns (28) have mixed types. Specify dtype option on import or set low_m
       df_train = pd.read_csv("/Users/hudsonpaul/Desktop/Malware Detection -2023/Malware data/train.csv")
```



```
df_test = pd.read_csv("/Users/hudsonpaul/Desktop/Malware Detection -2023/Malware data/test.csv")
#sampling
sample=50000
df_test = df_test.sample(sample)
df_test
```

```
[5]  ✓ 44.9s                                                                                     Python
...  /var/folders/f8/jsb4q1fd5zqg1l52wrt81ht00000gn/T/ipykernel_11195/3506891166.py:1: DtypeWarning: Columns (28) have mixed types. Specify dtype option on import or set low_m
       df_test = pd.read_csv("/Users/hudsonpaul/Desktop/Malware Detection -2023/Malware data/test.csv")
```

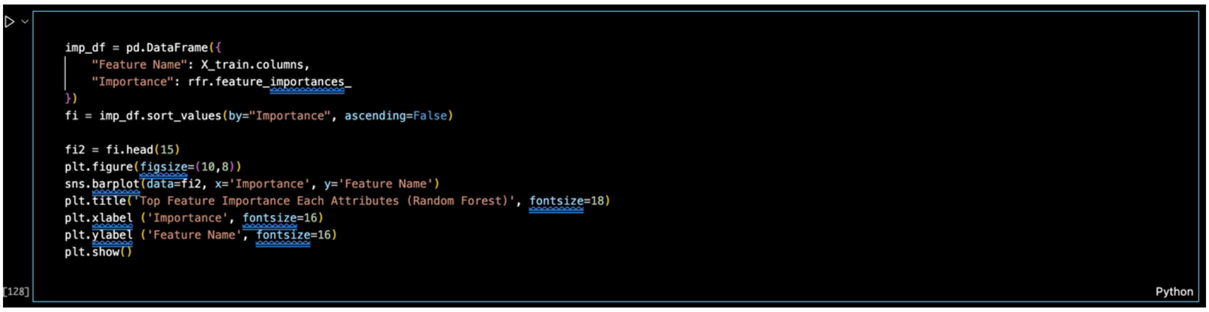
- ❖ Loading the Microsoft Malware dataset and removing the unwanted columns, duplicate rows, null values and removing other attack categories.
- ❖ Loading the Microsoft Malware dataset setting up the column plus data and creating a new csv file.



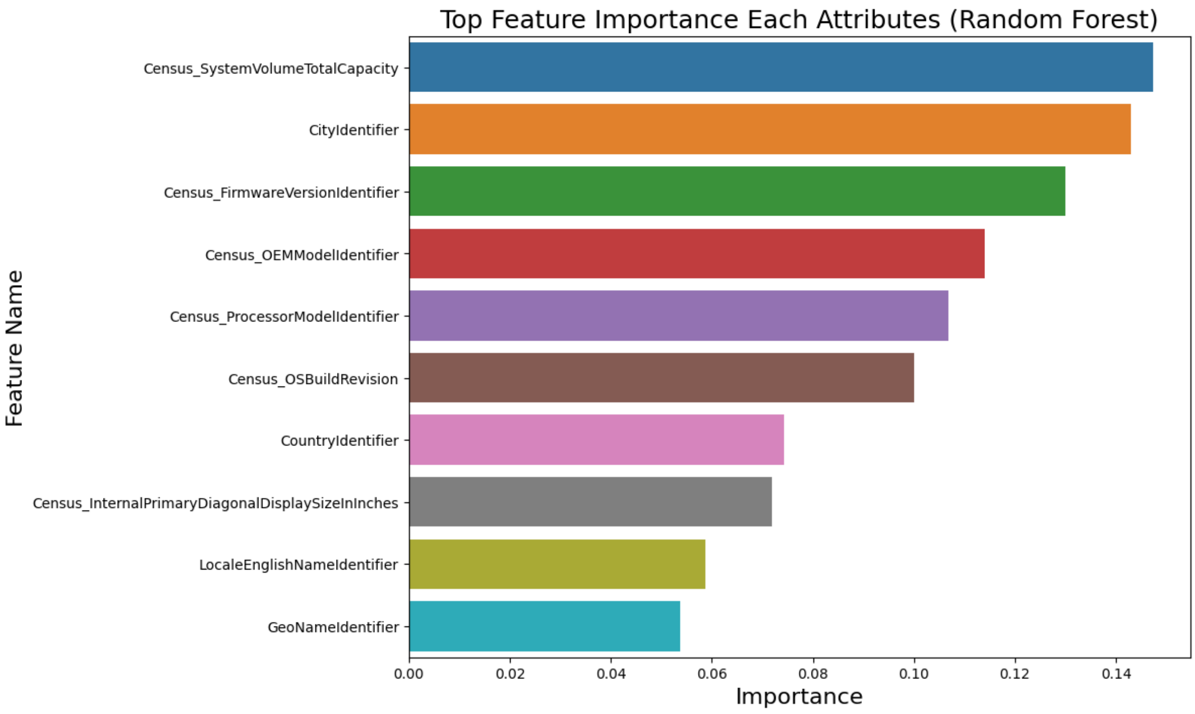
```
#drop variables with missing values >=20% in the train dataframe
i=0
for col in df_train.columns:
    if (df_train[col].isnull().sum()/len(df_train[col])*100) >=10:
        print("Dropping column", col)
        df_train.drop(labels=col,axis=1,inplace=True)
        i=i+1

print("Total number of columns dropped in train dataframe", i)
```

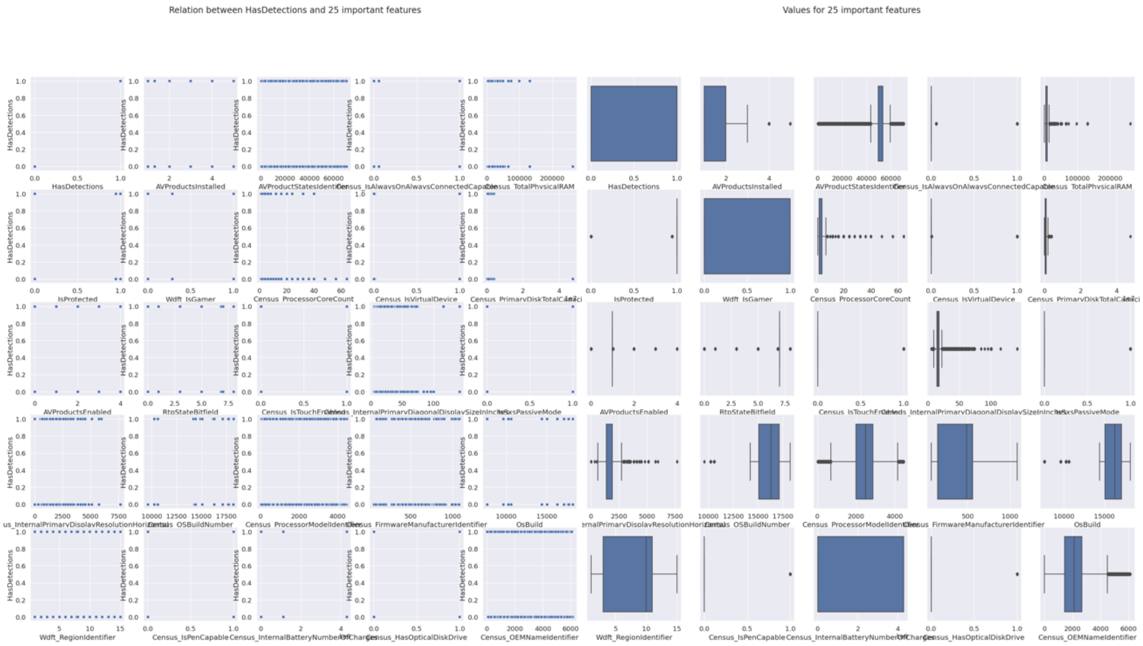
```
Dropping column DefaultBrowsersIdentifier
Dropping column OrganizationIdentifier
Dropping column PuaMode
Dropping column SmartScreen
Dropping column Census_ProcessorClass
Dropping column Census_InternalBatteryType
Dropping column Census_IsFlightingInternal
Dropping column Census_ThresholdOptIn
Dropping column Census_IsWIMBootEnabled
Total number of columns dropped in train dataframe 9
```

6.2 Feature selection

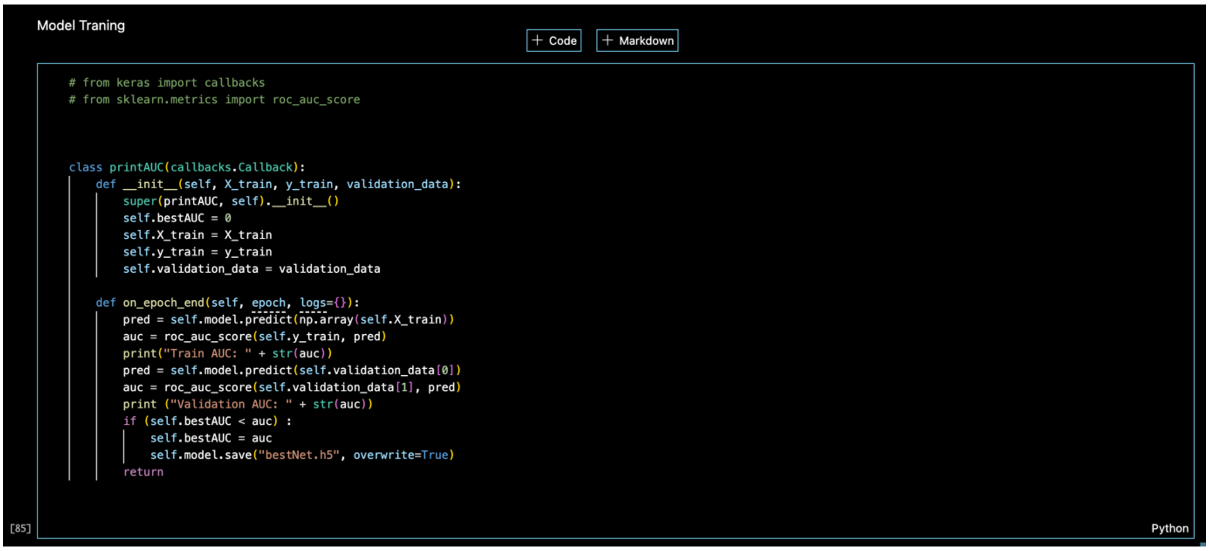


```
imp_df = pd.DataFrame({
    "Feature Name": X_train.columns,
    "Importance": rfr.feature_importances_
})
fi = imp_df.sort_values(by="Importance", ascending=False)

fi2 = fi.head(15)
plt.figure(figsize=(10,8))
sns.barplot(data=fi2, x='Importance', y='Feature Name')
plt.title('Top Feature Importance Each Attributes (Random Forest)', fontsize=18)
plt.xlabel ('Importance', fontsize=16)
plt.ylabel ('Feature Name', fontsize=16)
plt.show()
```



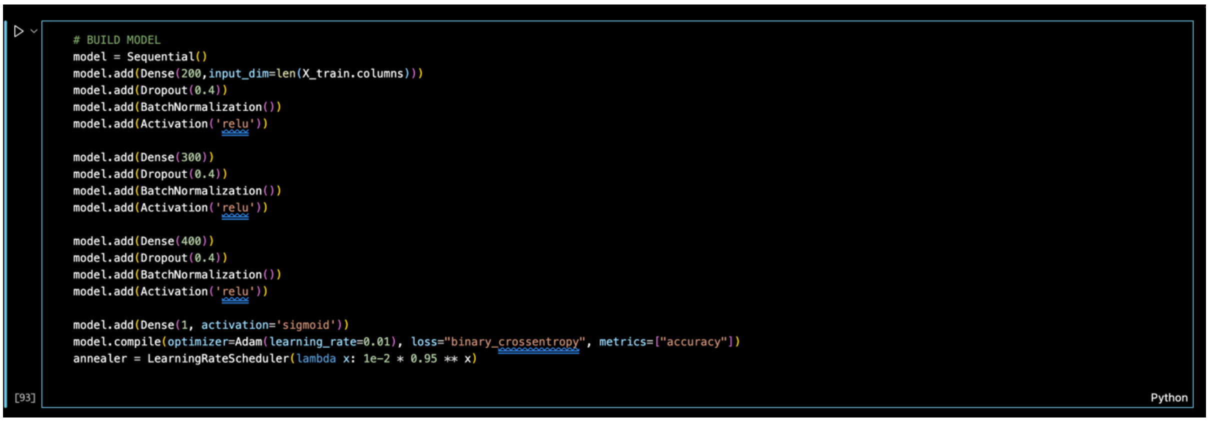


6.3 Training and Testing of models

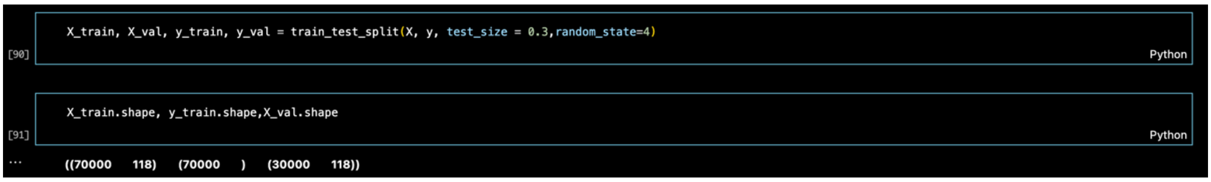


Model Traning

```
# from keras import callbacks
# from sklearn.metrics import roc_auc_score


class printAUC(callbacks.Callback):
    def __init__(self, X_train, y_train, validation_data):
        super(printAUC, self).__init__()
        self.bestAUC = 0
        self.X_train = X_train
        self.y_train = y_train
        self.validation_data = validation_data

    def on_epoch_end(self, epoch, logs={}):
        pred = self.model.predict(np.array(self.X_train))
        auc = roc_auc_score(self.y_train, pred)
        print("Train AUC: " + str(auc))
        pred = self.model.predict(self.validation_data[0])
        auc = roc_auc_score(self.validation_data[1], pred)
        print ("Validation AUC: " + str(auc))
        if (self.bestAUC < auc) :
            self.bestAUC = auc
            self.model.save("bestNet.h5", overwrite=True)
        return
```

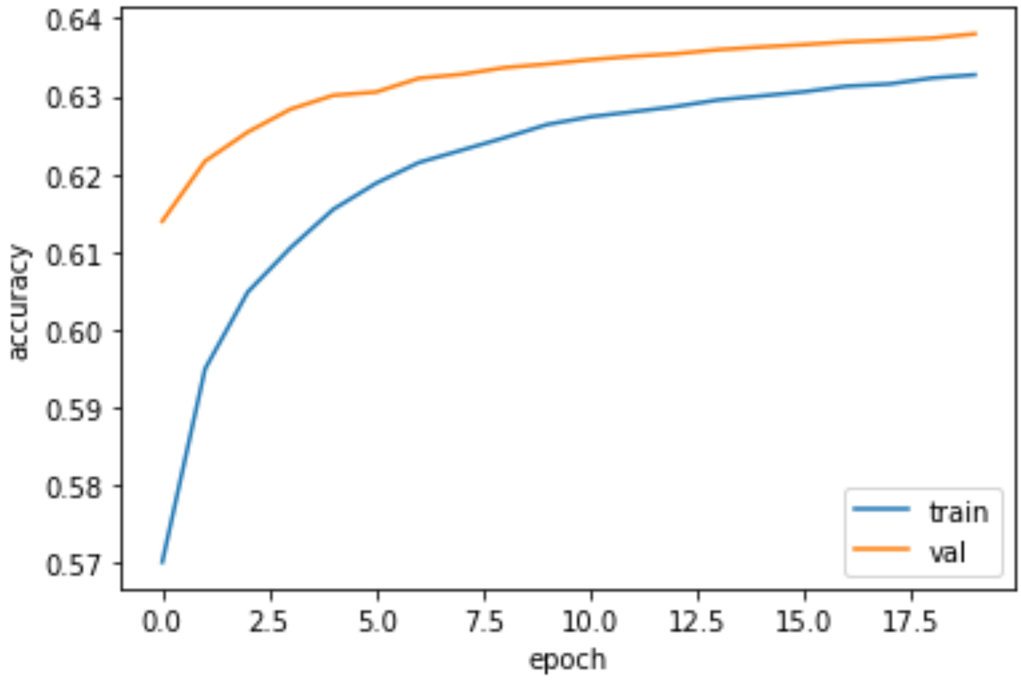


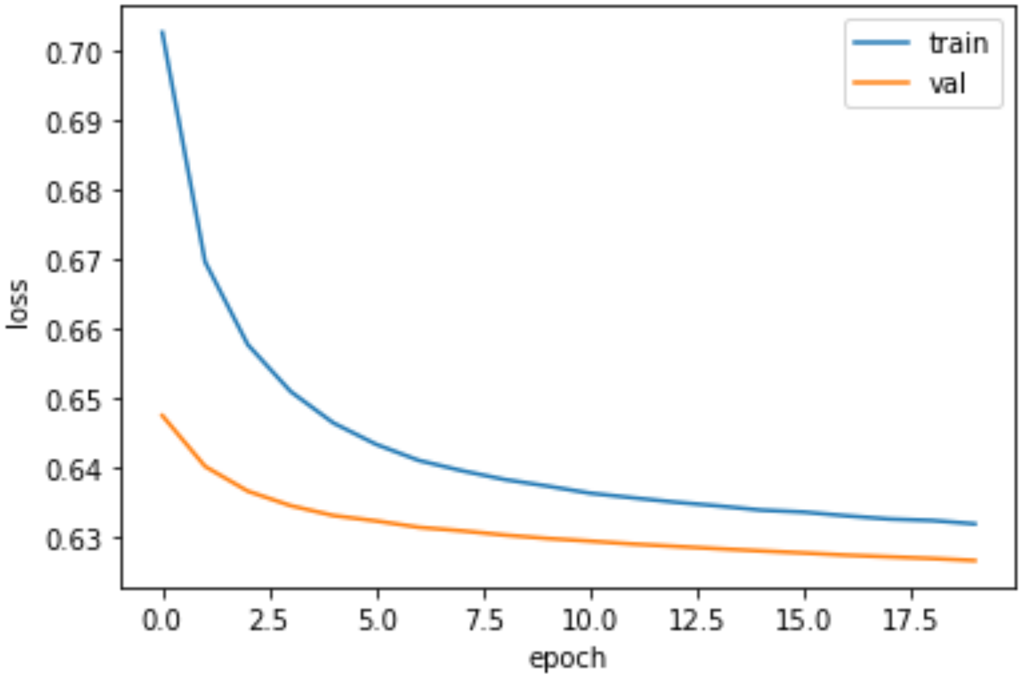
```
# BUILD MODEL
model = Sequential()
model.add(Dense(200,input_dim=len(X_train.columns)))
model.add(Dropout(0.4))
model.add(BatchNormalization())
model.add(Activation('relu'))

model.add(Dense(300))
model.add(Dropout(0.4))
model.add(BatchNormalization())
model.add(Activation('relu'))

model.add(Dense(400))
model.add(Dropout(0.4))
model.add(BatchNormalization())
model.add(Activation('relu'))

model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer=Adam(learning_rate=0.01), loss="binary_crossentropy", metrics=["accuracy"])
annealer = LearningRateScheduler(lambda x: 1e-2 * 0.95 ** x)
```



```
[90]    X_train, X_val, y_train, y_val = train_test_split(X, y, test_size = 0.3,random_state=4)
                                                                                        Python

[91]    X_train.shape, y_train.shape,X_val.shape
                                                                                        Python
...     ((70000    118)    (70000    )    (30000    118))
```

6.4 Results





References

Anaconda, 2023. *Anaconda Navigator.* [Online]
Available at: https://www.anaconda.com/download
Studio, V., 2023. *Visual Studio Code.* [Online]
Available at: https://code.visualstudio.com/download