Intrusion Detection with Machine Learning

Bryan Solis

Data 606

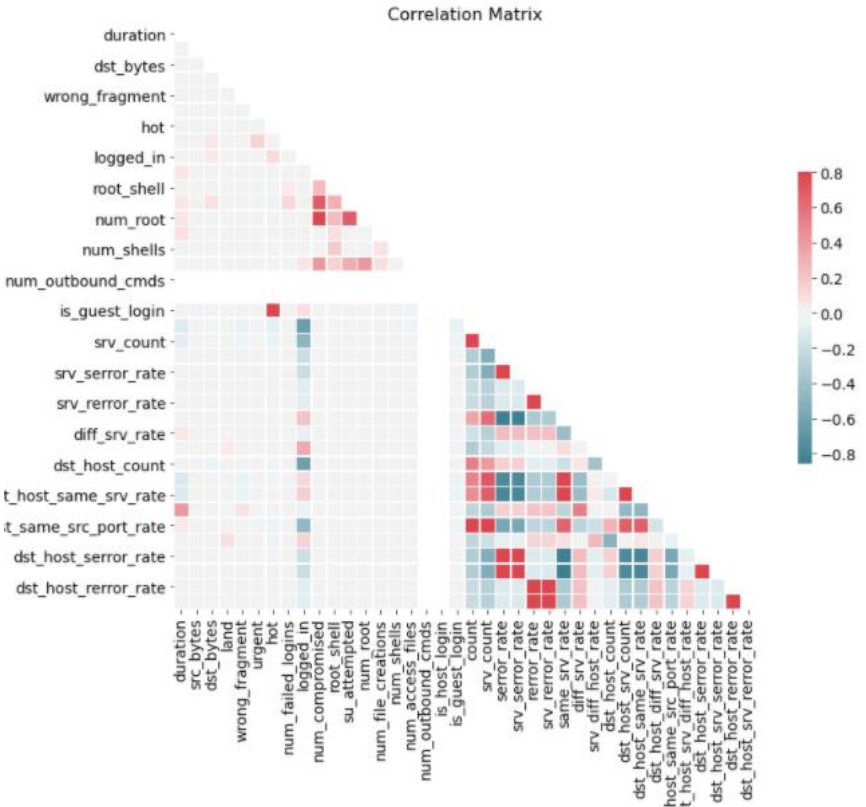
RECAP

- Intrusion Detection System using KDD-CUP 99 Dataset
- Neural Network that can work with the dataset
- Dataset is 75MB CSV File
- Dataset contains 41 unique features
- No nulls or missing values

PHASE II:DATA EXPLORATION & MODEL CONSTRUCTION

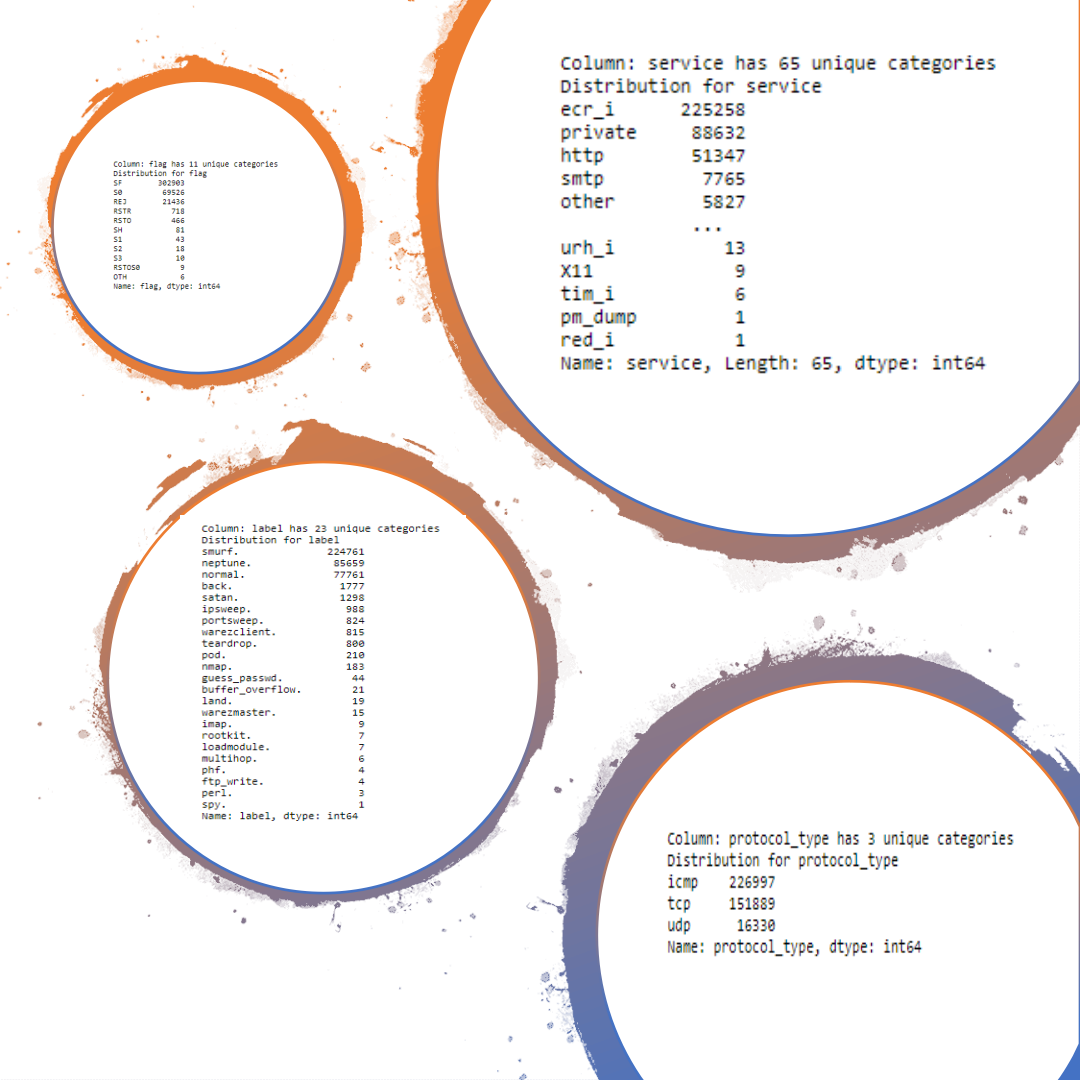
Creation of a Correlation Matrix

- Correlation of one column to another column in our dataset
- Dataset do not have a lot correlation

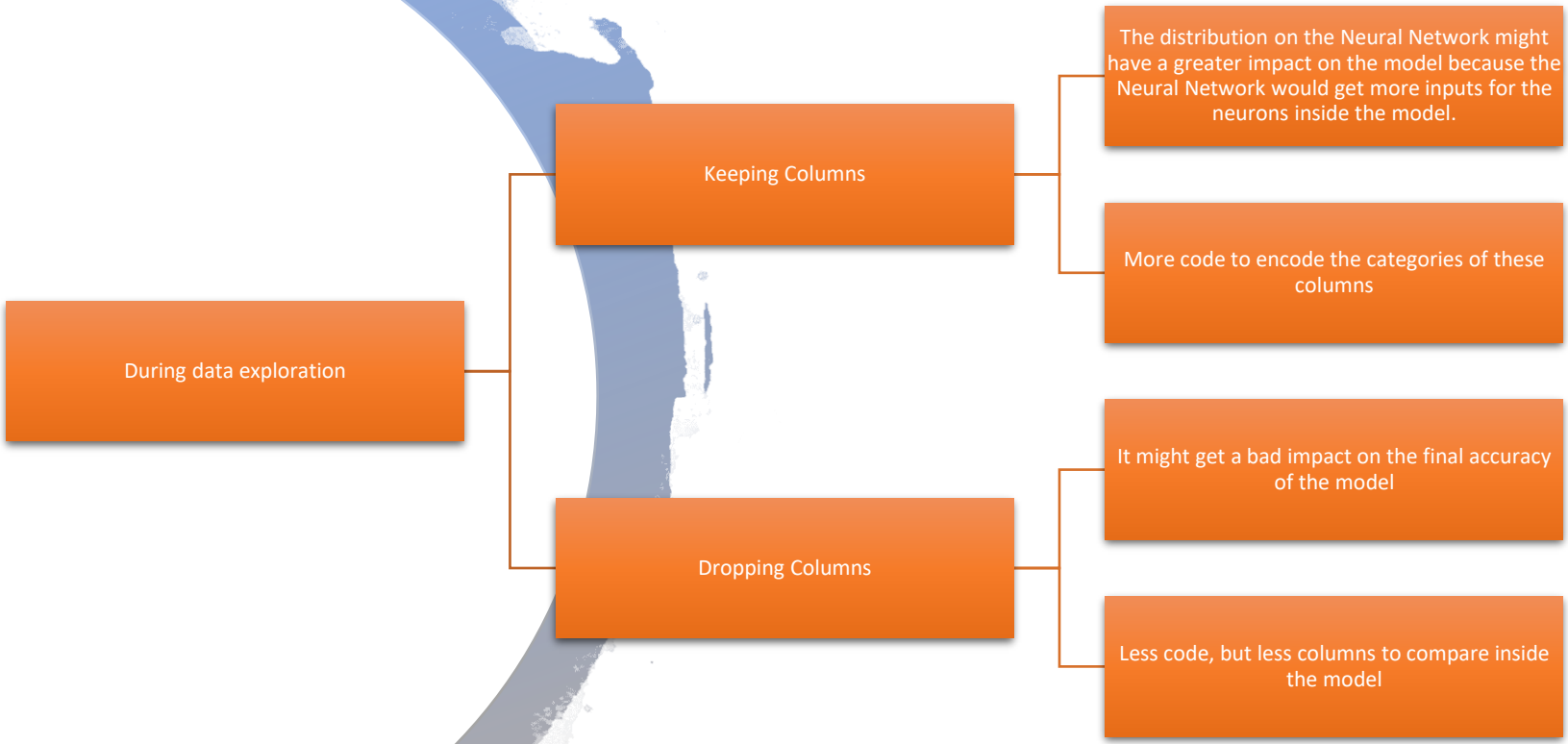


Data Exploration

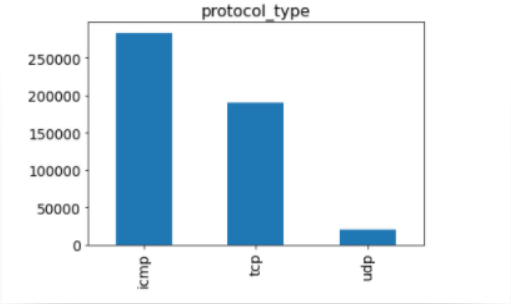
- Dataset contains 42 unique features including the label
- 39 features are numerical values
- 4 features columns are object types
  - Service has 64 unique categories
  - Flag has 11 unique categories
  - Protocol Type has 3 unique categories
  - Label(attack) has 23 unique categories

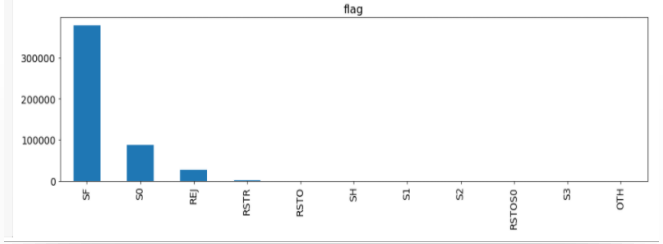


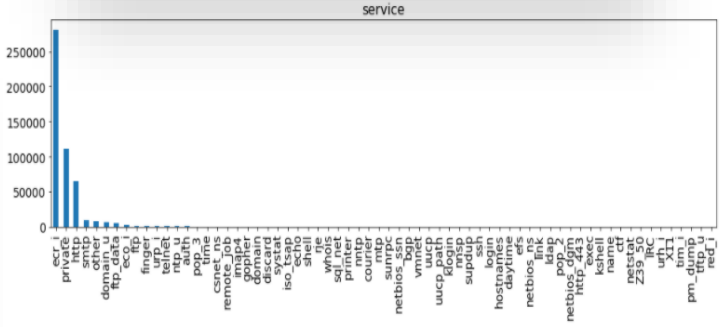
Decisions ?


During data exploration
Keeping Columns
The distribution on the Neural Network might have a greater impact on the model because the Neural Network would get more inputs for the neurons inside the model.
More code to encode the categories of these columns
Dropping Columns
It might get a bad impact on the final accuracy of the model
Less code, but less columns to compare inside the model

Distributions of these Columns







Dataset after keeping the Categories

- After encoding the categories, we ended up with a new database that contains 119 columns which is almost twice as large than the original
- Use labelEncoder() with oneHotEnconder() to transform the string values

Before Encoding

Out[31]:

	service	flag	protocol_type
288190	ecr_i	SF	icmp
338941	ecr_i	SF	icmp
138802	http	SF	tcp
271063	ecr_i	SF	icmp
342685	ecr_i	SF	icmp

After Encoding

Out[32]:

	service	flag	protocol_type
288190	14	9	0
338941	14	9	0
138802	22	9	1
271063	14	9	0
342685	14	9	0

Before Adding the New Categories

In [78]:
```
#printing the shape of all Datasets
print(f'Original Dataset: {df.shape} ,Train Dataset: {dfTrain.shape}, Test Dataset:{dfTest.shape}')
```

```
Original Dataset: (494021, 42) ,Train Dataset: (395216, 42), Test Dataset:(98805, 42)
```

After Adding the New Categories

In [79]:
```
#printing the shape of all Datasets after
print(f'Original Dataset: {df.shape} ,Train Dataset Encoded: {dfTrainFinal.shape}, Test Dataset Enconded:{dfTestFinal.shape}')
```

```
Original Dataset: (494021, 42) ,Train Dataset Encoded: (395216, 119), Test Dataset Enconded:(98805, 119)
```

Lessons

Scaling the data is very important?

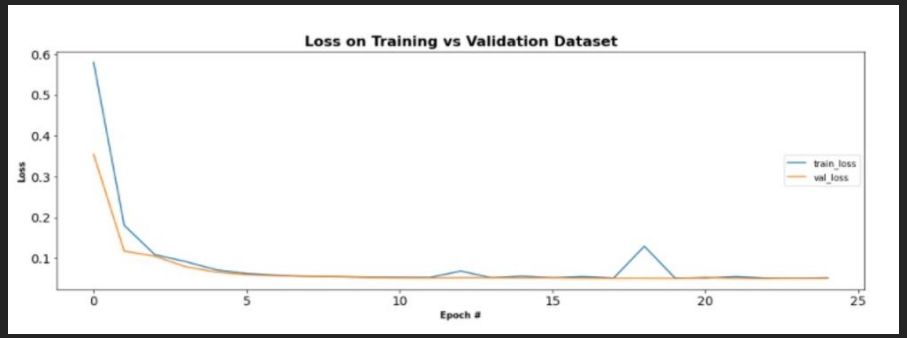
For neural network, scaling the features can create a balance between the different columns.

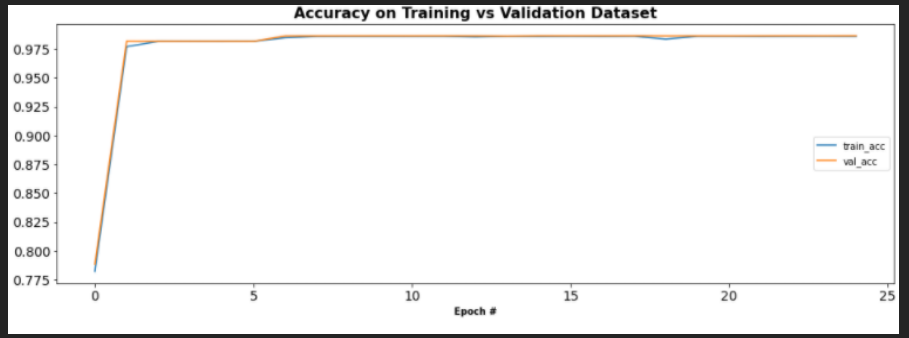
Better performance

Differences in the scales across input variables may increase the difficulty of the problem being modeled.

```
Epoch 23/25
17785/17785 [==============================] - 12s 668us/step - loss: 1.7766 - accuracy: 0.4093 - val_loss: 1.6932 - val_accu
racy: 0.4212
Epoch 24/25
17785/17785 [==============================] - 12s 668us/step - loss: 1.2961 - accuracy: 0.5321 - val_loss: 1.2481 - val_accu
racy: 0.5592
Epoch 25/25
17785/17785 [==============================] - 12s 668us/step - loss: 1.1561 - accuracy: 0.5878 - val_loss: 1.2366 - val_ac
curacy: 0.5559
```

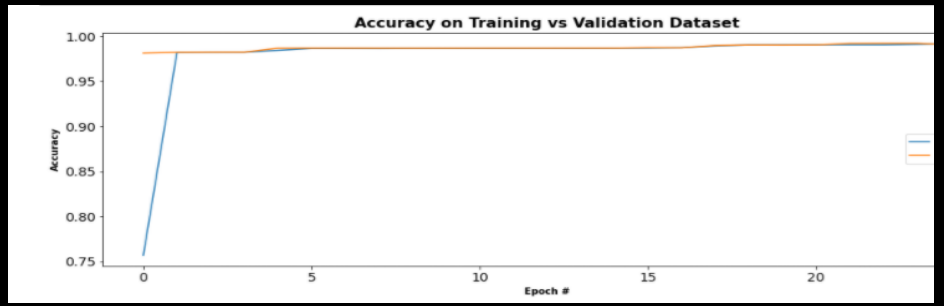
3 Layers





Neural Network models?

5 Layers

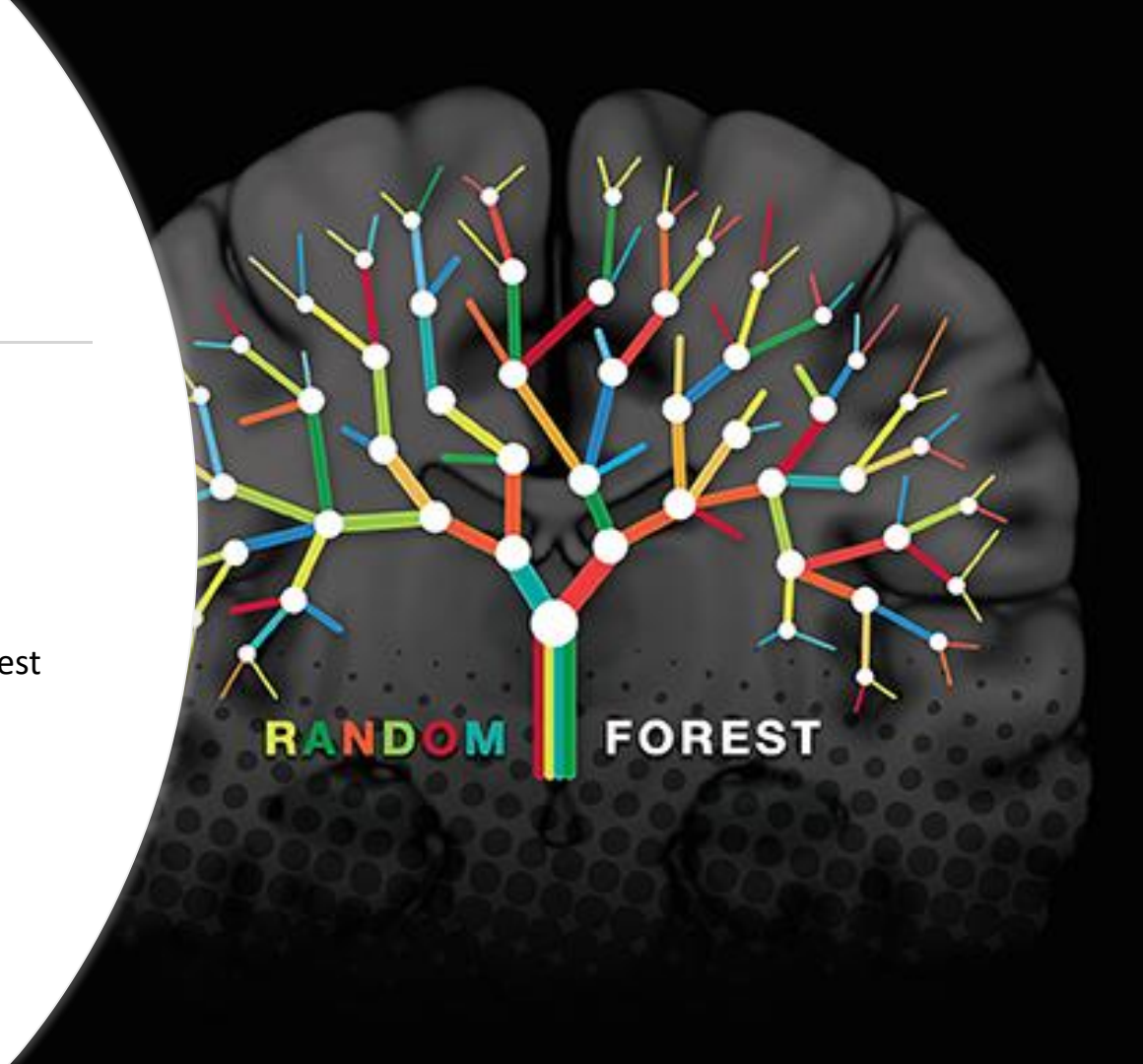


Coming Next

Create a complex Neural Network

- Create model that can have an early stopper for overfitting
- More layers vs More epochs

- Create a second Model using Random Forest
  - Compare Results
  - Advantages and Disadvantages of both models




Code Overview

New Sources

- Brownlee, J. (2020, August 25). How to use Data Scaling Improve Deep Learning Model Stability and Performance. Retrieved October 28, 2020, from https://machinelearningmastery.com/how-to-improve-neural-network-stability-and-modeling-performance-with-data-scaling/

- Valkov, V. (2019, Jul 01). Build your first Neural Network in TensorFlow 2: TensorFlow for Hackers (Part I). Retrieved October 28, 2020, from https://towardsdatascience.com/building-your-first-neural-network-in-tensorflow-2-tensorflow-for-hackers-part-i-e1e2f1dfe7a0

- Lolico, R. (2020, July 06). Creating a Vanilla Neural Network with Tensorflow. Retrieved October 28, 2020, from https://towardsdatascience.com/creating-a-vanilla-neural-network-with-tensorflow-96fec241e369