## Module 21 deep-learning-challenge - screen shots v1.docx

Raj Agrawal / SMU DS / Sep 2023

===

All code / information & output is attached as screen shots as well on the github

Starter_Code-pre-processing-Raj..pdf

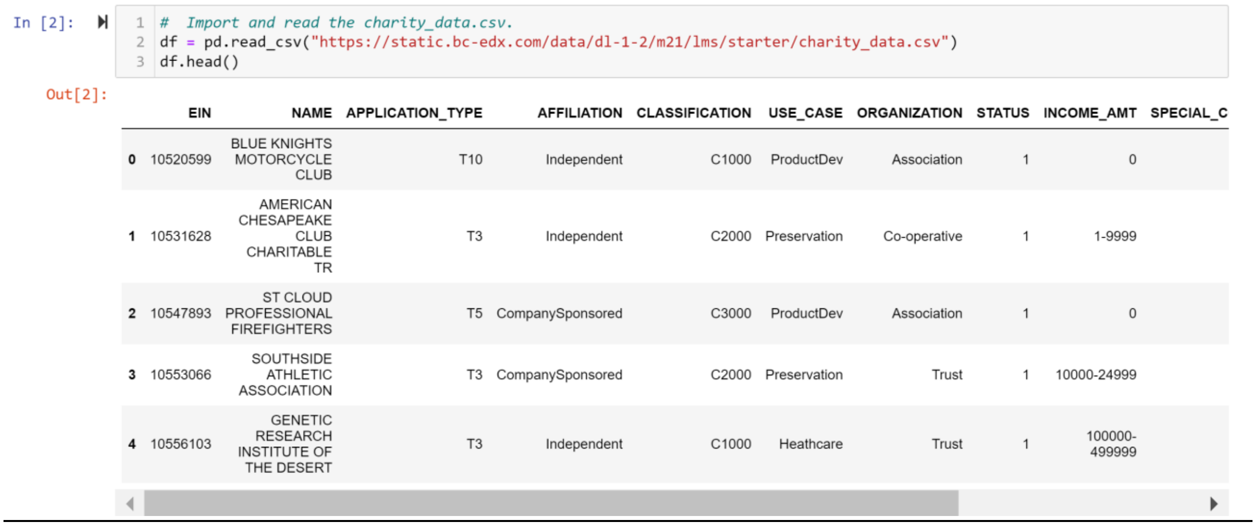
Alphabet_Soup_Charity_Optimization_Model_2.pdf

===

**Background –** The nonprofit foundation Alphabet Soup wants a tool that can help it select the applicants for funding with the best chance of success in their ventures. With your knowledge of machine learning and neural networks, you'll use the features in the provided dataset to create a binary classifier that can predict whether applicants will be successful if funded by Alphabet Soup.

From Alphabet Soup's business team, you have received a CSV containing more than 34,000 organizations that have received funding from Alphabet Soup over the years. Within this dataset are a number of columns that capture metadata about each organization

## Deliverable :

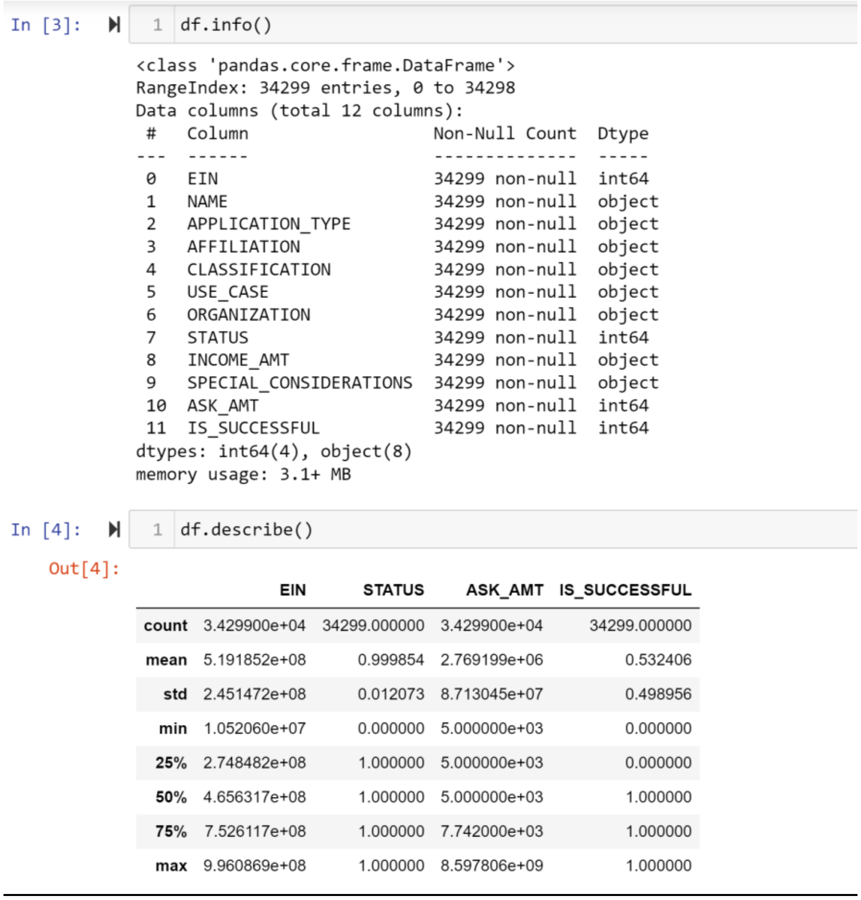


In [2]:

```
#  Import and read the charity_data.csv.
df = pd.read_csv("https://static.bc-edx.com/data/dl-1-2/m21/lms/starter/charity_data.csv")
df.head()
```

Out[2]:

| | EIN | NAME | APPLICATION_TYPE | AFFILIATION | CLASSIFICATION | USE_CASE | ORGANIZATION | STATUS | INCOME_AMT | SPECIAL_C |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 10520599 | BLUE KNIGHTS MOTORCYCLE CLUB | T10 | Independent | C1000 | ProductDev | Association | 1 | 0 | |
| 1 | 10531628 | AMERICAN CHESAPEAKE CLUB CHARITABLE TR | T3 | Independent | C2000 | Preservation | Co-operative | 1 | 1-9999 | |
| 2 | 10547893 | ST CLOUD PROFESSIONAL FIREFIGHTERS | T5 | CompanySponsored | C3000 | ProductDev | Association | 1 | 0 | |
| 3 | 10553066 | SOUTHSIDE ATHLETIC ASSOCIATION | T3 | CompanySponsored | C2000 | Preservation | Trust | 1 | 10000-24999 | |
| 4 | 10556103 | GENETIC RESEARCH INSTITUTE OF THE DESERT | T3 | Independent | C1000 | Heathcare | Trust | 1 | 100000-499999 | |



In [3]:

```
df.info()
```

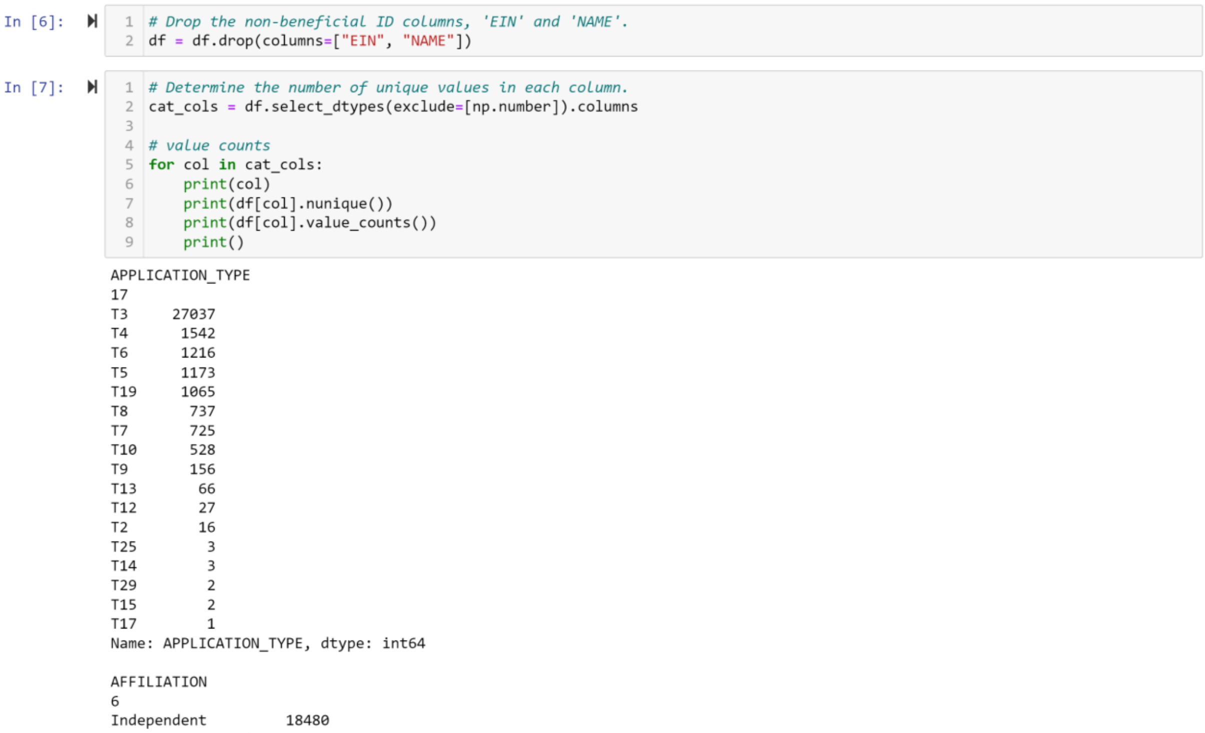
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34299 entries, 0 to 34298
Data columns (total 12 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   EIN                     34299 non-null  int64
 1   NAME                    34299 non-null  object
 2   APPLICATION_TYPE        34299 non-null  object
 3   AFFILIATION             34299 non-null  object
 4   CLASSIFICATION          34299 non-null  object
 5   USE_CASE                34299 non-null  object
 6   ORGANIZATION            34299 non-null  object
 7   STATUS                  34299 non-null  int64
 8   INCOME_AMT              34299 non-null  object
 9   SPECIAL_CONSIDERATIONS  34299 non-null  object
 10  ASK_AMT                 34299 non-null  int64
 11  IS_SUCCESSFUL           34299 non-null  int64
dtypes: int64(4), object(8)
memory usage: 3.1+ MB
```

In [4]:

```
df.describe()
```

Out[4]:

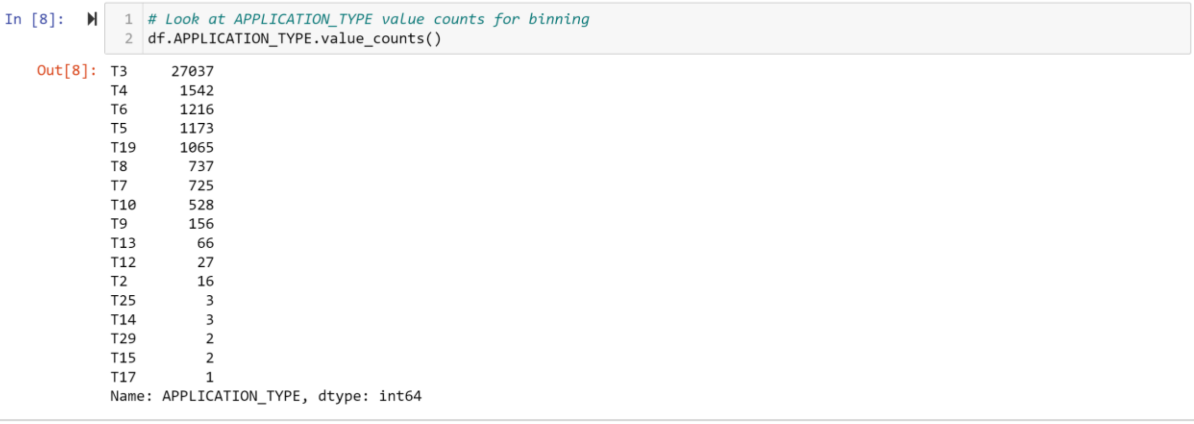
| | EIN | STATUS | ASK_AMT | IS_SUCCESSFUL |
|---|---|---|---|---|
| **count** | 3.429900e+04 | 34299.000000 | 3.429900e+04 | 34299.000000 |
| **mean** | 5.191852e+08 | 0.999854 | 2.769199e+06 | 0.532406 |
| **std** | 2.451472e+08 | 0.012073 | 8.713045e+07 | 0.498956 |
| **min** | 1.052060e+07 | 0.000000 | 5.000000e+03 | 0.000000 |
| **25%** | 2.748482e+08 | 1.000000 | 5.000000e+03 | 0.000000 |
| **50%** | 4.656317e+08 | 1.000000 | 5.000000e+03 | 1.000000 |
| **75%** | 7.526117e+08 | 1.000000 | 7.742000e+03 | 1.000000 |
| **max** | 9.960869e+08 | 1.000000 | 8.597806e+09 | 1.000000 |



```
In [6]:
# Drop the non-beneficial ID columns, 'EIN' and 'NAME'.
df = df.drop(columns=["EIN", "NAME"])
```

```
In [7]:
# Determine the number of unique values in each column.
cat_cols = df.select_dtypes(exclude=[np.number]).columns

# value counts
for col in cat_cols:
    print(col)
    print(df[col].nunique())
    print(df[col].value_counts())
    print()
```

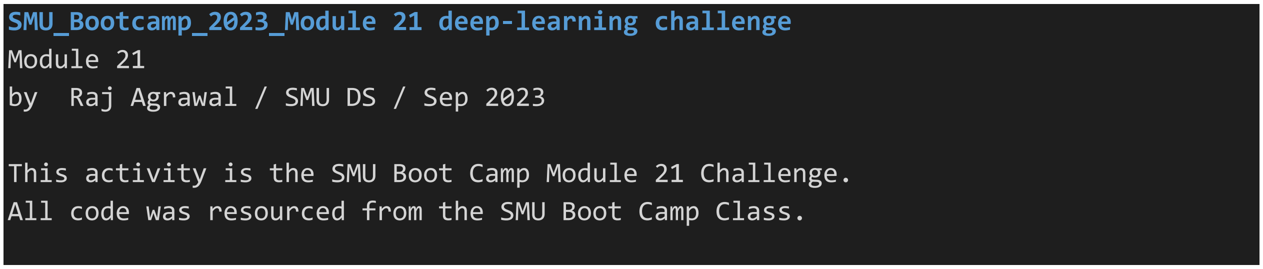
```
APPLICATION_TYPE
17
T3     27037
T4      1542
T6      1216
T5      1173
T19     1065
T8       737
T7       725
T10      528
T9       156
T13       66
T12       27
T2        16
T25        3
T14        3
T29        2
T15        2
T17        1
Name: APPLICATION_TYPE, dtype: int64

AFFILIATION
6
Independent         18480
```



```
In [8]:
# Look at APPLICATION_TYPE value counts for binning
df.APPLICATION_TYPE.value_counts()
```

```
Out[8]: T3     27037
        T4      1542
        T6      1216
        T5      1173
        T19     1065
        T8       737
        T7       725
        T10      528
        T9       156
        T13       66
        T12       27
        T2        16
        T25        3
        T14        3
        T29        2
        T15        2
        T17        1
        Name: APPLICATION_TYPE, dtype: int64
```

## Readme.md



```
SMU_Bootcamp_2023_Module 21 deep-learning challenge
Module 21
by  Raj Agrawal / SMU DS / Sep 2023

This activity is the SMU Boot Camp Module 21 Challenge.
All code was resourced from the SMU Boot Camp Class.
```

CODE IS RUN USING GOOGLE COLABORATORY

Module 21 Challenge
Background
The nonprofit foundation Alphabet Soup wants a tool that can help it select the applicants for funding with the best chance of success in their ventures. With your knowledge of machine learning and neural networks, you’ll use the features in the provided dataset to create a binary classifier that can predict whether applicants will be successful if funded by Alphabet Soup.

From Alphabet Soup’s business team, you have received a CSV containing more than 34,000 organizations that have received funding from Alphabet Soup over the years. Within this dataset are a number of columns that capture metadata about each organization, such as:

EIN and NAME—Identification columns

APPLICATION_TYPE—Alphabet Soup application type

AFFILIATION—Affiliated sector of industry

CLASSIFICATION—Government organization classification

USE_CASE—Use case for funding

ORGANIZATION—Organization type

STATUS—Active status

INCOME_AMT—Income classification

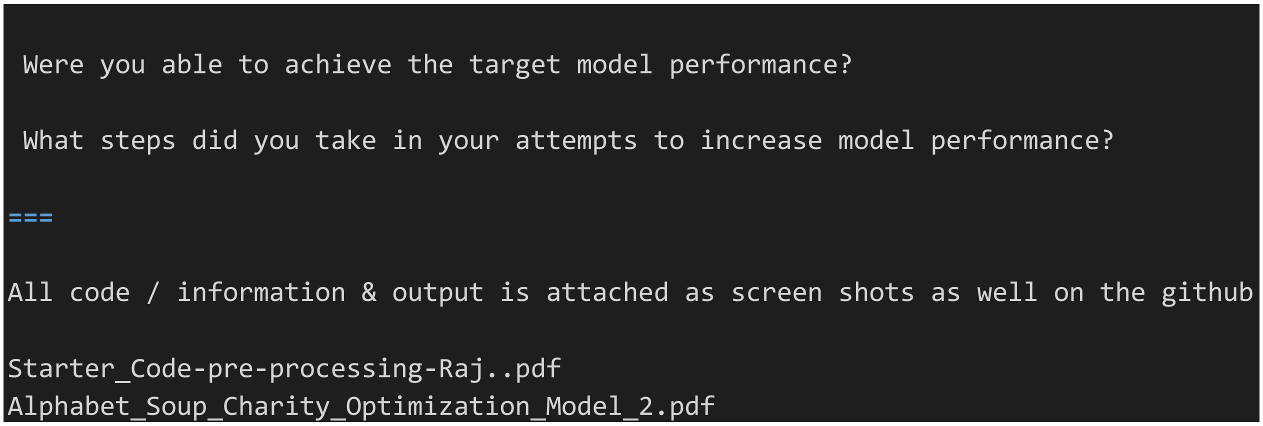
SPECIAL_CONSIDERATIONS—Special considerations for application

ASK_AMT—Funding amount requested

IS_SUCCESSFUL—Was the money used effectively

Step 1: Preprocess the Data
Step 2: Compile, Train, and Evaluate the Model
Using your knowledge of TensorFlow, you’ll design a neural network, or deep learning model, to create a binary classification model that can predict if an Alphabet Soup-funded organization will be successful based on the features in the dataset. You’ll need to think about how many inputs there are before determining the number of neurons and layers in your model. Once you’ve completed that step,

```
you’ll compile, train, and evaluate your binary classification model to calculate
the model’s loss and accuracy.

Step 3: Optimize the Model
Using your knowledge of TensorFlow, optimize your model to achieve a target
predictive accuracy higher than 75%. Use any or all of the following methods to
optimize your model:

Adjust the input data to ensure that no variables or outliers are causing
confusion in the model, such as: Dropping more or fewer columns.

Creating more bins for rare occurrences in columns.

Increasing or decreasing the number of values for each bin.

Add more neurons to a hidden layer.

Add more hidden layers.

Use different activation functions for the hidden layers.

Add or reduce the number of epochs to the training regimen.

Step 4: Write a Report on the Neural Network Model
For this part of the assignment, you’ll write a report on the performance of the
deep learning model you created for Alphabet Soup. The report should contain the
following:

Overview of the analysis: Explain the purpose of this analysis.

Results: Using bulleted lists and images to support your answers, address the
following questions:

Data Preprocessing

  What variable(s) are the target(s) for your model?

  What variable(s) are the features for your model?

  What variable(s) should be removed from the input data because they are neither
targets nor features?
Compiling, Training, and Evaluating the Model

 How many neurons, layers, and activation functions did you select for your
neural network model, and why?
```



```
 Were you able to achieve the target model performance?

 What steps did you take in your attempts to increase model performance?

===

All code / information & output is attached as screen shots as well on the github

Starter_Code-pre-processing-Raj..pdf
Alphabet_Soup_Charity_Optimization_Model_2.pdf
```

**#END**