Predicting A Personality Trait Using A Machine Learning Algorithm

Senior Capstone I IT451 - 01

Dustin Powell

Cumberland University

Spring 2020 4/22/20

Table of Contents

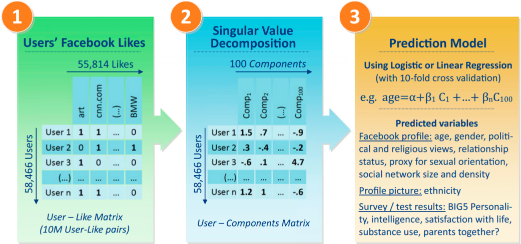
1. Introduction
2. PNAS Case Study
3. Analytics
4. Regression
5. Machine Learning
6. Experimental Planning
7. Experimental Setup
8. Experimental Results
9. Conclusion
10. References
11. Appendix

Introduction

- According to Our World In Data, 3.5 billion people use the internet, with 2.3 billion using the social network Facebook$^{\circledR}$ [1].
- About 640,000 new people access the internet daily [1].
- More of people's "information" is becoming publicly viewable compared to before the creation of the internet.
- The purpose of this work is to use online data to demonstrate that using a limited amount of information can lead to uncovering private details about an individual.
- A article published by the Proceedings of the National Academy of Sciences of the United States of America (PNAS) demonstrates this predictability with statistics [2].
- Python code is used to further investigate the overall concept from the PNAS study.

Case Study in PNAS

- The post–liking habits of 58,466 people were able to predict various private aspects about the user, such as age, race, intelligence, and political views [2].
- The data collected had a overall average of 170 likes per user [2].



	art	cnn.com	(...)	BMW
User 1	1	1	...	0
User 2	0	1	...	1
User 3	1	0	...	0
(...)	...	...	...	...
User n	1	1	...	0

User – Like Matrix (10M User-Like pairs)

	$Comp_1$	$Comp_2$	(...)	$Comp_{100}$
User 1	1.5	.7	...	-.9
User 2	.3	-.4	...	-.2
User 3	-.6	.1	...	4.7
(...)	...	...	...	...
User n	1.2	1	...	-.6

User – Components Matrix

Using Logistic or Linear Regression
(with 10-fold cross validation)

e.g. $age=\alpha+\beta_1 C_1 + \ldots + \beta_n C_{100}$

Predicted variables

Facebook profile: age, gender, political and religious views, relationship status, proxy for sexual orientation, social network size and density

Profile picture: ethnicity

Survey / test results: BIG5 Personality, intelligence, satisfaction with life, substance use, parents together?

Figure: The process from the papers of the PNAS study outlining the steps used to make predictions with Facebook® data [2].

Questions to Address

The alarming and possible harmful implications of the PNAS study prompts the following question.

Primary Question

Using a machine–learning algorithm what is the minimum amount of information needed to predict the personality trait of a given person accurately?

Hypothesis:

1. A given person's personality trait is predictable with one feature.
2. An overall average accuracy score of 90 percent or higher can be obtained.
3. The overall average accuracy will increase as features in the model increase.

Big Data

The case study conducted by Kosinski, Stillwell, and Gaepel (2013) reflects an overarching concept of a bigger project at hand called **big data**.

Big Data – The mass collection of data for the purpose of research by scientist, government officials, and commercial institutions.

- Data is generated from users by visiting websites, liking posts, and watching videos get stored into large databases [3].
- The database are located in warehouses full of computers called servers that have a range of purposes [3].
- Companies such as Google®, Amazon®, and Facebook® use these massive amounts of data to make predictions about their users [3].

Three Types of Data

Three types of data that are collected:

- **Structured** – Any data that is clearly understood, defined, and can immediately store in databases without any preprocessing [3].
  - Example: Sales Figures
- **Unstructured** – Any data that is vague and convoluted, which requires the information to be further processed and organized for it to be of use and stored [3].
  - Example: Social Media Posts
- **Semi–Structured** – Any data considered the middle ground between structured and unstructured data and has some organized and unorganized aspects to the structure of the information [3].
  - Example: SQL Scripts, Server Logs

Forms of Analytics

Three forms of analytics are performed on the data stored in databases [3].

- **Descriptive Analytics** – Analysis of data from past occurrences of events.
- **Predictive Analytics** – Analysis of data to make predictions.
- **Prescriptive Analytics** – Analysis of data to make decisions.

The Uses of Analytics

The application of statistics and computer science allow for these various forms of analytics to be conducted and used in various ways [3].

- Analytics is used by businesses to make educated decisions by using the resource as a way to understand markets [3].
- Predictive analytics can be used for predicting sales outcomes of a company's product versus the market competition [3].
- Marketing uses predictive analytics to determine which advertisments to deliver to internet users [3].
    - Example: Youtube Video Advertisements

Overview of The Relationship

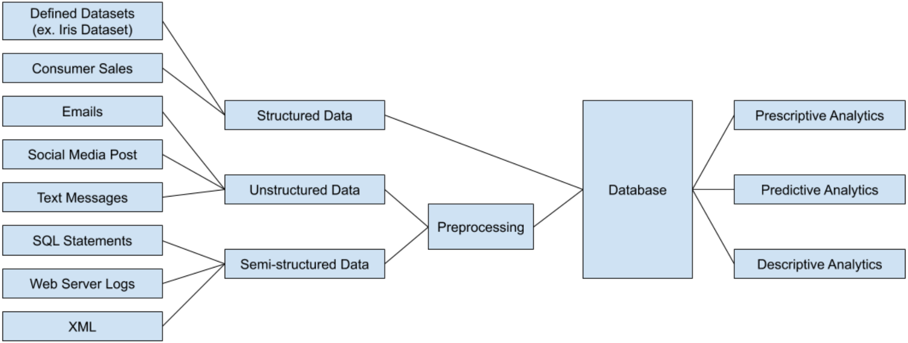


Figure: The relationship between the three types of Big Data and analytics inspired, from the article by S. Jeble, S. Kumari, and Y. Patil [3].

Regression and Predictive Analytics

One of the biggest purposes of collecting data is to find relationships that occur between data.

Linear Regression

- Determines the possible correlation between two variables, one independent and one dependent using a function that best represents the correlation between the two variables with the smallest error between the correlation, $\hat{y} = \alpha + \beta X$ [4].

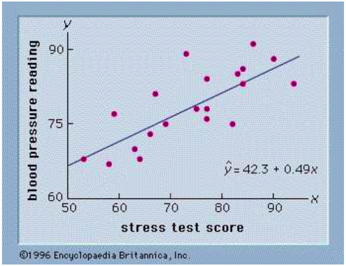


Figure: The correlation between blood pressure readings and stress test scores data. The x-axis represents stress test scores and the y-axis is blood pressure reading [11].

Regression and Predictive Analytic Continued

Multiple Regression

- Determines the possible correlation between multiple independent and one dependent variables using a function that best represents the correlation [4].

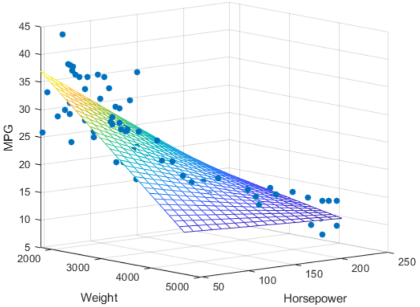


Figure: Multiple regression plane created from a set of data containing miles per gallon of fuel, vehicle weight, and horsepower [12].

What is Machine Learning

Machine Learning is a subsection of artificial intelligence that is specifically focused on using algorithms to make conclusions about a set of information [5].

Their are three types of machine learning:

- **Supervised Machine Learning** – Using a base set of data to train the algorithm which then makes predictions based on the learned information [5].
- **Unsupervised Machine Learning** – The use of an algorithm to classify and find relationships within a set of data without knowing of any results [5].
- **Reinforcement Machine Learning** – The use of a agent that learns from task carried out that results in a reward for correctly carrying out the task [5].

The Iris Flower Data Set and Machine Learning

One of the most well–known data sets used for machine learning examples is the iris flower data set.

- The iris data set consists of measurements were the sepal length and width, the petal length and width, and the subspecies name of each of the iris flowers [5].
- Each flower within the data set can be known as an instance, object, or set of features [5].
- The measurements that make up the flower are known as features [5].
- The subspecies names of the flowers are called the class label or targets, which is the value to be predicted [5].

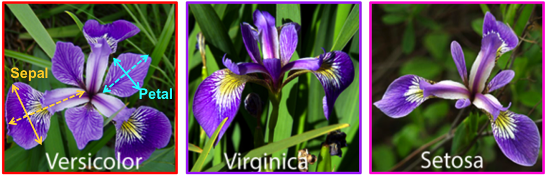


Figure: An image showing the iris flowers species and how they are measured [16].

Internal Workings of Machine Learning (Artificial Neurons)

Machine learning makes decisions using artificial neurons [5].

- **Artificial neuron** – A mathematical representation of the neurons found in a biological brain [5].
- Their are two types of neurons called the **perceptron** and **Adaline** [5].

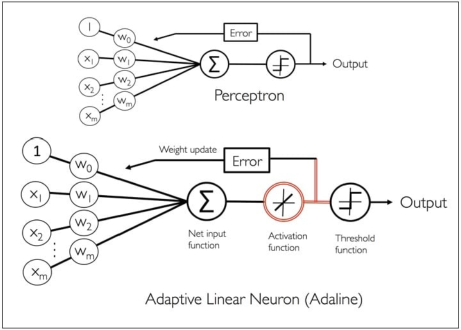


Figure: A diagram that shows the structure of the perceptron and Adaline neurons [9].

Internal Workings of Machine Learning (Artificial Neurons)

Internal Workings of Machine Learning (Artificial Neurons)

Additionally, to further enhance the accuracy of the artificial neuron, multiple iterations over a set of data can be used to keep updating the weights to create an accurate model; with each of these times, the set of data goes through the artificial neuron is called an epoch [5].

After an epoch, the weights update using the various functions depending on the type of artificial neuron used [5].

Just like the human brain, all of the artificial neurons can have connections to other artificial neurons creating a structure that resembles a human brain, which is called an **artificial neural network** [5].

Internal Workings of Machine Learning (Neural Network)

Three types of layers exist within a artificial neural network [5].

- **Input Layer** – The first set of neurons that interact with the data [5].
- **Output Layer** – The layer that outputs the results concluded from the input data [5].
- **Hidden Layer** – The layers between the input and output layer [5].

A artificial neural network that contains multiple hidden layers is called a deep artificial neural network [5].

One example of an artificial neural network is the multilayer perceptron (MLP)

- **Feedforward neural network** – the artificial neurons' outputs as inputs for the next layer of artificial neurons [5].

Internal Workings of Machine Learning (Neural Network) Continued

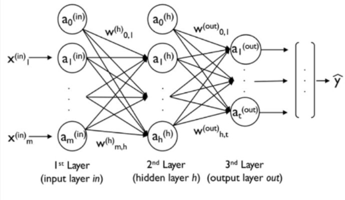


Figure: A diagram showing the layout of an artificial neural network called the mutilayer perceptron [5].

The MLP has to conduct four necessary steps during each epoch to create an accurate model [5].

1. Feed forward – The data is sent through the neural network to determine an output [5].
2. Output is determined and the error is found by minimizing the utilization of a cost function [5].
3. Backpropgration – Finding the derivative of the error using each weight in the model and updating the weights of the model repeats steps one through three as desired (epoch) [5].
4. Lastly, make predictions using forward propagation and a threshold function to predict an outcome [5].

Sub-types of Machine Learning

Sub-types of machine learning are:

- **Classification** – A form of machine learning algorithms used to predict an output of one or more features from a set of input data [5].
- **Regression** – Use of regression to make predictions from a set of data.
- **Clustering** – Group data together to find similarities between the elements in the dataset [5].
- **Dimensionality reduction** – Shrinks the size of a set of data into smaller dimensions [5]. The primary use of dimensionality reduction is to create visualizations of complex graphs of a collection of data [5].

Sub-types of Machine Learning

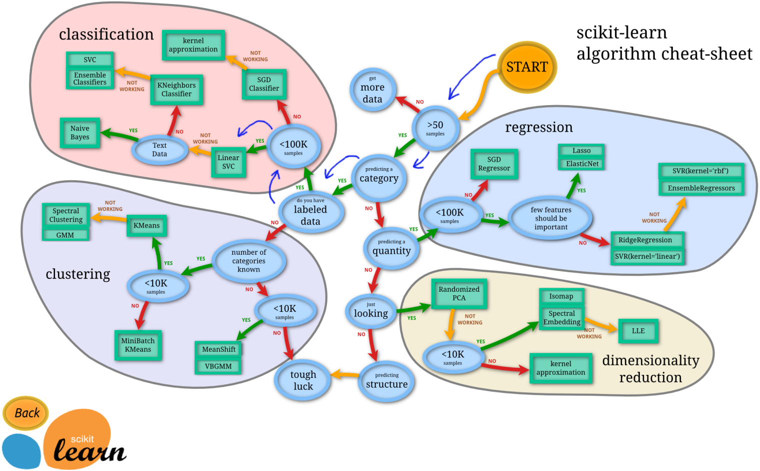


Figure: Map from the Scikit–learn website provided to help determine which estimators to use for machine learning [8]. The blue arrows represent the path taken.

Scikit–Learn Vs. TensorFlow

Two libraries and platforms used for machine learning with Python:

Sci–kit Learn

- Only uses Central Processing Unit (CPU) [5].
- Pre–made algorithms by various researchers [8].
- Can perform Classification, Regression, Clustering, and Dimensionality reduction [8].

TensorFlow

- Can use CPU and graphics processing unit (GPU) for processing data [5].
- Has the same functions as Scikit–learn [5].
- Custom neural networks using tensors [5].

Sci–kit learn was chosen over to TensorFlow because of familiarity at an early stage of the project and time constraints of the project.

The Experiment's Plan

Once a general understanding of the various topics was complete an experiment was set up in two parts.

1. A online data set.
2. An algorithm to process and display the information.

The algorithm will have to be able to compare the results of the machine learning models using standardize variables to show the relationship between the amount of data the models are using to make predictions.

The Experiment's Plan Continued

The standardized variables:

- Total time per set of models
- Number of features analyzed by the models
- Overall average accuracy score of analyzed models

One of the most important things about the algorithm is to make it easy to understand. The estimators and input data are designed to ensure that other parts in the code do not affect the final results of the test.

The Dataset

The chosen dataset is a study posted on Kaggle.com of people between the ages 15 and 30 years old with over 1000 responses [7].

The question within the survey pertained to music, movies, hobbies and interests, phobias, health habits, spending habits, and demographics [7].

The study contains 150 questions overall, with the majority over the topics of personality traits, views on life, and opinions [7].

The most of the questions are recorded as integers 1 through 4, and others based on category phrases [7].

The Dataset Preparation For Testing

The data was manually preprocessed with LibreOffice Calc

1. Phrases were changed to integer responses
2. Missing values were set to 999
3. The data was split into seven CSV files that had to match the set a features being tested

The predicted question from the survey is anger with the output as the interger rating according to the survey. The features used to predict a person's anger are the spending habit questions.

The Project Algorithm

The Main Challenge

- Choosing the proper estimator to work with as there are many of them.

The diagram recommends using the algorithms SVM and KNearestNeighbor, but we ultimately decided to use Scikit–learn's version of the multilayer perceptron called the MLPClassifer, which is not listed on the diagram [8].

The selection of the MLPClassifer ultimately leads up to the final algorithm after numerous instances of tinkering with the machine–learning code.

The code is divided into three sections called Data Handling, Model Creation and Prediction, and Displaying Results.

The code was inspired from a sample set by Jason Brownlee from machinelearningmastery.com that is used on the iris dataset [15].

The Project Algorithm

The models follows the following step within the program algorithm.

1. Read Data – Loading data from file.
2. Create Model – Using estimator to make the model.
3. Validate Model – Determine the accuracy of the model.
4. Store Results – Calculates and stores the result for a set of models.
5. Repeat last three steps till all sets of features are analyzed with all estimators.
6. Display Collected Data – Data stored is plotted on to 3D scatter plot for analysis.

The Experiment

The experiment will have three tests with each of the three estimators creating models from the set of features each test.

The number of models analyzed per set of features is set to ten.

- Estimator 1: Sci–kit learn default parameters with 1400 epochs.
- Estimator 2: The activation function is configured to tangent with 1400 epochs.
- Estimator 3: The activation function is set to logistic with 1400 epochs.

The reason the number of epochs are the same is because of convergence problems with the estimators were not reaching the most optimized point to determine the weights.

The Results: Test 1

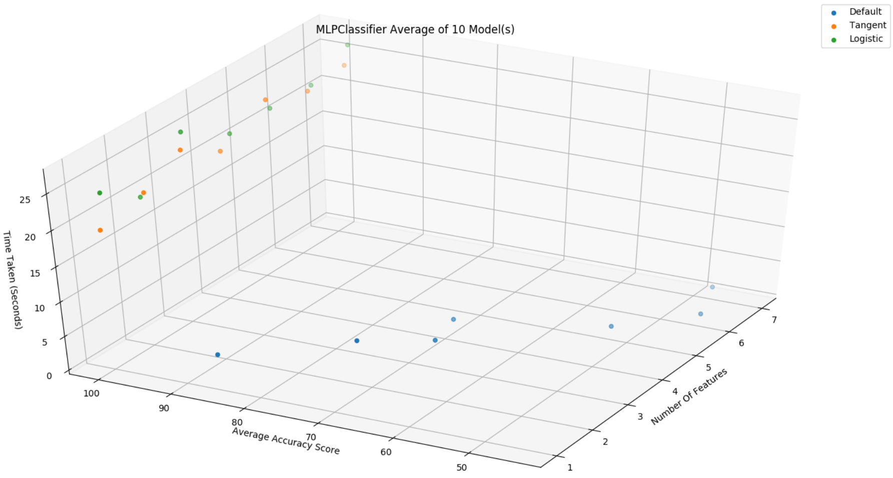


Figure: The results of the first test results of the MLPClassifer Comparison algorithm. X–axis is Number of Features, Y–axis is average accuracy score, and Z–axis is Time Taken in seconds.

The Results: Test 2

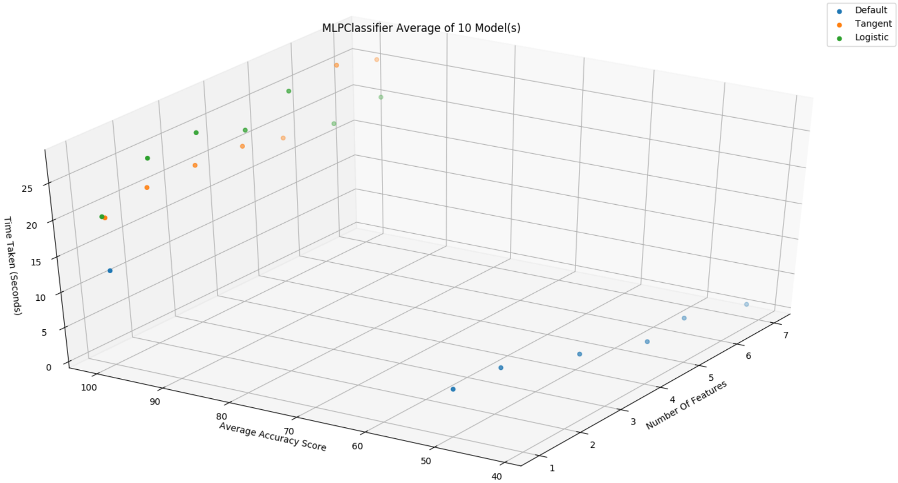


Figure: The results of the second test results of the MLPClassifer Comparison algorithm. X–axis is Number of Features, Y–axis is average accuracy score, and Z–axis is Time Taken in seconds.

The Results: Test 3

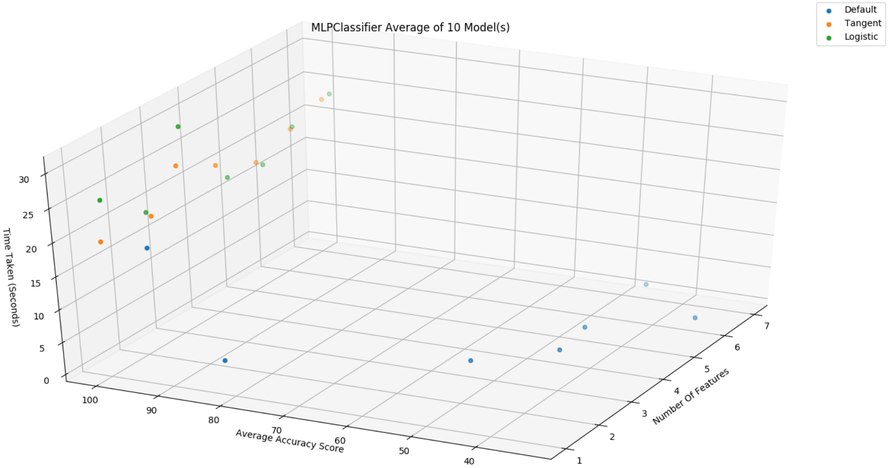


Figure: The results of the third test results of the MLPClassifer Comparison algorithm. X–axis is Number of Features, Y–axis is average accuracy score, and Z–axis is Time Taken in seconds.

The Results: Conclusion

After the testing was complete it led to the following conclusions:

- Out of all three tests combined, the Tangent and Logistic models had a higher average accuracy score but took more time to create and validate.
- The Default models were sporadic, with the majority of the one feature average accuracy scores being above at least 80 percent.
- The personality of a person with one feature is predictable using a multilayer perceptron classifier.
- The estimators must be tuned and tested to have the models make accurate predictions about a personality trait of a person.

Future Works

Given the results and concepts learned from the project, some other aspects could have been included.

- All parameters of the MLPClassifer could have been tested during the project, like the effects of tolerances on models.
- More depth by observing every process of multiple machine learning algorithms in exquisite detail, outlining the many mathematical algorithms that enable machine learning to work.
- Other types and sub-types of machine learning could have been used to investigate the data further and make conclusions from the data.

The overall limiting factor of the project is the time constraint of only one semester.

The work can be further built upon to analyze the data set or continue learning about the topic of machine learning.

Conclusion

Key take–aways from the project:

- Only one feature is needed to predict the personality trait of a given person.
- The goal was accomplished by fine-tuning of the parameters of the estimators to increase the accuracy score of the models.
- The broadness of predictive analytics implies that many industries use analytics for different purposes but have a common ground.
- Self–learning, self–discipline, and practice can allow for new exciting things to be learned.

References

- M. Roser, H. Ritchie, and E. Ortiz-Ospina (2020), "Internet". Published online at OurWorldInData.org. Retrieved from: 'https://ourworldindata.org/internet' [Website]

- M. Kosinski, D. Stillwell, and T. Graepel (2013), "Private traits and attributes are predicatable from digital records of human behavior", *PNAS* 110. 15. , 5802 - 5805. Google Scholar. Accessed January 19th 2020 URL: https://www.pnas.org/content/pnas/110/15/5802.full.pdf [Journal]

- S. Jeble, S. Kumari, and Y. Patil (2016), "Role of big data and predictive analytics". *International Journal of Automation and Logistics*. 2. 307-331. 10.1504/IJAL.2016.10001272. Google Scholar. Accessed January 21st 2020 URL: https://www.researchgate.net/publication/309809606 [Journal]

- J. Bothe, J. L. Brudney, and K. J. Meier (2015), *Applied Statistics For Public and Nonprofit Administration*. 4th ed. Stamford, CT USA Cengage Learning. [eBook]

- V. Mirjalili and S. Raschka (2017), *Python Machine Learning*. 2nd ed. Birmingham, UK Packt Publishing. [eBook]

- "Iris Data Set"', University of California Irvine Machine Learning Repository. Accessed February 5th 2020 URL: https://archive.ics.uci.edu/ml/datasets/iris [Website]

- M. Sabo (2016), "2013 Young People Survey". Accessed February 13th 2020 URL: https://www.kaggle.com/miroslavsabo/young-people-survey [Website]

References Continued

"scikit-learn Machine Learning in Python". Accessed February 17th 2020 URL: https://scikit-learn.org/stable/index.html [Website]

www.simplilearn.com (n.d), "How to Train an Artificial Neural Network". Accessed April 13th 2020 URL: https://www.simplilearn.com/how-to-train-artificial-neural-network-tutorial [Online Image]

S. Kakade, S. Ozdemir, and M. Tibaldeschi (2018), *Principles of Data Science*. 2nd ed. Birmingham, UK Packt Publishing Ltd. [eBook]

D. Anderson, D. Sweeney, and T. Williams (2020), "Statistics", Encyclopaedia Britannica. Accessed March 23rd 2020 URL: https://www.britannica.com/science/statistics/Experimental-design#ref367488 [Online Image]

"regress" (n.d), MathWorks. Accessed March 23rd 2020 URL: https://www.mathworks.com/help/stats/regress.html [Online Image]

J. Brownless (2018), "How to Calculate the SVD from Scratch with Python". Accessed March 30th 2020 URL: https://machinelearningmastery.com/singular-value-decomposition-for-machine-learning/ [Website]

K. Krzyk (2018), "Coding Deep Learning for Beginners Linear Regression (Part 2): Cost Function". Accessed March 30th 2020 URL: https://towardsdatascience.com/coding-deep-learning-for-beginners-linear-regression-part-2-cost-function-49545303d29f [Website]

J. Brownlee (2020), "Your First Machine Learning Project in Python Step-By-Step". Accessed February 5th 2020 URL: https://machinelearningmastery.com/machine-learning-in-python-step-by-step/ [Website]

S. Flaloke (2016), "Classification of Iris Varieties". Accessed April 21st 2020 URL: http://suruchifialoke.com/2016-10-13-machine-learning-tutorial-iris-classification/ [Online Image]

Thank You

Thank you for your time to attend the presentation.

Please do not hesitate to ask questions.

(By the way, if doing a senior project, start early.)

The Project Code: Initialization

The initialization of variables for the program.

```
#Main Function ----------------------------------------------------------------
def main():

    #Local Declarations -----------------------------------

    #Initialization of variables
    average = 10              #Number of times to make a model to determine
                              # overall average accuracy score

    feature = [1,2,3,4,5,6,7] #List of the amount of features used per model

    MLPdata1 = []             #List to store overall average accuracy
                              # of MLPClassifier

    MLPdata2 = []             #List to store overall average accuracy
                              # of MLPClassifier

    MLPdata3 = []             #List to store overall average accuracy
                              # of MLPClassifier

    final_time1 = []          #List to store time taken to analyze both models
                              # with n features

    final_time2 = []          #List to store time taken to analyze both models
                              # with n features

    final_time3 = []          #List to store time taken to analyze both models
                              # with n features
```

The Project Code: Data Handling Section

```
#Local Statements -------------------------------------

    #SECTION: Data Handling----------------------------------------------------

    print("        MLPClassifier Model Comparison         ")
    print("-----------------------------------------------")
    print("Please wait until the 3D scatter plot displays ")

    #For loop to iterate through all models being observed
    for iter_selection in range(3):

        #For loop to iterate through data sets used by the models and estimators
        for i in range(7):

            #If statements to that determine which set of data is being used
            if i == 0:
                #Load data set values from CSV file
                csvData = "responses-1feature-finances.csv"

                #Array to store names of feature(s) and target feature
                names = ['Getting angry', 'Finances']

                #Stores the data values with names of questions
                dataset = read_csv(csvData, names=names)

                #Stores array values into an array
                array = dataset.values

                #Initialization of two NumPy X and y arrays
                X1 = array[:,0:2]
                y1 = array[:,0]

                #Function to split the question values into a training set where 20 percent of the
                # data set is used for validation
                (X_train, X_validation, Y_train, Y_validation) = train_test_split(X1, y1, test_size=0.20)
```

The Project Code: Data Handling Section

```
if i == 1:
    #Load data set values from CSV file
    csvData = "responses-2features.csv"

    #Array to store names of feature(s) and target feature
    names = [’Getting angry’, ’Finances’,’Shopping centres’]

    #Stores the data values with names of questions
    dataset = read_csv(csvData, names=names)

    #Stores array values into an array
    array = dataset.values

    #Initialization of two NumPy X and y arrays
    X2 = array[:,0:3]
    y2 = array[:,0]

    #Function to split the question values into a training set where 20 percent of the
    # data set is used for validation
    (X_train, X_validation, Y_train, Y_validation) = train_test_split(X2, y2, test_size=0.20)

if i == 2:
    #Load data set values from CSV file
    csvData = "responses-3features.csv"

    #Array to store names of feature(s) and target feature
    names = [’Getting angry’, ’Finances’,’Shopping centres’, ’Branded clothing’]

    #Stores the data values with names of questions
    dataset = read_csv(csvData, names=names)

    #Stores array values into an array
    array = dataset.values

    #Initialization of two numpy X and y arrays
    X3 = array[:,0:4]
    y3 = array[:,0]

    #Function to split the question values into a training set where 20 percent of the
    # data set is used for validation
    (X_train, X_validation, Y_train, Y_validation) = train_test_split(X3, y3, test_size=0.20)
```

The Project Code: Data Handling Section

```
if i == 3:
    #Load data set values from CSV file
    csvData = "responses-4features.csv"

    #Array to store names of feature(s) and target feature
    names = [’Getting angry’, ’Finances’,’Shopping centres’, ’Branded clothing’,
             ’Entertainment spending’]

    #Stores the data values with names of questions
    dataset = read_csv(csvData, names=names)

    #Stores array values into an array
    array = dataset.values

    #Initialization of two numpy X and y arrays
    X4 = array[:,0:5]
    y4 = array[:,0]

    #Function to split the question values into a training set where 20 percent of the
    # data set is used for validation
    (X_train, X_validation, Y_train, Y_validation) = train_test_split(X4, y4, test_size=0.20)
```

The Project Code: Data Handling Section

```
if i == 4:
    #Load data set values from CSV file
    csvData = "responses-5features.csv"

    #Array to store names of feature(s) and target feature
    names = [’Getting angry’, ’Finances’,’Shopping centres’, ’Branded clothing’,
             ’Entertainment spending’, ’Spending on looks’]

    #Stores the data values with names of questions
    dataset = read_csv(csvData, names=names)

    #Stores array values into an array
    array = dataset.values

    #Initialization of two numpy X and y arrays
    X5 = array[:,0:6]
    y5 = array[:,0]

    #Function to split the question values into a training set where 20 percent of the
    # data set is used for validation
    (X_train, X_validation, Y_train, Y_validation) = train_test_split(X5, y5, test_size=0.20)
```

The Project Code: Data Handling Section

```
if i == 5:
    #Load data set values from CSV file
    csvData = "responses-6features.csv"

    #Array to store names of feature(s) and target feature
    names = ['Getting angry', 'Finances','Shopping centres', 'Branded clothing',
             'Entertainment spending', 'Spending on looks', 'Spending on gadgets']

    #Stores the data values with names of questions
    dataset = read_csv(csvData, names=names)

    #Stores array values into an array
    array = dataset.values

    #Initialization of two numpy X and y arrays
    X6 = array[:,0:7]
    y6 = array[:,0]

    #Function to split the question values into a training set where 20 percent of the
    # data set is used for validation
    (X_train, X_validation, Y_train, Y_validation) = train_test_split(X6, y6, test_size=0.20)
```

The Project Code: Data Handling Section

```
if i == 6:
    #Load data set values from CSV file
    csvData = "responses-7features.csv"

    #Array to store names of feature(s) and target feature
    names = [’Getting angry’, ’Finances’,’Shopping centres’, ’Branded clothing’,
             ’Entertainment spending’, ’Spending on looks’, ’Spending on gadgets’,
             ’Spending on healthy eating’]

    #Stores the data values with names of questions
    dataset = read_csv(csvData, names=names)

    #Stores array values into an array
    array = dataset.values

    #Initialization of two numpy X and y arrays
    X7 = array[:,0:8]
    y7 = array[:,0]

    #Function to split the question values into a training set where 20 percent of the
    # data set is used for validation
    (X_train, X_validation, Y_train, Y_validation) = train_test_split(X7, y7, test_size=0.20)
```

The Project Code: Model Creation and Prediction Section

```
#Section: Model Creation and Prediction------------------------------------

      #Sets the overall average accuracy to zero before each test
      # extra information is not used is the next test’s results
      MLPavg = 0

      #https://scikit-learn.org/stable/index.html
      #Initialization of the model to be used along with parameters to use

      #Time function to begin timer
      start_time = tm.time()

      for i in range(average):

          if iter_selection == 0:

              #Default MLPClassifier
              estimator_MLP = MLPClassifier(max_iter=1400)

          if iter_selection == 1:

              #MLPClassifier with tangent activation function
              estimator_MLP = MLPClassifier(activation=’tanh’, max_iter=1400)

          if iter_selection == 2:

              #MLP Classifier with logistic activation function
              estimator_MLP = MLPClassifier(activation=’logistic’, max_iter=1400)

          #Function to train the model with the data set
          estimator_MLP.fit(X_train, Y_train)

          #Prediction function for MLPClassifer
          prediction_MLP = estimator_MLP.predict(X_validation)

          #Determine the accuracy score of the model
          MLPavg = accuracy_score(Y_validation, prediction_MLP) * 100 + MLPavg
```

The Project Code: Model Creation and Prediction Section

```
#Stores the information of the first model
if iter_selection == 0:

    #Stores accuracy results of test into an array
    MLPdata1.append(float(MLPavg))

    #Stores time results of test in an array
    final_time1.append(float(stop_time))

#Stores the information of the second model
if iter_selection == 1:

    #Stores accuracy results of test into an array
    MLPdata2.append(float(MLPavg))

    #Stores time results of test in an array
    final_time2.append(float(stop_time))

#Stores the information of the third model
if iter_selection == 2:

    #Stores accuracy results of test into an array
    MLPdata3.append(float(MLPavg))

    #Stores time results of test in an array
    final_time3.append(float(stop_time))
```

The Project Code: Displaying Results Section

This section of the code is meant for displaying the results of the test with a MatPlotLib 3D Scatter plot

```
    #Section: Displaying Results-----------------------------------------------

    #Initialization of MatPlotLib model
    ax = plt.axes(projection=’3d’)

    #Plots MLPClassifier data
    ax.scatter3D(feature, MLPdata1, final_time1)
    ax.scatter3D(feature, MLPdata2, final_time2)
    ax.scatter3D(feature, MLPdata3, final_time3)
    ax.legend([’Default’,’Tangent’,’Logistic’])

    #Axis labels and figure title
    ax.set_xlabel(’Number Of Features’)
    ax.set_ylabel(’Average Accuracy Score’)
    ax.set_zlabel(’Time Taken (Seconds)’)
    title = ’MLPClassifier Average of ’ + str(average) + ’ Model(s)’
    ax.set_title(title)

    #Shows graph to screen
    plt.show()

#Main function call------------------------------------------------------------
main()
```