

1602835 Jake Russell EIGP

by Jake Russell

Submission date: 16-Mar-2020 01:21PM (UTC+0000)

Submission ID: 122094197

File name: 12019_Jake_Russell_1602835_Jake_Russell_EIGP_210055_2028148916.docx (412.43K)

Word count: 2091

Character count: 10499

EIGP ASSIGNMENT 2

Jake Russell

1602835

Table of Contents

Management & Operations	3
Team working and collaboration:	3
Skills:	3
Development.....	4
Python practice p.1:	4
Python practise p.2:	6
References	7
Appendix	8
Reflection	8
Networking	8
Team Progress sheets	9

Management & Operations

Team working and collaboration:

For a means of communication, the team mainly used email. The team made a group within Microsoft outlook called F.I.R.N.N (Facial image recognition neural network). Here they would discuss topics, send updates or ask questions about the team or project. In order to collaborate and exchange files the team used Microsoft teams. The during would often have team meetings in order to practice presentations or important pitches. In terms of effectiveness the email would usually be fine however sometimes team members would miss an email or two, this would usually be fine, as most emails were just small discussions and updates however the risk still stands that a member could miss important information. Using Microsoft teams however was a very effective means to collaborate, it allowed the team to easily edit and add to file and work uploaded to the group, for example being the initial pitch presentation, as each member could add their slides with ease.

Skills:

The overall skills needed for this potential tech start-up are as follows:

- A knowledge of machine learning
- Ability to code machine learning algorithms

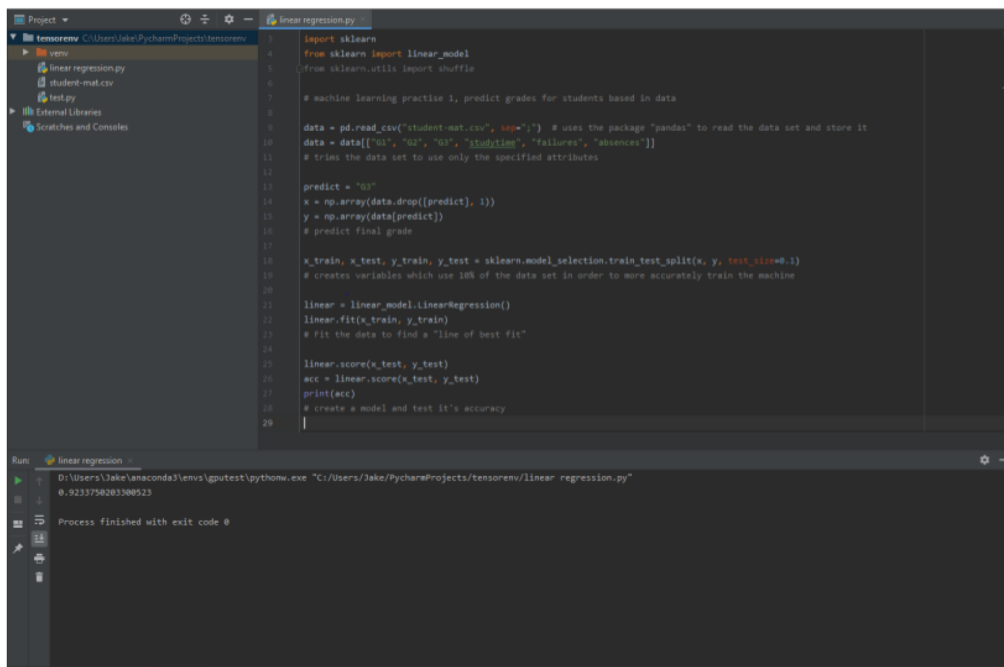
In terms of skill gaps, most members of the team did not have the knowledge or experience in machine learning or the necessary coding ability to create a neural network for the prototype. These were both solved via MatLab's services. MatLab has a deep learning course in which the developers could learn about the basics of machine learning and its implementation. This was great to increase the team's knowledge, but it is nowhere near enough for the team to build their own neural network from scratch. Thankfully MatLab has a multitude of pretrained networks available to use, this allowed to the team to take this pretrained network and mould it to work with faces.

Development

The technical skills the author of this report decided to develop were that of machine learning, however, rather than going for matlab to which would contribute to the final prototype, they instead focused on developing their python skills, as they believed that would be a better industry standard to learn in order to give them experience and a step up towards their preferred career path.

Python practice p.1:

The first thing the author went about was learning how to code a basic machine learning algorithm that takes a data set and uses linear regression in or to make predictions of future data of said data set. To start the author went and got a data set to use, in the end they used a data set to do with student performance from the UCI machine learning repository (1). The program will take a set number of attributes from the data set and make a prediction on the final grade of the student. The program will display the actual final grade saved in the data set as a comparison to judge the accuracy of the machine.



```
1 import sklearn
2 from sklearn import linear_model
3 from sklearn.utils import shuffle
4
5 # machine learning practice 1, predict grades for students based in data
6
7 data = pd.read_csv("student-mat.csv", sep=";") # uses the package "pandas" to read the data set and store it
8 data = data[["G1", "G2", "G3", "studytime", "failures", "absences"]]
9 # tries the data set to use only the specified attributes
10
11 predict = "G3"
12
13 x = np.array(data.drop([predict], 1))
14 y = np.array(data[predict])
15 # predict final grade
16
17 x_train, x_test, y_train, y_test = sklearn.model_selection.train_test_split(x, y, test_size=0.1)
18 # creates variables which use 10% of the data set in order to more accurately train the machine
19
20 linear = linear_model.LinearRegression()
21 linear.fit(x_train, y_train)
22 # Fit the data to find a "line of best fit"
23
24 linear.score(x_test, y_test)
25 acc = linear.score(x_test, y_test)
26 print(acc)
27 # create a model and test it's accuracy
28
```

Figure 1: Linear regression code p.1

The code above is the author's with comments to explain. First the author loads the data set which is saved as a CSV (comma separated variables) file and saves it as a variable called data. They then update the variable to have only the attributes to be used with the predictions, this being "grade 1, grade 2, study time and absences". The author chose these specifically as they were all integer-based attributes.

The author then made a variable called predict and assigned it the label "G3" as they want the program to determine what G3 is. The author then creates an array which is a new data frame which doesn't have G3 and will hold all the attributes, they then make an array which will hold all of the labels, which as it stands is only "predict". The author then creates variable to split the data which use 10 % of the data to test the machine so the machine is only trained in 90% of the data. This is so there is actual information the machine doesn't know for it to actually predict as supposed to already knowing the answer. The author then saves the linear model of linear regression as "linear" and then fits the data from "x_train" and "y_train" to find a line of best fit for the data. Finally, the author adds an accuracy test which will print a number from 0 to 1 representing the accuracy percent. The percent returned in the test above as shown was approximately 92%.

The screenshot shows a Jupyter Notebook with a file explorer on the left containing 'linear regression.py', 'Student-mat.csv', and 'test.py'. The main area displays Python code for linear regression using sklearn. The code includes data loading, splitting into training and testing sets, model fitting, and accuracy calculation. The output at the bottom shows a list of 20 data points, each with a predicted value and the actual value in brackets.

```

10 # tries the data set to use only the specified attributes
11
12
13 predict = "G3"
14 x = np.array(data.drop(predict, 1))
15 y = np.array(data[predict])
16 # predict final grade
17
18 x_train, x_test, y_train, y_test = sklearn.model_selection.train_test_split(x, y, test_size=0.1)
19 # creates variables which use 10% of the data set in order to more accurately train the machine
20
21 linear = linear_model.LinearRegression()
22 linear.fit(x_train, y_train)
23 # fit the data to find a "line of best fit"
24
25 linear.score(x_test, y_test)
26 acc = linear.score(x_test, y_test)
27 print(acc)
28 # create a model and test it's accuracy
29
30 predictions = linear.predict(x_test)
31 # take the array of arrays and make predictions based on the data
32
33 for x in range(len(predictions)):
34     print(predictions[x], x_test[x], y_test[x])
35     # print the predictions
36

```

Run: linear regression

```

4.317526159831047 [ 6 5 2 0 6 ] 6
9.84416580355565 [10 10 1 0 4] 10
9.183884893287768 [11 9 1 1 19] 10
11.795781896884205 [12 12 2 1 12] 13
8.395929871828282 [ 9 9 2 0 0] 10
10.76968115825472 [11 11 2 0 4] 11
12.7478781230888 [11 13 1 1 10] 13
11.257368863744087 [ 9 12 3 0 3] 11
11.838078033071518 [13 11 2 0 4] 11
9.18263236818845 [ 8 10 2 0 0] 9
8.682243635863328 [10 9 2 0 4] 11
18.337284644626965 [19 18 3 0 0] 19
8.618823857638992 [ 8 8 1 0 30] 8

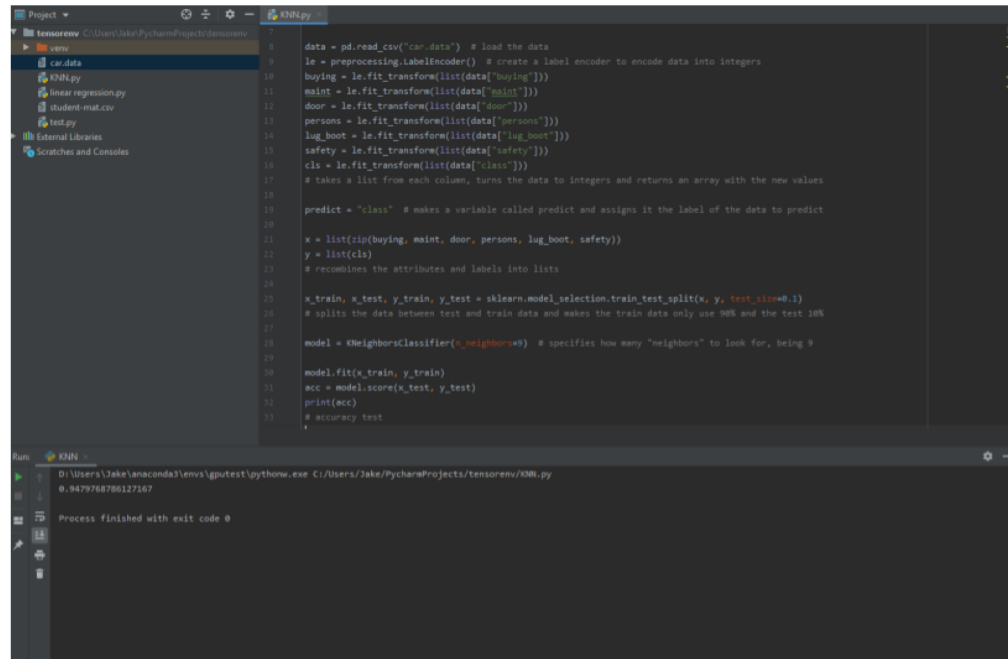
```

Figure 2: Linear regression code p.2

To finish the program the author wrote a line of code which takes the data and uses linear regression to make a set of predictions of the final grade, saving them to "predictions". Lastly, they make a for loop to print the predictions followed by the data used to make said prediction and then the correct final grade. As seen above the machine has been fairly accurate with it's predictions with a couple outliers such as the one where it predicted 9 to be the final grade where the final grade was in fact 0.

Python practise p.2:

The next skill the author went about developing was the K nearest neighbour algorithm, or KNN for short. The reason for this was because it's not a very difficult algorithm to start learning how to use but it's used quite often in machine learning. Once again to do this the author went and got a data set to use, this time they needed a data set that had some kind of class system, they ended up using a car evaluation data set from the UCI (2), the program will take the data and try and predict the car's class based on acceptability.



```
1 data = pd.read_csv('car.data') # load the data
2 le = preprocessing.LabelEncoder() # create a label encoder to encode data into integers
3 buying = le.fit_transform(list(data['buying']))
4 maint = le.fit_transform(list(data['maint']))
5 door = le.fit_transform(list(data['door']))
6 persons = le.fit_transform(list(data['persons']))
7 lug_boot = le.fit_transform(list(data['lug_boot']))
8 safety = le.fit_transform(list(data['safety']))
9 cls = le.fit_transform(list(data['class']))
10 # takes a list from each column, turns the data to integers and returns an array with the new values
11
12 predict = 'class' # makes a variable called predict and assigns it the label of the data to predict
13
14 x = list(zip(buying, maint, door, persons, lug_boot, safety))
15 y = list(cls)
16 # recombines the attributes and labels into lists
17
18 x_train, x_test, y_train, y_test = sklearn.model_selection.train_test_split(x, y, test_size=0.1)
19 # splits the data between test and train data and makes the train data only use 90% and the test 10%
20
21 model = KNeighborsClassifier(n_neighbors=9) # specifies how many "neighbors" to look for, being 9
22
23 model.fit(x_train, y_train)
24 acc = model.score(x_test, y_test)
25 print(acc)
26 # accuracy test
```

Run KNN

D:\Users\Jake\anaconda3\envs\gpu\test\python.exe C:/Users/Jake/PycharmProjects/tensorem/KNN.py
0.9479736786127367

Process finished with exit code 0

Figure 3:KNN code p.1

The code above is the author's implementation with comments. First the program loads the data from the file car.data. Then an encoder is made to make the string variables in the file integers in order to work within the program. This encoder is then applied to each column and returns arrays with the new values. Next a "predict" variable is made with the "class" label assigned to it as this is what the algorithm is going to predict. The attributes and the labels are then recombined into regular lists ready for the machine to train off them. The data is then split into train variables and test variables, where the train variables use 90% of the data and the test use 10%, this is done for the same reason as the previous program. Next a model variable is made which specifies how many "neighbours" to look for, in this case being 9. And accuracy test is then done in the same way as the previous program except by using KNN with the result being approximately 95% accuracy.

```

13 safety = le.fit_transform(list(data["safety"]))
14 cls = le.fit_transform(list(data["class"]))
15 # takes a list from each column, turns the data to integers and returns an array with the new values
16
17 predict = "class" # makes a variable called predict and assigns it the label of the data to predict
18
19 x = list(zip(buying, maint, door, persons, log_boot, safety))
20 y = list(cls)
21 # recombines the attributes and labels into lists
22
23 x_train, x_test, y_train, y_test = sklearn.model_selection.train_test_split(x, y, test_size=0.1)
24 # splits the data between test and train data and makes the train data only use 90% and the test 10%
25
26 model = KNeighborsClassifier(n_neighbors=4) # specifies how many "neighbours" to look for, being 4
27
28 model.fit(x_train, y_train)
29 acc = model.score(x_test, y_test)
30 print(acc)
31 # accuracy test
32
33 predicted = model.predict(x_test)
34 names = ["unacc", "acc", "good", "vgood"]
35 for x in range(len(predicted)):
36     print("Predicted: ", names[predicted[x]], "Data: ", x_test[x], "Actual: ", names[y_test[x]])
37 # create the prediction and the names for the predictions and output each prediction followed by the data used
38 # and then the actual answer
39
40
41

```

```

Run: KNN
D:\Users\Jake\anaconda3\envs\gputest\python.exe C:/Users/Jake/PycharmProjects/tensorem/KNN.py
0.9248554913294798
Predicted: good Data: (1, 3, 3, 1, 0, 0) Actual: unacc
Predicted: good Data: (0, 3, 1, 0, 1, 0) Actual: good
Predicted: good Data: (1, 1, 1, 1, 1, 1) Actual: good
Predicted: good Data: (1, 1, 2, 0, 2, 1) Actual: good
Predicted: good Data: (1, 3, 0, 2, 0, 0) Actual: unacc
Predicted: good Data: (2, 3, 2, 0, 1, 1) Actual: good
Predicted: good Data: (0, 0, 0, 0, 2, 2) Actual: good
Predicted: acc Data: (1, 1, 2, 2, 2, 0) Actual: acc
Predicted: unacc Data: (3, 1, 3, 2, 0, 2) Actual: unacc
Predicted: good Data: (2, 3, 0, 1, 0, 1) Actual: good
Predicted: good Data: (3, 2, 2, 0, 2, 2) Actual: good

```

Figure 4: KNN code p.2

In the same vein as before the author finishes by outputting the predictions followed by the data used and finally the actual answer. In terms of results the algorithm has worked very well with very few wrong predictions.

References

1. UCI. [Online] [Cited: 15 March 2020.] <https://archive.ics.uci.edu/ml/datasets/Student+Performance>.
2. UCI. [Online] [Cited: 15 March 2020.] <https://archive.ics.uci.edu/ml/datasets/Car+Evaluation>.

Appendix

Reflection

Looking back the author believes they have made good progress in developing the machine learning skills, they have learnt how to make both linear regression and K nearest neighbour which are both the building blocks for machine learning and should serve as a good foundation for future endeavours. Through learning these the author also improved on their python skills which to start were practically none-existent, while they still haven't learnt all the syntax, yet they've got a good grasp so far.

Networking

For networking the author attended this webinar: <https://www.techgig.com/webinar/Project-Management-Series-Session-1-Introduction-to-Project-Management-15>

Here they learnt about the methods and phases of project management. The phases being as follows:

- Initiate the project
- Do the project planning
- Execute the project
- Monitor and control
- Close the project

The webinar also gave an introduction into starting a career in project management, they listed the key aspects as follows:

- Understand requirement capturing, estimation, scheduling process
- Start managing small modules
- Work with people effectively
- Delegate with confidence
- Build good networking with and outside the organization
- Do not hesitate to work with difficult stakeholders
- Recruit right people and recognize their contributions
- Written and oral communication skills
- Are most important
- Report right information always

Team Progress sheets

Weekly Progress Report 1 -

Team Name: Facial/Image recognition neural network

Team Member Names & Roles:

AM: Amelia – Lead Developer

JR: Jake – Developer

S: Sai – Research Analyst

KA: Kieran - Project Manager

FR: Farhan - Business Analyst

SA: Samia – Business Analyst

Week commencing 28/10/19

Amelia

- Started writing notes about stakeholders on the shared “PID notes” document
- Started writing notes for Gantt chart/network diagram

Kieran

- Added to notes about stakeholders
- Tried to pick an order for the Gantt Chart/Network Diagram tasks

Weekly Progress Report 2

Currently unconfirmed team roles.

Date: 06/11/2019

In-Attendance: KA, JM, FR, S. Absent: JR, SA

Note: Can't currently assign Samia or Jake any tasks or team roles as we haven't had any group meetings with them.

Individual Updates

Update: Officially assigned some team roles

KA Update: Working on the Project Initiation Document, looking into the project planning tasks.
Reading and Researching Neural Networks

AM Update: Working on the Project Initiation Document, continuing to learn Matlab/looking at deep-learning basics.

Tasks Set by PM for the Week:

KA – Look and possibly make a draft of requirements for the project.

AM – Continue to research how we would develop our product, in particular the development side to it.

FR – Research the market, are there products like what we have thought of? Are they popular and have they proven useful?

S – Research and document findings around Neural Networks and what is the best approach we could take

SA – Help farhan with business side of things/research.

Task for all: All of us come up with 3 ideas for a Name and report back next week.

Weekly progress report 3

KA

AM – Finished matlab deep learning basics, starting advanced matlab deep learning. Not thought of idea for a name yet, some sort of acronym could work.

J – Deep learning basics

FR

S

SA – Started looking at the cost requirements and possible issues that may arise

Weekly progress report 4

KA – Managing group members

AM – Imported network into matlab

J –

FR – Market research

S - Deep learning research

SA – Business research

Goals for coming week:

Start building a set of images to train the network. Think about initial pitch

Weekly Progress Report 5

KA – Managing group members, started the presentation, assigned tasks for everyone for the presentation.

AM – Started idea section of presentation. Learning about deep learning training methods and how to monitor network training

J –

FR –

S -

SA –

We've now decided on a name for our group, the pitch has been started. I'd like you all to look at the section of the presentation that you have been assigned and add some notes to your section of the presentation, then update it on the Microsoft Teams group

Weekly Progress Report 6 Week commencing 3/2/2020

KA – Assigned Tasks for the group

AM – Continued practice with MATLAB deep learning toolbox. Started building set of images to train the network.

J –

FR –

S -

SA –

Weekly Progress Report 7 Week commencing 10/2/2020

In Attendance - Sai, Kieran, Amelia, Farhan

Absent – Jake and Samia

KA – Looking into the tasks for assignment 2 to assign roles for everyone

AM – Decided on ideation section for assignment 2

J –

FR – Chosen the Market and Competitors for assignment 2

S – Chose Tech start-up for assignment 2

SA –

Weekly Progress Report 8 Week commencing 17/2/2020

In Attendance - Sai, Kieran, Amelia, Farhan

Absent – Jake and Samia

KA –

AM –

J –

FR –

S –

SA –

Weekly Progress Report 9 Week commencing 24/2/2020

In Attendance - Sai, Kieran, Amelia, Farhan

Absent – Jake and Samia

KA –

AM –

J – Started training in Python Machine learning

FR –

S –

SA –

Weekly Progress Report 10 Week commencing 4/3/2020

In Attendance - Sai, Kieran, Amelia, Farhan

Absent – Jake and Samia

KA –

AM –

J – Completed linear regression algorithm program

FR –

S –

SA –

Weekly Progress Report 11 Week commencing 11/3/2020

In Attendance - Sai, Kieran, Amelia, Farhan

Absent – Jake and Samia

KA –

AM – Learning how to do object detection

J – Learning K nearest neighbour algorithm programs

FR –

S –

SA –

FINAL GRADE

GENERAL COMMENTS

Instructor

58/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

RESEARCH

Engagement with Literature Skills.



+ Citations: 2. Good. – could be extended to show how research has helped to increase knowledge and understanding. + References: 2. - could be extended to include additional sources.

4

KNOWLEDGE

Knowledge and Understanding Skills.



+ Team working and skill gap identification. Note: Good use of MS Teams. + Programming.

8

ANALYSIS

Cognitive and Intellectual Skills.



+ Team collaboration analysis. + Code analysis. Good clear narration to describe process and code. + Code comments.

13

PRACTICAL

Practical Application Skills.



+ Online Team working. + Python and Machine learning.

17

TRANSFERABLE SKILLS

Transferable Skills for Life and Professional Practice.



+ Good report presentation. – Format could be enhanced with section numbers. – Structure could be enhanced with an introduction and a conclusion. + Reflection. Good. + Event attendance. + Team progress sheets. – Some evidence of team contribution and collaboration documented, could be increased. Note: Evidence of using Teams taken into consideration, screenshots could be documented to enhance this area .

16