

# Instructions to Authors for the Preparation of Papers for the 18th Australian International Aerospace Congress

*Please select category below:*

Normal Paper

Student Paper

Young Engineer Paper

## Using K-Nearest-Neighbours (KNN) Machine Learning Technique to Classify Archived Helicopter Wear Debris Data

E. Lee

*Aerospace Division, Vehicle Dynamics and Diagnostics, Defence Science and Technology Group, PO Box 4331, Melbourne, Victoria, 3001,  
Australia*

### Abstract

Two decades ago, data reduction processes and techniques were common features in many advanced Health and Usage Monitoring Systems (HUMS) due to storage limitation and cost. Since then data storage volume and reduction in cost have increased exponentially. The drawback with this surge in cheap storage is many HUMS sensor data are now preserved in high resolution. For the past ten years excessive data has become a major challenge for HUMS and many other industries. For the last five years buzzwords such as Big Data, Data Fusion, and Data Analytics or Mining (e.g. Machine Learning, ML) are gaining momentum and seen by some as the Holy Grail that will resolve the huge data issues. This paper presents an investigation of using ML, in particular, the K-Nearest-Neighbours technique (KNN), with helicopter Wear Debris data to gauge the potential of ML as a Data Analytic tool.

**Keywords:** Data Analytic, Data Mining, Machine Learning, K-Nearest-Neighbour, Wear Debris Analysis

### Introduction

What to do with the archived Health and Usage Monitoring Systems (HUMS) data is becoming a major concern across many disciplines of the HUMS community. Unless the stored HUMS data can be analysed and turned into information, it is useless to the operator. The question is why the stored HUMS data are rarely being looked at? In the past, sensor availability and data resolution had restricted the use of HUMS data. However, in recent years data explosion caused by a sensor rich environment coupled with drastic increase in data storage capacity has becoming the limiting factor of the frequent use of HUMS data. Owing to the labour intensive nature of analysing vast amount of HUMS data, the only time these data are being looked at is during major events such as an accident or incident investigation.

As HUMS is becoming an integral part of newer platforms (such as F-35) the amount of preserved HUMS data will only intensify. As of today, the amount of accumulated HUMS

data is already reaching a critical limit. Once this threshold is exceeded the amount of data will be humanly impossible to go through.

Massive data problems are not only restricted to Aerospace industry. Industry sectors such as Banking, Astronomy, Ecology, or Meteorology etc. are all facing the issue of extreme data. In dealing with this data problem, buzzwords such as Big Data, Data Fusion, and Data Analytics (e.g. Machine Learning, ML) are gaining traction in recent years. Astronomy is at the forefront of the ML application. The sheer amount of astronomy data that needs to be analysed is already beyond human comprehension. Currently there is extensive literature in astronomy describing the use of ML to detect anomalies such as gravitational waves [1] or black holes [2].

Data analytics, especially the automated version of data analytics such as Machine Learning, is a relatively new buzzword in Aerospace community. Many sectors of the aerospace community are now claiming the application of ML; however, only very few tangible examples have been shown so far. This study attempts to explore the use of ML, especially the K-Nearest-Neighbours (KNN) technique, with aircraft data such as Helicopter Main Gearbox (MGB) Wear Debris Analysis (WDA) data. The aim is to gauge whether ML has the potential to assist with large data issues or is just another buzzword that eventually fades away.

## **Machine Learning (ML)**

### **What is Machine Learning?**

“Machine Learning is the science of getting computers to learn and act like humans do, and improve their learning over time in autonomous fashion, by feeding them data and information in the form of observations and real-world interactions [3]”. ‘Supervised’ and ‘Unsupervised’ learning are the two most widely adopted machine learning methods [4]; however, other methods such as ‘Semi-supervised’ and ‘Reinforcement’ learnings are also used in some circumstances [5]. The KNN algorithm explored in this paper is a supervised ML technique where features of learning and label or class for all features need to be nominated before learning.

### **K-Nearest-Neighbours (KNN)**

The K-Nearest-Neighbours is a simple classification algorithm that is commonly referred to as an entry level ML technique [6]. KNN is an easy algorithm to understand and to implement [7], and in addition very versatile and with high accuracy, it was selected in this study to gauge the ML potential for aerospace applications.

KNN is a non-parametric learning algorithm. Non-parametric means not making any assumptions on the underlying data distribution, hence the dataset could be linear or non-linear. This algorithm essentially classifies data by finding the most similar data points in a training dataset and groups them accordingly [7] as demonstrated in Figure 1. Once the entire dataset has been classified, with the learned patterns, this knowledge is then tested in the test dataset for accuracy. When a classification is needed for an unseen data point, KNN will use the learned knowledge to find instances or data points (K points) closest to the unseen data point. Once K points are identified, a voting process will decide which label (or class) group the unseen data point belongs to.

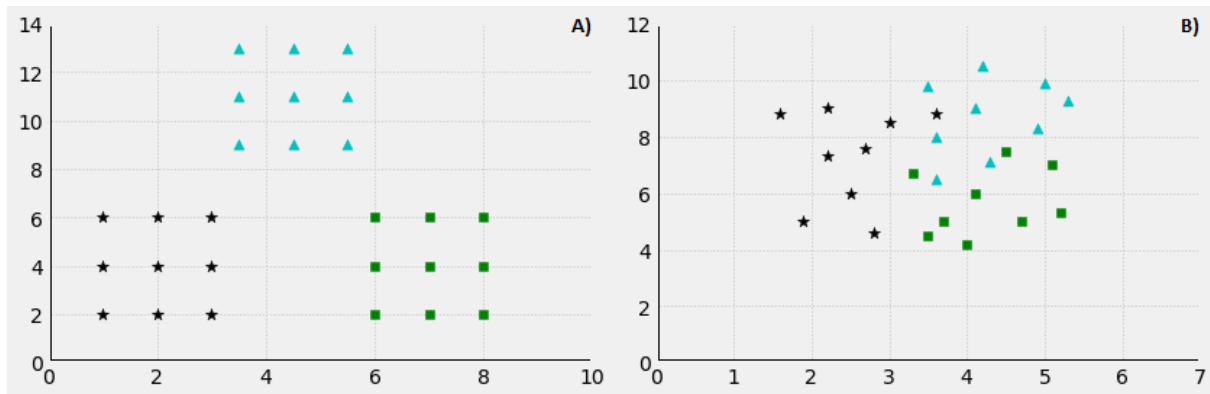


Fig. 1: A) Linear dataset processed by KNN, and B) Non-linear dataset processed by KNN

## K points

Selecting a good value for K (number of nearest points to the unseen data point) is a trial and error process. A number of ways to calculate K has been suggested in references [8, 9, 10, 11], but due to the non-parametric nature of the dataset it is difficult to have one single best formulation for calculating K.

Finding a good K value is important because when a very large dataset is to be classified and a small K was chosen, this increases the risk of overfitting. For instance, a K value of 3 was selected for a very large dataset. There are reasonable chances that noisy data points are close enough to each other to outvote the correct data points in some parts of the dataset. By the same token, if a large value K was chosen for a small dataset the effect of over smoothing can occur. Over smoothing eliminates some of the important aspects about the dataset and could result in inaccurate classification. In short, a good K value is one that is large enough to avoid overfitting and small enough to avoid over smoothing the distribution of the dataset [9].

Although there is not a single best way of calculating K value, there is some general guidance for what K value should not be. The K value must be larger than the total number of the label (or class) types. If there are a total of four different labels used in KNN, but K equals to 3 was selected this means at least one label was always excluded during the voting process. The other general rule is, the K value must be an odd number to avoid confusion between two labels of data (basically to avoid tie or draw situation during voting) [10].

## Nearest distances

Once the number of K points has been decided, the next step is to identify the K points closest to the unseen data point. The way to determine the closest points to the unseen data point is by distance. For KNN, Euclidean, Minkowski, and Manhattan are methods that can be used to calculate the nearest distances [10, 11]. Euclidean distance method is the one typically used in KNN technique [12]. In general, the distance between points x and y in a Euclidean space  $R^n$  is presented as follow [13]:

$$d = |x - y| = \sqrt{\sum_{i=1}^n |x_i - y_i|^2} \quad (1)$$

Figure 2 shows examples of K equals five (blue circle) and the distances (red arrows, Euclidean method) of the nearest five data points to the unseen data point (red cross) for both, linear and non-linear examples in Figure 1. In Figure 2-A (linear dataset), four out of the five nearest points are triangles (rest is a star) therefore by majority vote the unseen data point belongs to the triangle group. In Figure 2-B (non-linear dataset), three out of the five nearest

points belong to the triangles group (the rest are one star and one rectangle) by majority vote the unseen data point belongs to the triangles group.

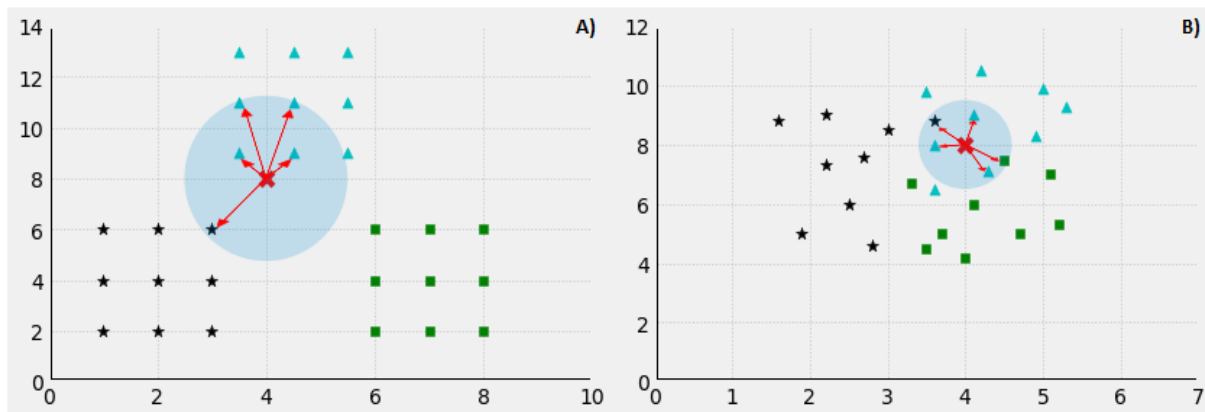


Fig. 2: Unseen data point (Red Cross),  $K$  equals 5 (Blue Circle), and nearest distances (Red Arrows)

## Pros and Cons of KNN

The KNN technique has clear advantages but also some disadvantages. The following bullet points categorise pros and cons of the KNN:

Pros [15, 16, 17]:

- Simple and powerful
- Works with multiclass datasets
- New training examples can be added easily
- Non-parametric nature (can easily handle dataset that may be highly ‘unusual’)
- The decision boundaries can be of arbitrary shapes

Cons [15, 16, 17]:

- To determine the nearest neighbour of an unseen data, KNN must compute the distance to all data points in the training dataset. Runtime performance will be slow if database is extremely large (in this case other ML techniques should be used).
- Prone to skewed class distributions or irrelevant attributes
- Need to determine a practical  $K$  value

## Wear Debris Analysis (WDA)

### ADF WDA Laboratory

The Australian Defence Force (ADF) WDA laboratory (lab) is a joint lab between Defence Aviation Safety Authority (DASA) Engine Structural Integrity (ESI) and Defence Science and Technology Group (DST Group). This lab was established to help determine aircraft serviceability and augment any existing condition monitoring programs by providing detailed analysis of wear debris retrieved from aircraft engines, gearboxes, and other mechanical systems. This lab provides free services for ADF aviation assets. This lab is not an ‘Oil’ analysis lab, but is for determining the size, quantity, morphology, and elemental composition of the debris that then enables the origin of the debris to be determined. The origin (e.g. bearing housing material, gear steel, etc.) can then be used to make an informed maintenance or serviceability decision.

### WDA reports

Since the creation of the ADF WDA Lab, there has been a large depository of WDA reports from both fixed and rotary wing ADF air assets. Each WDA report follows a pre-defined template as shown in Figure 3. The size, quantity, shape and features input in the report follow the American Society for Testing and Materials (ASTM) international 7898-14 standard. The Material composition can be either determined using a Scanning Electron Microscope (SEM) or with an X-ray Fluorescence (XRF) [14]. Typical source of debris is generally determined using the Original Equipment Manufacturer (OEM) supplied metal map once size, quantity, morphology, and material composition of the debris is known. In terms of ML, it is pretty obvious that size, quantity, morphology, and material composition are the ‘Features’ and Typical Source is the ‘Label’ (or class) for learning.

Attention	Sample Date
Aircraft	Report Date
Component	Serial No
TSN	Comments
ADF-WDA Lab Ref	Objective Ref
References: A. ASTM 7898-14 Standard Practice for Lubrication & Hydraulic Filter Debris Analysis for Machinery Condition Monitoring B. Metal Map	

Location	Quantity <sup>1</sup>				Shape and Features <sup>2</sup>	Material <sup>3,4</sup>	Typical Source (Ref. B)
	Extreme	Large	Few	Small			
Size Range	100-250 µm						
	250-500 µm						
	500-1000 µm						
	1000+ µm						
	Total						

Fig. 3: ADF WDA Lab report template

### WDA database creation

In this study helicopter MGB WDA reports were used to construct the database needed for the ML application (in particular the KNN). Each WDA report equates to an entry in the database and in total 307 entries is created. A portion of the helicopter MGB WDA database is presented in Figure 4 where Feature (red) and Labels (blue) are highlighted.

<b>Size:</b> Very Large (1000+ µm) = 4, Large (500-1000 µm) = 3, Medium (250-500 µm) = 2, Small (100-250 µm) = 1	<b>Quantity:</b> Extreme (100+ Particles) = 4, Large (26 to 100 Particles) = 3, Few (6 to 25 Particles) = 2, Small (1 to 5 Particles) = 1	<b>Curlicue</b> = 1, <b>Flake (silver)</b> = 2, <b>Chunk</b> = 3, <b>Strand (wire)</b> = 4	<b>% Ferrous debris (SEM, EDS, and XRF):</b> Titanium (Ti), Vanadium (V), Chromium (Cr), Manganese (Mn), Iron (Fe), Nickel (Ni), Molybdenum (Mo), Silicon (Si), Cadmium (Cd)																	<b>% Non-Ferrous debris:</b> Silver (Ag), Chlorine (Cl), Tantalum (Ta), Calcium (Ca) Carbon (C), Sulfur (S), Phosphorus (P), Copper (Cu), Lead (Pb), Tin (Sn), Magnesium (Mg), Antimony (Sb), Zinc (Zn), Aluminium (Al)											<b>From System Metal Map:</b> Bearing cage metal = 1, Gear steels = 2, Silver plating metal = 3, Not part of Metal Map = 4
Size	Quantity	Shape (Morphology)	Ti	V	Cr	Mn	Fe	Ni	Mo	Si	Cd	Ag	Cl	Ta	Ca	C	S	P	Cu	Pb	Sn	Mg	Sb	Zn	Al	Typical Source					
4	2	2	0.00	0.30	2.80	0.00	95.40	0.00	0.30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1				
2	1	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2				
2	3	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3				
4	2	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3				
4	1	1	19.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4				
3	1	1	0.00	0.80	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1				

Fig. 4: Helicopter MGB WDA database with ‘Features’ and ‘Labels’ indicated

### Data quarantine

Three data entries from the database were quarantined from ML training and testing. These three entries were used as the unseen data as they were totally isolated from ML. These isolated entries served as the final verification and confidence building for the KNN application to the helicopter MGB WDA database. If the KNN has done its learning appropriately, it is expected that the KNN will be able to identify labels 3, 2, and 1 based only on the ‘Features’ of the three isolated entries. Details of the three isolated entries are shown in

Figure 5. It is worth mentioning that the sequence of the designation 3, 2, and 1 does not represents any significance; it is purely for labelling only. Figure 6 shows actual Python codes for features (only features no label) of the three quarantined data entries (acted as the unseen data points) that were fed into the Python KNN for their label identification.

Size	Quantity	Shape (Morphology)	Ti	V	Cr	Mn	Fe	Ni	Mo	Si	Cd	Ag	Ci	Ta	Ca	C	S	P	Cu	Pb	Sn	Mg	Sb	Zn	Al	Typical Source	
<i>Silver is commonly used as a plating on bearing cages</i>																											
2	1		2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3
<i>EX53 is used for pinions of MGB</i>																											
4	1		2	0.00	0.00	1.10	0.50	87.70	2.10	5.10	1.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2
<i>AISI 4130 steel is found in nuts, liners, miscellaneous components, cages of most critical bearings</i>																											
4	1		4	0.00	0.00	1.10	0.80	98.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1

Fig. 5: Three quarantined entries with 3 = Sliver plating material, 2 = Gear steel, and 1 = Bearing gear steel

```
'''3 Silver is commonly used as a plating on bearing cages but is not explicitly mentioned in the metal map'''
Checksamle=[2,1,2,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,100.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00]
vote = k_nearest_neighbour(train_set, Checksamle, 5)
print('Sample is ID. as %d using KNN method!' %(vote))

'''2 EX53 is used for pinions of MGB'''
Checksamle=[4,1,2,0.00,0.00,1.10,0.50,87.70,2.10,5.10,1.20,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,2.30,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00]
vote = k_nearest_neighbour(train_set, Checksamle, 5)
print('Sample is ID. as %d using KNN method!' %(vote))

'''1 AISI 4130 steel is found in nuts, liners, miscellaneous components, cages of most critical bearings'''
Checksamle=[4,1,4,0.00,0.00,1.10,0.80,98.10,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00]
vote = k_nearest_neighbour(train_set, Checksamle, 5)
print('Sample is ID. as %d using KNN method!' %(vote))
```

Fig. 6: Input features to KNN

## Database split for training and testing

A common practice in ML is to randomly split the entire database into two parts; one part for training and the other for testing. The ratio of the split varies from 70-80% for training and 20-30% for testing. Amazon, a major international company, uses 70% training and 30% testing for their Machine Learning [18]. However, references [19, 20, 21] all suggested 80% and 20% data split is a good starting point.

For the helicopter MGB WDA database (total of 304 entries without the 3 quarantined entries), it was randomly split with 80% for training and 20% for testing. The whole process was repeated 25 times. The average test accuracy over 25 runs was obtained in the end.

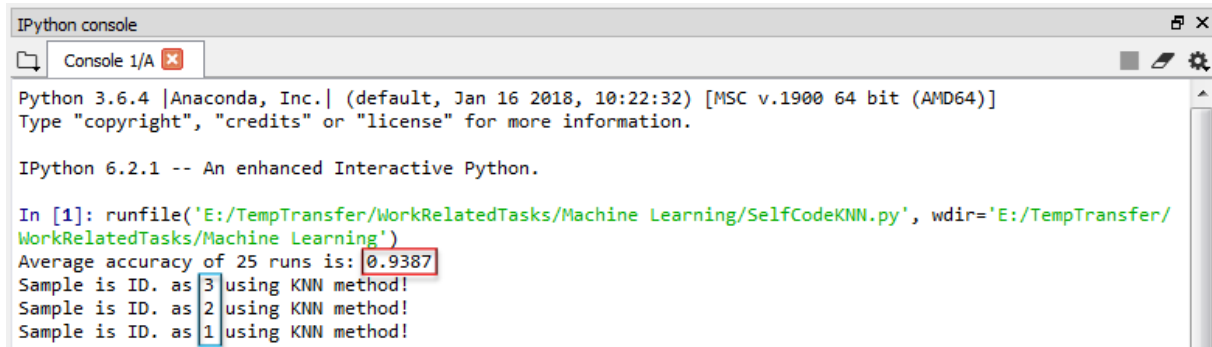
## Results

The KNN ML was performed using Python programming. The split of the database, the training, and the testing were all done within Python. In Python there are a number of libraries required when conducting ML; each library has its own function and purpose. For ML in Python the library that can perform KNN is called Scikit-Learn. This library provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python [22]. The function code for KNN in Python is 'neighbors.KNeighborsClassifier()'.

For comparison purpose, KNN using Scikit-Learn and KNN programming codes constructed from ground-up (based on KNN formulations) were tested on the helicopter MGB WDA database. The results obtained are almost identical (the actual comparison is not presented in this paper due to very small delta). Therefore, for convenience and efficiency KNN from Kcikit-Learn library should be utilised.

As shown in Figure 5, labels for the three quarantined data entries are 3, 2, and 1 respectively. If the ML KNN has the right features and labels to train during the training process, as well as correctly learned the patterns during classification, then the labels identified by the KNN

should be 3, 2, and 1. Figure 7 shows the KNN outputs where label 3, 2, and 1 are correctly identified by the KNN and highlighted in the blue vertical rectangular box in Figure 7. The KNN program was run 25 times and achieved an average test accuracy (from the 20% of database split for testing) of 93.87% as highlighted in the red horizontal rectangular box in Figure 7.



```

Python 3.6.4 |Anaconda, Inc.| (default, Jan 16 2018, 10:22:32) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 6.2.1 -- An enhanced Interactive Python.

In [1]: runfile('E:/TempTransfer/WorkRelatedTasks/Machine Learning/SelfCodeKNN.py', wdir='E:/TempTransfer/
WorkRelatedTasks/Machine Learning')
Average accuracy of 25 runs is: 0.9387
Sample is ID. as 3 using KNN method!
Sample is ID. as 2 using KNN method!
Sample is ID. as 1 using KNN method!

```

Fig. 7: KNN results for the three quarantined helicopter MGB WDA entries

## Conclusion

The ML KNN technique was applied to helicopter MGB WDA database. This example provided useful insight to the application of ML on real data. As KNN was able to correctly identify the labels for the three quarantined data entries, it has demonstrated a great potential for aerospace applications, in particular the automated classification of wear debris data. A number of different aerospace applications have been planned at DST group to further examine the KNN capability. As KNN can only comfortably handle data sizes in gigabytes, for larger data sets other ML techniques such as Support Vector Machines (SVM) or Deep Neural Net Work, etc., will need to be explored.

## References

1. Daniel, G., and Eliu, H., "Deep Learning for real-time gravitational wave detection and parameter estimation: Results with Advanced LIGO data", ELSEVIER, Volume 778, 10 March 2018, pp. 64-70
2. Huppenkothen, D., "Classifying Black Hole States with Machine Learning", American Astronomical Society, AAS Meeting #231, January 2018, id. 225.02
3. Daniel, F., "What is Machine Learning", <https://www.techemergence.com/what-is-machine-learning/>, accessed October 2018
4. SAS, "Machine Learning – What it is and why it matters", [https://www.sas.com/en\\_au/insights/analytics/machine-learning.html](https://www.sas.com/en_au/insights/analytics/machine-learning.html), accessed October 2018
5. Expert System, "What is Machine Learning? A definition", <https://www.expertsystem.com/machine-learning-definition/>, accessed October 2018
6. Towards Data Science, "Introduction to k-Nearest-Neighbors", <https://towardsdatascience.com/introduction-to-k-nearest-neighbors-3b534bb11d26>, accessed October 2018
7. Machine Learning Mastery, "Tutorial To Implement k-Nearest Neighbors in Python From Scratch", <https://machinelearningmastery.com/tutorial-to-implement-k-nearest-neighbors-in-python-from-scratch/>, accessed October 2018
8. Stackoverflow, "Value of k in k nearest neighbour algorithm", <https://stackoverflow.com/questions/11568897/value-of-k-in-k-nearest-neighbor-algorithm>, accessed October 2018

9. The Shape of Data, “K-Neares Neighbors”, <https://shapeofdata.wordpress.com/2013/05/07/k-nearest-neighbors/>, accessed October 2018
10. Quora, “How can I choose the best K in KNN (K nearest neighbour) classification?”, <https://www.quora.com/How-can-I-choose-the-best-K-in-KNN-K-nearest-neighbour-classification>, accessed October 2018
11. Scikit learn, “sklearn.neighbors.KNeighborsClassifier”, <http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>, accessed October 2018
12. Peterson, L. E., “K-nearest Neighbor”, Scholarpedia, Volume 4, Number 2, 2009, pp. 1883
13. WolframMathworld, “Distance”, <http://mathworld.wolfram.com/Distance.html>, accessed October 2018
14. Andrew, B., and Peter, S., “Improvements to Filter Debris Analysis in Aviation Propulsion Systems”, DSTO report, DSTO-TR-2773, December 2012
15. Izabela, M., Evangelos, P., and Dirk, H., “K-Nearest Neighbour Classifier”, ETHzurich, <https://www.ethz.ch/content/dam/ethz/special-interest/gess/computational-social-science-dam/documents/education/Spring2015/datascience/K-Nearest-Neighbour-Classifier.pdf>, accessed October 2018
16. Kevin, Z., “A Complete Guide to K-Nearest-Neighbors with Applications in Python and R”, <https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/>, accessed October 2018
17. Cynthia, R., “K-NN”, MIT, [https://ocw.mit.edu/courses/sloan-school-of-management/15-097-prediction-machine-learning-and-statistics-spring-2012/lecture-notes/MIT15\\_097S12\\_lec06.pdf](https://ocw.mit.edu/courses/sloan-school-of-management/15-097-prediction-machine-learning-and-statistics-spring-2012/lecture-notes/MIT15_097S12_lec06.pdf), accessed October 2018
18. Amazon Machine Learning, “Splitting the Data into Training and Evaluation Data”, <https://docs.aws.amazon.com/machine-learning/latest/dg/splitting-the-data-into-training-and-evaluation-data.html>, accessed October 2018
19. ResearchGate, “Is there an ideal ratio between a training set and validation set? Which trade-off would you suggest?”, [https://www.researchgate.net/post/Is\\_there\\_an\\_ideal\\_ratio\\_between\\_a\\_training\\_set\\_and\\_validation\\_set\\_Which\\_trade-off\\_would\\_you\\_suggest](https://www.researchgate.net/post/Is_there_an_ideal_ratio_between_a_training_set_and_validation_set_Which_trade-off_would_you_suggest), accessed October 2018
20. Beyond The Lines, “How to split a dataset”, <https://www.beyondthelines.net/machine-learning/how-to-split-a-dataset/>, accessed October 2018
21. Nishank, K. S., “Splitting CSV Into Train and Test Data”, <https://medium.com/themlblog/splitting-csv-into-train-and-test-data-1407a063dd74>, accessed October 2018
22. Jason, B., “A Gentle Introduction to Scikit-Learn: A Python Machine Learning Library”, Machine Learning Mastery, <https://machinelearningmastery.com/a-gentle-introduction-to-scikit-learn-a-python-machine-learning-library/>, accessed October 2018