Assignment 1 Data Mining Techniques

Christos Petalotis[1 [2739394]], Julia Wesselman[2 [2715996]], and Matteo De Rizzo[3 [2749303]]

Vrije Universiteit Amsterdam, 1081 HV Amsterdam, Netherlands
c.petalotis@student.vu.nl
m.de.rizzo@student.vu.nl
j.a.wesselman@student.vu.nl

Introduction

In this research paper different Data Mining techniques are investigated. First, a small dataset, created by students from the Data Mining Course, is explored. Second, we participated in competition using data from the titanic. Last, some theoretical aspects from the Data Mining techniques are discussed.

1 Explore a small dataset

1.1 Exploration

In this section, exploratory data analysis of the provided ODI Dataset is performed and some important findings are gathered and discussed. The data is comprised of 17 attributes, 16 of which contain the values of the answers to a questionnaire filled out by a total of 304 students who follow the Data Mining Techniques course. The dataset also includes the timestamp at which each person completed the survey, which is the 17th attribute of the dataset. Some of the questions had predefined answers from which one would have to select a value, while others could be answered using free text. There are numerous instances in which free text questions were left blank, as well as others in which the answer did not respond to the question, rather to something arbitrary, thus making such entries unusable. Furthermore, not every question provides useful or insightful information and therefore a selection took place before the analysis of the data was performed.

The provided dataset was transformed before some conclusions could be driven by the data in it. This was necessary as missing or "junk" values, that deviate from the appropriate type of answer to a specific question, wouldn't allow to infer useful conclusions for the data.

Out of the 16 total attributes related to student answers, only 6 were deemed important for analysis. This analysis is done per-question below:

1. What program are you in?
   The students that participated follow different master's programs, and the distributions of them has been investigated. We started by grouping and counting all the answers to this question into 8 different programs, with related answers ending up on the same list. Then we plotted the percentage of the total number of data that each program takes. This distribution is visible in the form of a pie chart in figure 1. Even though a few large groups account for the majority of the entries, some instances seldom appeared. If they failed to reach the threshold of five appearances they were assigned to a bulk attribute defined as 'other'. Looking at the chart we can see that the main program that students of the Data Mining Techniques course attend are Artificial Intelligence (37.5%) and Computer Science (12.2%).

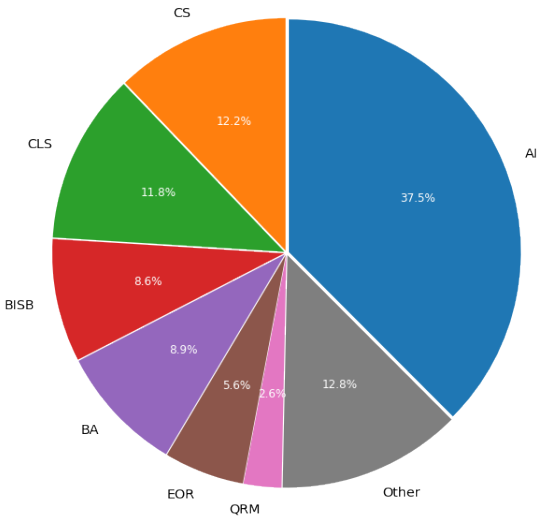

Fig. 1. Answers to the question "What program are you in?"

2. What is your stress level (0 -100)?
   This attribute had a specification as for the requested answer format. Nonetheless, many of the answers did not follow the instructions and were invalid. When an exaggerated answer, either upward or downward, was found, it was replaced with the lower or higher limit of the requirement ('0' or '100'). When wrong formatted answers were given, the value was replaced with the average of the acceptable entries. The cleaned data showed a standard deviation of 31.29 and a mean of 48.49.

3. Time you went to bed yesterday
   In this question, no predetermined format was required to answer, which resulted in a wide array of responses. After cleaning and properly formatting the data we found that the average time that people indicated went to sleep the previous day of the survey was 09:56:10 am.

4. Have you taken a course on statistics\databases\machine learning?
   The background knowledge of each student of the Data Mining course was also established. Firstly, the attendance at a statistical course was looked upon. The possible answers to this question were to this question are 'mu' (i.e. 'yes'), sigma (i.e. 'no'), and unknown. Similarly to statistics, the background knowledge of databases was asked. The possible answers were 'ja' (i.e. 'yes'), 'nee' (i.e. 'no') and unknown. Additionally, whether or not a course on machine learning was taken was looked at, with possible answers being 'yes', 'no', and 'unknown'.

Figure 2 exhibits the proportions of the answers for each topic in a chart. It shows that the vast majority of students have taken a statistics course before. Moreover, the participants were almost equally divided into yes and no when it comes to knowledge of databases, thus indicating that the overall level of familiarity of the students with database subjects is not strong. Regarding the familiarity with the topic of machine learning, only a third of the students indicated they had already taken a relevant course.

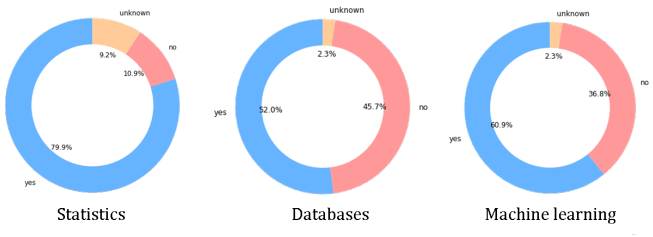

Fig. 2. Familiarity with different knowledge topics

An interesting realization came up when we looked at the data about the indicated stress level and time that an individual declared that they went to bed. In average, the later someone went to bed, the higher level of stress they indicated to feel. Particularly, after cleaning the provided answers, we split each instance in the dataset to one of the 3 catergories that we generated for the data related to stress. These categories are "Low Stress", "Moderate Stress" and "High Stress". Then for each category we averaged, using the mean metric, the declared sleep times of the instances that belonged to this particular group. The results were 10:40 pm for low stress, 10:44 pm for moderate stress and 08:25 am for high stress levels.

1.2 Basic Classification/Regression

For this task we downloaded an insurance dataset from kaggle.com. The dataset contains the attributes 'age', 'sex', 'bmi', 'children', 'smoker', 'region' and 'charges' for a total of 1338 people. Some of those attributes contain numerical values, while others are categorical.

The goal of the experiment is to predict the total charges that an individual would be assigned by their insurance, based on the values for the attributes that are available to us in the dataset. Since the target value that we want to estimate consists of a numerical attribute, regression needs to be performed to make predictions. For this reason, two algorithms, linear regression (LR) and random forest regressor (RFR) were used to make predictions. Moreover, the

categorical attributes of the dataset were transformed to binary ones, using one-hot encoding, to allow the regression models to better integrate them into their predictions.

In order to predict the performance of a specific model, a process called Cross Validation was utilized. Firstly, we split them into training and test sets, with a size of 70% and 30% of the total dataset respectively. Then, to find the optimal hyperparameters for each model, we performed a grid search for different combinations of parameters, applicable to each mode, that used cross-validation to find the best possible values for those parameters. The number of folds that were used is 5, as this leads to less bias and variance in the results [13].

When this process was complete, we extracted the suggested parameters. Next, we trained each model on the training set, using the hyperparameters suggested by the grid search operation. Once this was done, we used the trained models to estimate the charges that each individual would have to pay for their insurance, when some characteristics that define them are taken into consideration. Lastly, we measured the performance of each regression model that was used using the mean squared error. This metric for LR was 36713702.43 while for RFR was 20075737.43, leading us to believe that the RFR better predicted the charges for each individual, as the error from the actual values was smaller. Therefore, we can say the random forest regressor model built better understand the dataset used and is superior to the linear regression model we have when it comes to predicting a person's potential insurance charges.

2 Kaggle Competition to predict titanic survival

2.1 Data Exploration and Feature Engineering

Many factors could have affected the survival rate of the passengers of the Titanic. The datasets available to us, training and test, give us information on various attributes of those passengers. Some of them are categorical, such as the features Survived, Pclass (Passenger Class), Sex, and Embarked, while others have discrete values, such as PassengerId, Name, Age, SibSP (Siblings - Spouse), Parch (Parents - Children), Ticket, Fare, and Cabin. There are 12 common features in each dataset. As the values of the dataset attributes may provide an indication of the importance of each one of them in determining the survival of a passenger, it can be useful to explore some descriptive statistics and distributions for them.

The Titanic dataset comprises 891 passenger entries, of which 342 survived and 549 died, leading to a 38 % survival rate for the people on board. Looking at the sex of the passengers of the ship, we realize that the survival rate of the 314 women on board was almost 4 times higher than the one for the 577 men, as the survival rates were 74.2% and 18.8% respectively.

An important factor in determining the passengers that survived the Titanic accident was also the class of each passenger, as people with passenger class 1 survived at a 62.9 % rate, while the survival rates for classes 2 and 3 were 47.2%

and 24.2% respectively. It should be noted that class 3 was the most popular among the passengers, comprised of 491 people, yet this class demonstrated the lowest percentage of survivors.

Important conclusions can be also drawn when looking at the age of the Titanic passengers. When combined with the feature of sex, we can see that most young males and females of all ages survived the accident (see figure 3). At the same time, passengers of class 1 demonstrated greater survival rates, when compared to other classes, irrespective of age, while younger ages had higher survival rates in classes 2 and 3 than the rest of the ages in their class (see figure 4). 177 values are missing in the Age column of the training dataset and 86 values missing in the test dataset.

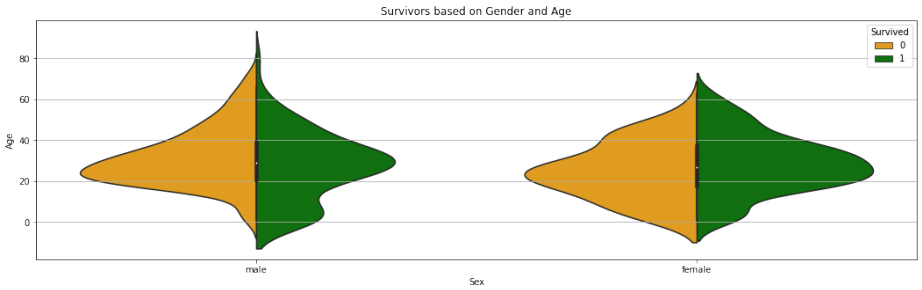

Fig. 3. Violin plot of survivors considering the Gender and Age attributes

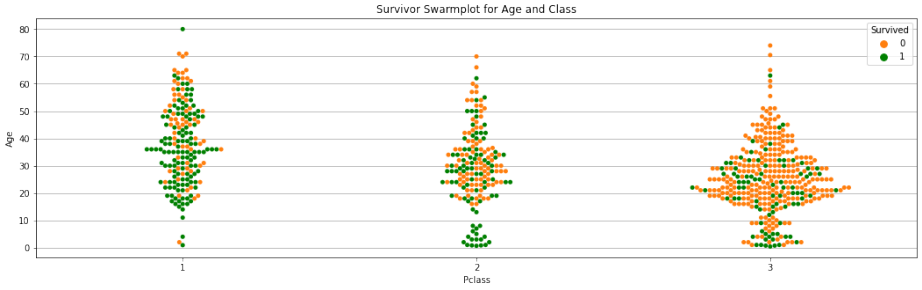

Fig. 4. Swarmplot of survivors considering the Passenger Class and Age attributes

Regarding the fare that passengers paid to onboard the Titanic, the mean price was 32.2$ while the max price was 512$ and the lowest one was 0$. Its distribution is left-skewed, not a normal one, with most of the fare prices being in the range of 0-100$.

Furthermore, it is important to mention that passengers that embarked on the Titanic from Cherbourg had the highest survival rate, with a value of 55.3%, while passengers from Southampton, which were the majority, with 644 of them being on board, survived at a rate of only 33.6%. This can be explained mostly by the fact that most of the passengers coming from Cherbourg were class 1 passengers, while the most popular class among the Southampton passengers was 3, which, as mentioned earlier, had a lower survival rate than the other classes.

Lastly, looking at the data, it can be noticed that most of the people on board were traveling aboard and that a small company increased the chances of survival when compared to people with a larger number of companions or people that were traveling alone.

It can be concluded that females and younger aged people had a higher survival rate than the rest of the passengers, indicating that priority was given to them while evacuation procedures were taking place at the time of the Titanic accident. Furthermore, all passengers with a fare above 500$ survived irrespective of gender, and all females that paid at least 200$ for the fare survived; this was not the case for the male passengers of the ship.

After this exploration of the provided datasets, some transformations were performed to prepare the data for use in the selected models. First of all, the attribute PassengerID was removed as it does not provide any valuable information. Then, the null values of the Age column were filled for each case with the median age of a passenger's group, when a group is formed by taking sex, passenger class, and title into consideration. This is done as each group may have a different median value. The median metric is used instead of the mean age because it is more robust to outlier values. Also, the null values for Fare were replaced using the mean fair price, missing Embarked values were filled with the most frequent value of this attribute and missing Cabin values were replaced with U, for unknown. After this, we used the passenger names to extract the titles that are included in them, to infer the social status of each one of them, which may lead to different survival rates based on high or low status. When this was done, we dropped the Name attribute, since Title is used in its place. As the features that indicate the number of Siblings - Spouses, and the number of Parents - Children on board the Titanic, both indicate the family size of an individual, we created a new attribute called Family Size. From it, we inferred another feature, namely the Family Type, based on the number of people in one's family. Next, for the feature of Cabin the first letter of itself, or U (for unknown), is used to differentiate between its different values, while for the Ticket attribute the first part of the ticket value is used. Finally, for all the attributes that are selected to be used in the predictive model in the next step, dummy encoding has been performed. This is done so that these attributes can be used in prediction models, as such models require all input to be numeric. The original attributes were then dropped from the table, as they are not going to be used by the models [10]. The result of this process is that we end up with 67 features.

2.2 Modeling and Prediction

In this section, we use the adjusted training dataset to build statistical models that can predict whether a previously unseen passenger would survive the Titanic accident or not, based on their specified characteristics. For this purpose, two different models were explored and evaluated: logistic regression [11] and random forest [14].

These models' ability to correctly estimate the class of a certain entry as survived 1 (lived) or 0 (died) was evaluated using a 5-fold cross-validation process. This method splits the data into 5 different groups, and each unique group is used as a test dataset, while the remaining 4 become the training dataset. This was done as this value leads to less bias and variance in the results [13]. The cross_val_score functionality, provided by the scikit-learn library was used for this purpose.

We evaluated and trained the selected models using both the full dataset of 67 features and a reduced dataset, using only the most important features for a high accuracy prediction, as suggested by a Random Forest classification applied to the training data. The full dataset led to better evaluations of the models, thus we were prompted to perform predictions using the larger set. Based on the evaluations of the models, we decided to use the logistic regression model, as its performance was superior to the random forest. Their scores, the output of the cross_val_score method, were 0.829 and 0.823 respectively.

Before applying the selected statistical models to the training data, we also performed a standardization of the data set, as this is a requirement for the estimators that were used, and were imported from the scikit-learn python module [14]. More specifically, the preprocessing method StandardScaler was used.

We used the logistic regression classifier, as it is the highest scored model we have, to fit the model to the transformed training data. When this was done, the results were extracted to a CSV file and uploaded to Kaggle for evaluation. The score we achieved was 0.76555 and our place on the leaderboard was 10610 at the time of the submission.

According to the evaluation of the logistic regression model on unseen data, that was performed using 5-fold cross-validation we expected a score around 0.8. However, some deviation was also something we thought could come up, as the evaluation process performed by Kaggle might differ from ours, therefore resulting in different prediction scores.

3 Research and Theory

3.1 Research - State of the art solutions

In this section a data mining competition from kaggle.com is analysed. The name of the competition was "TensorFlow - Help Protect the Great Barrier Reef." The competition took place in November 2021 and expanded until February 2022. The participants had the opportunity to win prizes ranging from 10.000$ to 30.000$ dollars. The goal of this competition was to create a model able to

detect 'coral-eating crown-of-thorns starfish' (in short 'COTS') in sequences of underwater images. COTS are a threat to the Great Barrier Reef in Australia. Hence, to control COTS outbreaks, there must be a system to detect them with the use of cameras. The dataset used in the competition consists of a set of pictures (among which only some contain COTS), alongside information about the image.

Qishen Ha, the winner of the competition, used an F2 algorithm to optimize the cross-validation. An F2 measure is a type of Fbeta measure, therefore a configurable single-score metric able to evaluate a binary classification model with the use of predictions made for the positive class [15]. Compared to other Fbeta measures, F2 gives less weight to precision and more weight to recall in the calculation of the score. First, a 2-stage pipeline is used. This implies the task is divided into two sub-tasks: object detection and classification re-score. For object detection 6 yolov5 models were created. The models are optimized by Cross-validation after they are adjusted. The CV is then 0.716. For the Classification re-score, a classification model was used so to boost the CV. The CV is then equal to 0.73. After the 2-stage pipeline, a post-processing method is performed. The post-processing method boosts the CV. According to Qishen Ha; "For example, the model has predicted some boxes B at #N frame, select the boxes from B which has a high confidence, these boxes are marked as "attention area"."[16] Finally, a value of F2= 0.74 is found by 3-fold cross-validation, which means CV = 0.74.

The peculiarity of this approach was its simplicity. Indeed, a 2-stage pipeline with 6 yolov5 models utilizing 3 folds performed much better than a single 10-fold Yoki5L6. This was a surprise to the winner of the competition.

3.2 Theory - MSE vs MAE

The mean squared error (MSE) and mean absolute error (MAE) are two error measures. The formulas for the two are as follows (from the slides):

$$MSE = \sum_{i=1}^{n} \frac{(x_i - y)^2}{n} \tag{1}$$

$$MAE = \sum_{i=1}^{n} \frac{|x_i - y|}{n} \tag{2}$$

Both the MSE and the MAE have pros and cons. Both are widely used in model performance [1]. According to Chicco et al. (2021), [3] the MSE is more sensitive to outliers than the MAE, thus making it less robust. However, depending on the loss function, it might be useful to penalize values further from the mean disproportionally more than values closer to the predicted value, hence MSE would be preferred. Additionally, the fact that the MSE is minimized using the mean of all the cases, makes it vulnerable to non-symmetrical data and can lead to misleading results. On the other hand, MAE is minimized using the median of the observed data and can be used equally well regardless of

the distribution of the data [17]. Moreover, in the MSE the unit measurement is squared, thus not providing any real value in practice. One would have to derive the RMSE (Root MSE) from it, for the original unit to be used, which is not required in MAE, therefore offering a better understanding of the result. Furthermore, MSE can be used to measure the variance of the errors of the predictions, whereas MAE measures the average of the residuals in the dataset [18].

The situation where MAE and MSE give identical results is when:

$$\sum_{i=1}^{n}(x_i - y)^2 - \sum_{i=1}^{n}|x_i - y| = 0 \tag{3}$$

This can happen when the errors, $(x_i - y)$, are equal to zero and when the errors are equal to -1 or $+1$.

We run an experiment using a dataset [19] that includes 21613 rows for different homes, with 21 features for each entry in the data. The distribution of this dataset is left-skewed, and there are some extreme outliers, so we expect that the best evaluation measurement between MSE and MAE, to be the second one, as explained in the previous paragraph.

We used the regression algorithms LinearRegression (LR) and RandomForestRegressor (RFR), from the sklearn python library. Splitting the available data to train (70%) and test (30%) we fit the two algorithms on the training data, and then performed estimations for the price of each home entry in the test data.

Running the MSE and MAE algorithms on the predictions of those models and the actual values for the price, we found that the MSE and MAE for the LR model were 48066104757.67 and 128106.34, respectively, while the same values for the RFR were 36599042118.80 and 105469.26. Because of the type of the distribution of the examined data, looking at the value of MSE will not provide any useful information. Using the MAE measurements, we can see that the RFR performed better than LR, as it managed to minimize the error to a greater extent. The main reason that could have led to this outcome is the fact that RFR supports non-linear solutions [4], which is not true for LR, thus it could fit better into the dataset, which is not linear. There is a moderate amount of noise in the data which usually reduces the performance of LR. Also, RFR tends to perform better than LR on average [20].

Thes data used was chosen as the predicted value is a numerical value, the price of a house, for the estimation of which regression is required and the performance of which can be measured using MSE and MAE. All of the features in the dataset are represented using numerical values, such as square feet of different rooms and the number of bathrooms in them. This enables us to use them in a regression problem to estimate the price for a house, based on the values of its features.

3.3 Theory - Analyze a less obvious dataset

The data in the SMS collection contains only regular text. A modeling technique that could be used on this type of data is Categorization. This is the process

of identifying the domain of a particular text (document) and assigning certain predefined tags, or classes, to it, based on its contents. Algorithms used in such models learn the likelihood that a given set of words are related to a particular category and are great for spam filtering and SMS categorization [4].

Another modeling technique that can be used on text data is clustering, which is an unsupervised machine learning process that finds natural groups in the feature space of input data[5]. It groups terms and patterns from various texts together based on similarity and creates clusters that contain several different texts with related meanings. The topics of a particular text are identified and used to place this text in the most appropriate cluster. This is done without previous knowledge of classes, as it tries to detect intrinsic structures in textual information and organize them into relevant groups.

To be able to use the text data in models, which is usually ambiguous and complex in its initial form, this data needs to become structured so that it can be analyzed and classified. Preprocessing and transformation techniques can be applied to this data to achieve that.

Such methods include the tokenization of the text, the removal of stopwords, and the lemmatization of each entry to allow the models to focus on the important words. This is useful to be performed as stopwords do not improve meaning and excluding them might increase the speed and precision of the model, and make better relations between words from the same family. One-hot encoding of both the words in each text and the labels that are assigned to them was also performed, enhanced by TF-IDF [6]. This was necessary as the applied model only accepts numeric values. Lastly, we split the provided dataset to train and test data in an 80:20 ratio.

The model that was used for the text classification was Naive Bayes, as each word was treated as a feature, and weights were assigned to each of them using TF-IDF.

The accuracy of this model was measured to be 0.971 when applied to the test data. However, looking closer into its metrics, its precision and f1-score are lacking when it comes to the "spam" label.

One way to improve the quality of the model would be to perform some feature selection after each word has been used as a different data feature and to also find auxiliary features for them [7]. Another way that the model could be to increase the number of training data to enable the model to learn better and thus provide more precise predictions. Additionally, the class imbalance that exists in the dataset, as text is labeled as ham 4825 times and as spam 747 times, can lead to low predictive accuracy for the infrequent class [8]. Balancing the different classes, using for example downsampling, could be helpful. The problem of class imbalance could also be mitigated by using a different algorithm to perform the text classification; prompting for tree-based, or more complex ensemble algorithms, could improve the quality of our model as such algorithms tend to perform well on imbalanced data [9]. Lastly, performing some model validation to optimize the hyperparameters used in the underlying algorithm, could lead to better performance for our model.

Bibliography

[1] Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE). Geoscientific Model Development Discussions, 7(1), 1525-1534.

[2] Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Climate research, 30(1), 79-82.

[3] Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. PeerJ Computer Science, 7, e623.

[4] Top 7 Text Mining Techniques, https://www.analyticssteps.com/blogs/top-7-text-mining-techniques. Last accessed 21 Apr 2022

[5] 10 Clustering Algorithms With Python, https://machinelearningmastery.com/clustering-algorithms-with-python/#:~:text=Cluster%20analysis%2C%20or%20clustering%2C%20is,or%20clusters%20in%20feature%20space. Last accessed 21 Apr 2022

[6] Understanding TF-IDF, https://monkeylearn.com/blog/what-is-tf-idf/. Last accessed 21 Apr 2022

[7] Zhang, W. & Gao, F. (2011). An Improvement to Naive Bayes for Text Classification. SciVerse ScienceDirect, 2160-2164

[8] Class Imbalance Problem, https://link.springer.com/referenceworkentry/10.1007/978-0-387-30164-8_110. Last accessed 21 Apr 2022

[9] 10 Techniques to deal with Imbalanced Classes in Machine Learning, https://www.analyticsvidhya.com/blog/2020/07/10-techniques-to-deal-with-class-imbalance-in-machine-learning/. Last accessed 21 Apr 2022

[10] 3 Ways to Encode Categorical Variables for Deep Learning, https://machinelearningmastery.com/how-to-prepare-categorical-data-for-deep-learning-in-python/. Last accessed 21 Apr 2022

[11] Logistic Regression for Machine Learning, https://www.capitalone.com/tech/machine-learning/what-is-logistic-regression/. Last accessed 21 Apr 2022

[12] Random forest, https://en.wikipedia.org/wiki/Random_forest. Last accessed 21 Apr 2022

[13] Introduction to k-fold Cross-Validation, https://machinelearningmastery.com/k-fold-cross-validation/. Last accessed 21 Apr 2022

[14] Preprocessing data, scikit-learn, https://scikit-learn.org/stable/modules/preprocessing.html. Last accessed 21 Apr 2022

[15] A Gentle Introduction to the Fbeta-Measure for Machine Learning, https://machinelearningmastery.com/fbeta-measure-for-machine-learning/. Last accessed 22 Apr 2022

[16] Trust CV – 1st Place Solution, https://www.kaggle.com/c/tensorflow-great-barrier-reef/discussion/307878. Last accessed 22 Apr 2022

[17] Mean vs Median, https://web.archive.org/web/20181017210824/http://www.conceptstew.co.uk/pages/mean_or_median.html. Last accessed 21 Apr 2022

[18] MAE, MSE, RMSE comparison, https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-met
Last accessed 21 Apr 2022

[19] Dataset used in task 3B, https://github.com/zaynaib/machineLearningWashington/blob/week1/home_data.csv. Last accessed 21 Apr 2022

[20] Comparative Study on Classic Machine learning Algorithms, https://towardsdatascience.com/comparative-study-on-classic-machine-learning-algorithms-24f9ff6ab222.
Last accessed 21 Apr 2022