REPORT ON FINDINGS (Word Limit: 1 000 words)

1. INTIAL DATA REVIEW

The data provided is made up of approximately 12 330 data entries spread across 18 categorical and numerical features, where 'Revenue' has been determined as the target feature. See *Table 1.1* below.

Data type	Feature
Numerical	Administrative, Administrative_Duration, Informational, Informational_Duration , ProductRelated , ProductRelated, Duration , BounceRates , ExitRates , PageValues , SpecialDay
Categorical	Month, OperatingSystems, Browser, Region, TrafficType, VisitorType, Weekend
Target	Revenue

Table 1.1

Upon initial inspection there were no missing values or any characters synonymous with unknown values (such as '?' string values). Following this, the data was split into a 70/30 training/test set, using the 'train_test_split' function from Scikit-Learn: representing 8 631 and 3 699 data points, respectively. Splitting the data first ensured no data leakage during both data analysis and model creation. This also removed contamination risk for the test data (see section 1.3 of the notebook).

2. DATA ANALYSIS

The data was then analysed by considering categorical and numerical features separately.

2.1. CATEGORICAL FEATURE ANALYSIS

To ensure stratified sampling, the proportion of categories in each categorical feature of the training set was compared to that of the whole dataset, see section 2.1.1 of the notebook. The differences between the two for each of the categories were added together and are represented in *Table 2.1.1*, showing minimal differences.

Feature	Combined Differences for All Categories in Each Feature
Operating System	-2.7647e-17
Browser	9.8527e-18
Region	-2.0817e-17
Traffic Type	3.6809e-17
Visitor Type	5.5511e-17
Weekend	5.5511e-17
Month	4.6838e-17
Total	**1.5953e-16**

Table 2.1.1

Each of the categories were visually evaluated to better understand their relationship to ‘Revenue.’ *Figure 2.1.1* illustrates the percentage of ‘True’ and ‘False’ revenue for categories in each categorical feature.

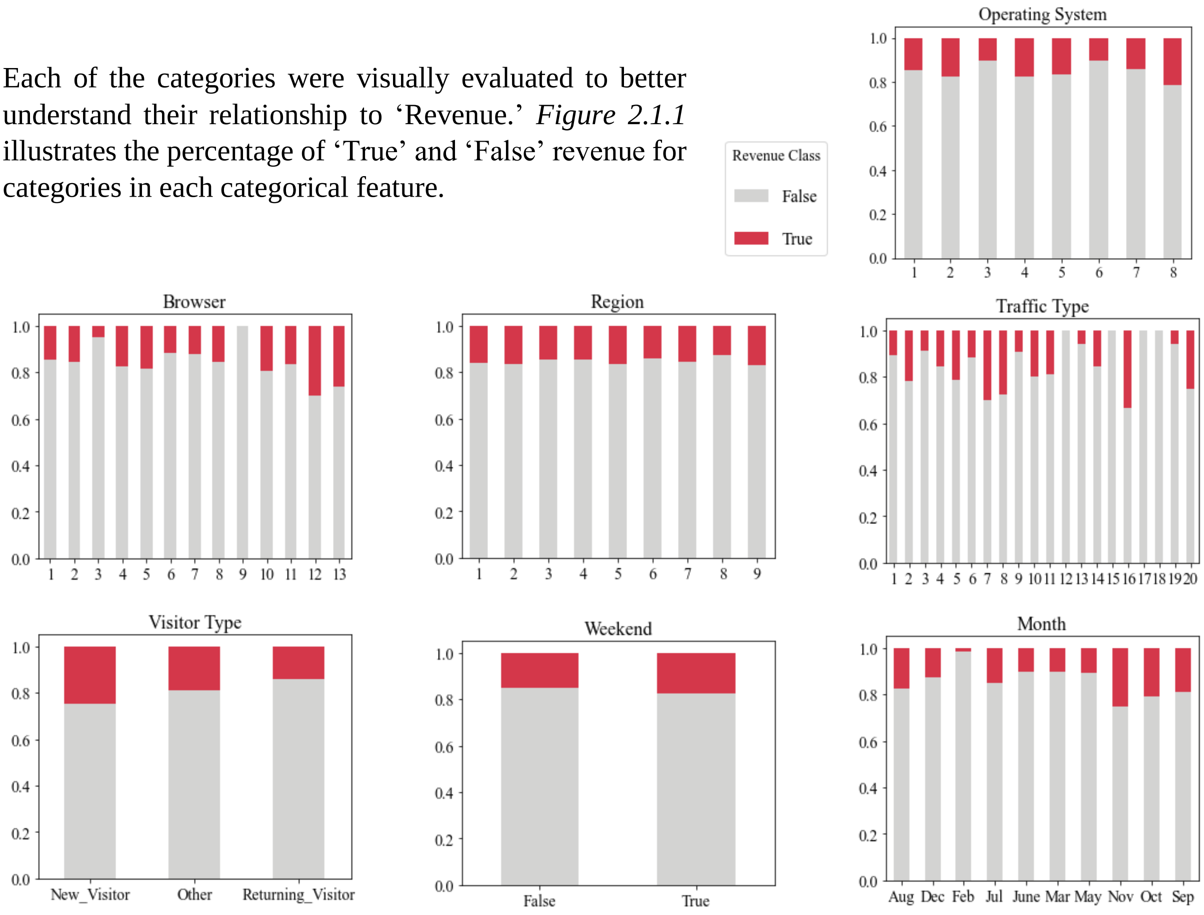

Figure 2.1.1

2.1.1. CATEGORICAL FEATURE RELEVANCE

Relevance of categorical features was analysed using the Chi-Squared score with the ‘SelectKBest’ function from Scikit-Learn (see section 2.1.3 of the notebook).

CHI-SQUARED: The features ‘Traffic Type’ and ‘Weekend’ do not appear to be significant at the 95% confidence level, see *Table 2.1.2* and *Figure 2.1.2*.

Feature	Chi-Squared Statistic	P-Value
0: Operating System	5.68	0.0172
1: Browser	4.17	0.0412
2: Region	5.95	0.0147
3: Traffic Type	1.17	0.2798
4: Visitor Type	25.84	3.7036e-07
5: Weekend	3.80	0.0511
6: Month	62.28	2.9815e-15

Table 2.1.2

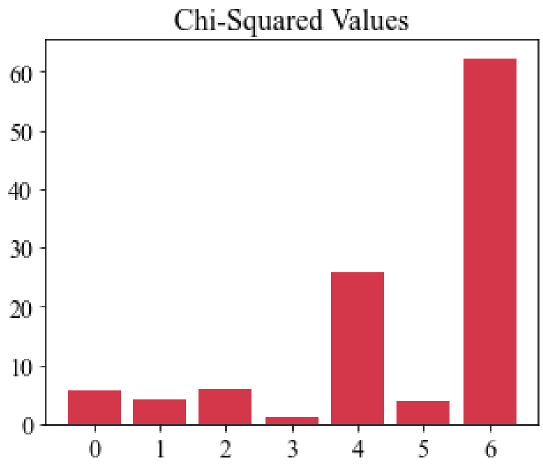

Figure 2.1.2

2.2. NUMERICAL FEATURE ANALYSIS

Figure 2.2.1 displays the non-linear relationships between the numerical features. A statistical summary of the features can also be seen in *Table 2.2.1*. Generally, numerical features appear to display both a high level of standard deviation and presence of outliers.

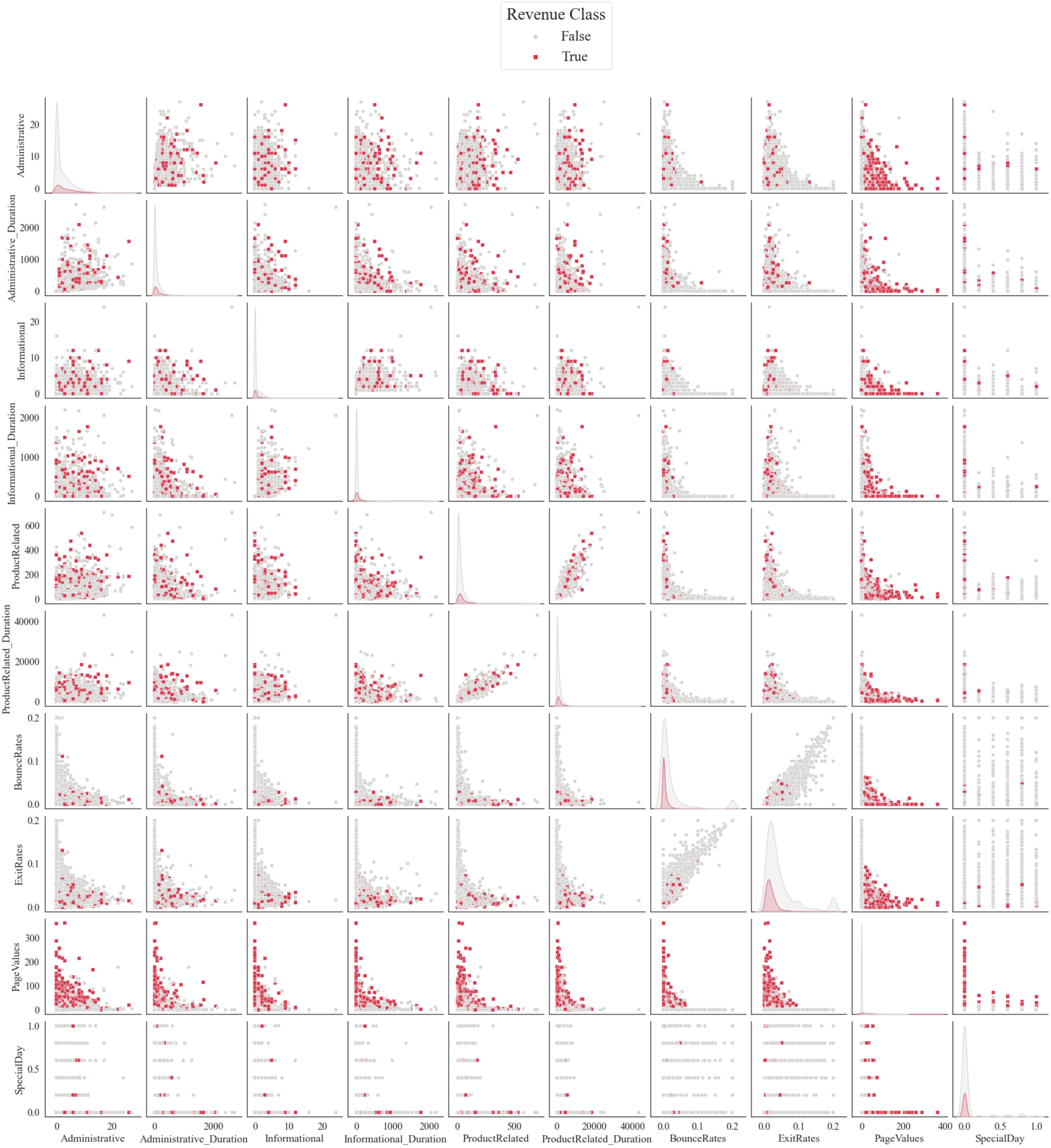

Figure 2.2.1

Feature	Count	Mean	Std	Min	25%	50%	75%	Max
Administrative	8 631	2.302	3.291	0.000	0.000	1.000	4.000	27.000
Administrative Duration	8 631	80.510	173.09	0.000	0.000	7.625	91.950	2720.5
Informational	8 631	0.509	1.281	0.000	0.000	0.000	0.000	24.000
Informational Duration	8 631	34.315	135.65	0.000	0.000	0.000	0.000	2195.3
Product Related	8 631	31.688	44.379	0.000	7.000	18.000	37.000	705.00
Product Related Duration	8 631	1195.7	1828.5	0.000	186.48	602.88	1477.6	43171
Bounce Rates	8 631	0.022	0.047	0.000	0.000	0.003	0.017	0.200
Exit Rates	8 631	0.042	0.048	0.000	0.014	0.025	0.050	0.200
Page Values	8 631	5.931	18.840	0.000	0.000	0.000	0.000	361.76
Special Day	8 631	0.060	0.196	0.000	0.000	0.000	0.000	1.000

Table 2.2.1

2.2.1. NUMERICAL FEATURE RELEVACE

Feature relevance was tested by a combination of the Kendal Tau Rank Coefficient using SciPy, and the ANOVA F-Classification using the 'SelectKBest' functionality from Scikit-Learn (see section 2.2.3 of the notebook).

KENDALL TAU COEFFICIENT: All correlations appear to be significant at the 95% confidence level, see *Table 2.2.2*.

Feature	Kendall Correlation	P-Value
Administrative	0.14	3.2137e-49
Administrative Duration	0.14	5.5048e-49
Informational	0.11	6.4001e-27
Informational Duration	0.11	4.6722e-26
Product Related	0.16	7.5339e-70
Product Related Duration	0.17	6.2079e-85
Bounce Rates	-0.14	1.2423e-47
Exit Rates	-0.21	1.4202e-123
Page Values	0.59	0.0000e+00
Special Day	-0.09	1.3728e-17

Table 2.2.2

ANOVA F-CLASSIFICATION: All features appear to be significant at the 95% confidence level, see *Table 2.2.3* and *Figure 2.2.2*.

Feature	F-Statistic	P-Values
0: Administrative	131.99	2.493e-30
1: Administrative Duration	78.33	1.044e-18
2: Informational	80.83	2.986e-19
3: Informational Duration	41.95	9.878e-11
4: Product Related	187.16	3.639e-42
5: Product Related Duration	195.78	5.245e-44
6: Bounce Rates	197.31	2.477e-44
7: Exit Rates	382.29	2.466e-83
8: Page Values	2 713.91	0.000
9: Special Day	67.92	1.949e-16

Table 2.2.3

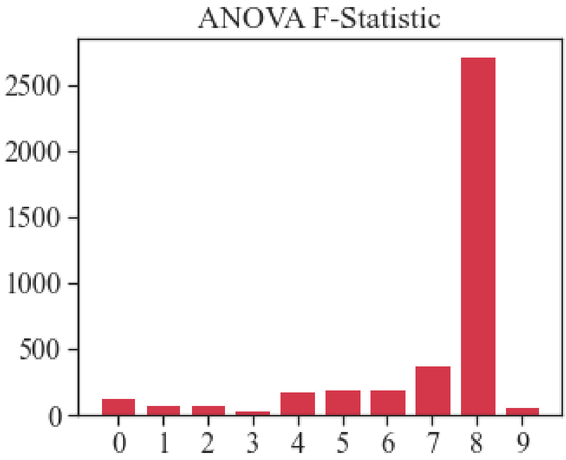

Figure 2.2.2

3. DATA PRE-PROCESSING

Data pre-processing occurs in sequential steps using both the ‘Pipeline’ and ‘ColumnTransformer’ functions from Scikit-Learn. Data transformations for each type of feature were assigned to Pipeline object steps. The associated features were then combined with the relevant transformation steps using a ColumnTransformer object (see section 3 of the notebook). All non-specified features were dropped to ensure that all data used by the machine learning models are in the correct format. In both categorical and numerical features, an ‘Imputer’ function was used to account for any potential missing values in the future.

3.1. PRE-PROCESSING: CATEGORICAL FEATURES

As per the results of the Chi-Squared test, in section 2.1.1 above, the features ‘Weekend’ and ‘TrafficType’ were removed from the dataset to improve cardinality. Furthermore, given that most of the features have six or more subcategories, a binary encoder was used to maintain the nominal nature of the features while also limiting dimensionality.

3.2. PRE-PROCCESING: NUMERICAL FEATURES

The Kendall Correlation and F-Test in section 2.2.1 above indicated that all numerical features are statistically significant, thus all numerical features were included. All attributes, other than ‘SpecialDay,’ were standardised via the ‘StandardScaler’ function from Scikit-Learn to account for the outliers as noted in section 2.2 above. Separately, the attribute ‘SpecialDay’ was normalised with the ‘MinMaxScaler’ function in Scikit-Learn as the nature of the feature limits data points to the range [0,1].

3.3. PRE-PROCESSING: OTHER

Principal Component Analysis (PCA) with a 95% explained variance requirement was tested on each model; and used if it showed that the model performance was not negatively impacted. *Figure 3.1* shows the impact of PCA on the processed data.

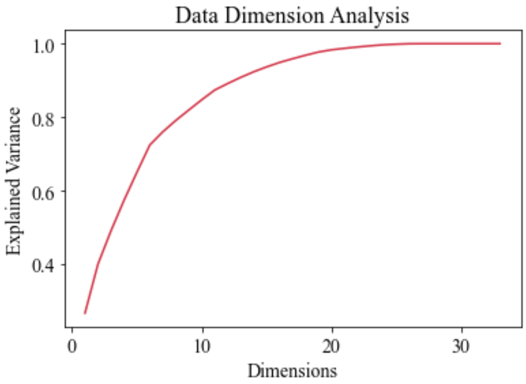

Figure 3.1

4. MACHINE LEARNING MODELS

Three independent classification models were selected to ensure a diverse and relevant mix of machine learning algorithms. Furthermore, each model used 'SMOTE' functionality from Imbalanced-Learn to account for imbalanced data. Hyperparameters were initially tested using a Random Search Analysis and subsequently fine-tuned using a Grid Search Analysis. The models were then tested and refined using the training data (see section 4 of the notebook), and only then applied to the test data for final evaluation (see section 5 of the notebook). Given the noisy and imbalanced nature of the dataset, all models were programmed with high levels of regularisation.

4.1. LOGISTICAL REGRESSION MODEL

A simple yet effective algorithm that is not computationally expensive but relies on a linear relationship. To address the non-linearity identified in section 2.2 above, the 'PolynomialFeatures' function at the second degree was used from Scikit-Learn. Furthermore, PCA was utilised to combat the curse of dimensionality. The model's inability to handle noisy labels resulted in low precision and can be seen in the evaluation results below (see *Table 4.1.1* and *Table 4.1.2*).

	Precision	**Recall**	**F-Measure**	**Accuracy**	**ROC Curve**
Training Data	51.8%	77.4%	61.5%	84.1%	89.2%
Test Data	51.2%	77.2%	61.5%	85.0%	81.8%
Difference	*-0.6%*	*-0.2%*	*0.0%*	*0.9%*	*-7.4%*

Table 4.1.1

		Predicted		**Total**
		True	*False*	
Actual	*True*	444	131	575
	False	424	2 700	3 124
Total		868	2831	**3 699**

Table 4.1.2

4.2. SUPPORT VECTOR MACHINE MODEL (SVM)

A memory efficient model that performs well in classification. A polynomial kernel was used at the second degree combined with a 'coef0' term of 0.2 to account for the non-linearity of the dataset. See *Table 4.2.1* and *Table 4.2.2* for results.

	Precision	**Recall**	**F-Measure**	**Accuracy**	**ROC Curve**
Training Data	58.4%	75.2%	65.8%	87.9%	90.3%
Test Data	57.4%	77.6%	66.0%	87.6%	83.5%
Difference	*-1.0%*	*2.4%*	*0.2%*	*-0.3%*	*-6.8%*

Table 4.2.1

		Predicted		**Total**
		True	*False*	
Actual	*True*	446	129	575
	False	331	2 793	3 124
Total		777	2 922	**3 699**

Table 4.2.2

4.3. RANDOM FOREST CLASSIFICATION MODEL

This algorithm is useful for larger datasets and is more robust with respect to noise, therefore it was considered the most appropriate given the data. The oob score was used to evaluate the model, and control overfitting by balancing the number of trees, tree depth, maximum features, and minimum leaf size. Evaluation results are seen in *Table 4.3.1* and *Table 4.3.2*.

	Precision	Recall	F-Measure	Accuracy	ROC Curve
Training Data	57.7%	79.5%	67.3%	88.0%	92.2%
Test Data	57.1%	79.7%	66.5%	87.5%	84.3%
Difference	*-0.6%*	*0.2%*	*-0.8%*	*-0.5%*	*-7.9%*

Table 4.3.1

		Predicted		**Total**
		True	*False*	
Actual	*True*	458	117	575
	False	344	2 780	3 124
Total		802	2 897	**3 699**

Table 4.3.2

5. RESULTS DISCUSSION

All the models show good generalisation with minimal differences between training and test set predictions, and relatively high recall, accuracy, and ROC scores. The high recall scores demonstrate the ability of the models to identify positive revenue intentions and minimise false negative predictions, which coupled with high accuracy could be important for an e-commerce website. However, low precision rates indicate that the models have difficulty determining the difference between false positives and true positives. Considering all models experienced this, the nature of the data (imbalanced and noisy labels) could be a factor to low precision. Considered individually, all three models performed similarly; however, the Random Forest model outperformed which is expected given the model’s unique ability to handle both numerical and categorical features as well as noisy data points.

REFERENCES

Brownlee, J. 2020. SMOTE for Imbalanced Classification with Python. Available at: https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/ [Accessed 14 January 2021]

Géron, A. 2019. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 2nd ed. California: O'Reilly Media.

Guillaume, L. et al. 2017. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research*18(17), pp. 1-5. Available at: https://jmlr.org/papers/v18/16-365 [Accessed 03 February 2021]

Li, Y. 2020. *Data Pre-Processing*. [Lecture to Msc Data Science and Analytics]. Cardiff University, 07 December 2020.

Li, Y. 2020. *Regression, Generalisation & Model Evaluation*. [Lecture to Msc Data Science and Analytics]. Cardiff University, 14 December 2020.

Li, Y. 2021. *Classification*. [Lecture to Msc Data Science and Analytics]. Cardiff University, 11 January 2021.

Li, Y. 2021. *Ensemble Learning*. [Lecture to Msc Data Science and Analytics]. Cardiff University, 18 January 2021.

Pedregosa, F. et al. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*12(85), pp. 2825-2830. Available at: https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html [Accessed 03 February 2021]