**Week 3**

**Summary**

**Milestones achieved**

* Ran classifier model data set, and recorded the major metrics like: accuracy, TP rate, FP rate, precision, recall, F measure, the ROC area etc**.**
* Compared between different evaluation metrics.
* Performed data preprocessing, feature engineering. and a chi-square test for identifying significant feature correlations with the target variable (Loyality).

**Conclusions**

* Logistic Regression and Decision Tree have perfect scores across all metrics.
* Random Forest comes very close to perfection but is slightly lower in recall and F1 score.
* KNN, while still performing well, shows slightly lower recall, F1 score, and ROC AUC compared to the other models. It struggles a bit more in identifying all true positives (lower recall) but has high precision, meaning when it does predict positive, it's usually right.
* In summary, Logistic Regression and Decision Tree are the best performers in this comparison, followed closely by Random Forest, while KNN performs well but not as strongly as the others.

**References**

* <https://medium.com/@outside2SDs/an-overview-of-correlation-measures-between-categorical-and-continuous-variables-4c7f85610365>
* <https://colab.research.google.com/drive/1mqfvb_NTjm1g_OPYrnck_EreWclqHzhx?usp=sharing>
* <https://www.tutorialspoint.com/machine_learning_with_python/index.htm>
* Müller, AC, & Guido, S 2016, Introduction to Machine Learning with Python : A Guide for Data Scientists, O'Reilly Media, Incorporated, Sebastopol. Available from: ProQuest Ebook Central. [29 September 2024].
* <https://ebookcentral.proquest.com/lib/hw/reader.action?docID=4698164&ppg=28>
* <https://github.com/ageron/handson-ml2/blob/master/03_classification.ipynb>

**Next Steps**

* Clustering Algorithms
* Organize the Pipeline; Modularize the code: Each step (downloading, loading, preprocessing, visualization, etc.) should be a function, making it easier to maintain and extend.
* Ensure the pipeline runs end-to-end: From downloading the dataset (if necessary) to splitting it into train and test sets.
* Working on image data set