Appendix A: Model Hyperparameters and Values

Model	Hyperparameter	Values	Comment
K-Nearest Neighbours (KNN)	*K*	*2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17, 20, 25, 30, 35, 40, 45, 50*	*K is the number of neighbouring points used to define the boundary - larger values generate a smoother curve*
	Weights	*Uniform, Distance*	*How proximity is defined. Uniform means all neighbours have the same weighting. Distance means the weighting is inversely proportional.*
Logistic Regression (LG)	*Penalty*	*None, L1 (Lasso), L2 (Ridge), ElasticNet (Ridge + Lasso)*	*The type of Penalty used to prevent overfitting.*
	C	*0.001, 0.005, 0.01, 0.015, 0.02, 0.03, 0.05, 0.07, 0.1, 0.12, 0.15, 0.2, 0.25, 0.3, 0.5, 1, 3, 10, 30, 100*	*Regularisation strength. Larger values means weaker regularisation.*
	*L1:L2 Ratio**	*0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1*	**Ratio of influence between L1 and L2 regularisation - only applicable for ElasticNet.*
Support Vector Machine (SVM)	*Kernel*	*Linear, Radial Basis Function (RBF), Polynomial and Sigmoid.*	*The function used to transform the values into the feature space. Allows a hyperplane to act as a non-linear boundary.*
	C	*0.001, 0.005, 0.01, 0.015, 0.02, 0.03, 0.05, 0.07, 0.1, 0.12, 0.15, 0.2, 0.25, 0.3, 0.5, 1, 3, 10, 30, 100*	*Regularisation strength. Larger values means weaker regularisation.*
	Polynomial Degree	*2, 3, 4*	*Only for Kernel = Polynomial*
	Gamma	*“Scale” or “Auto”*	*The coefficient of the non-linear kernel - not applicable to Linear kernels. See Sci-kit documentation for calculation of gamma. (Here)*
	Class Weight	*Balanced, None*	*Applies weight to coefficients of C for each class ($Weight_{class}$ x C_{class}). None sets all weights to one while Balanced makes*

			them inversely proportional to their frequency in the data.
*Linear Support Vector Machine (LSVM)**	*Kernel*	*Linear*	**The specialised SVM which only uses a linear kernel but has more options for regularisation and loss.*
	Penalty	*L1 (Lasso), L2 (Ridge)*	*The type of Penalty used to prevent overfitting.*
	Loss	*Hinge, Hinge Squared*	*Hinge loss is the default loss function for SVM for classification.*
	C	*0.001, 0.005, 0.01, 0.015, 0.02, 0.03, 0.05, 0.07, 0.1, 0.12, 0.15, 0.2, 0.25, 0.3, 0.5, 1, 3, 10, 30, 100*	*Regularisation strength. Larger values means weaker regularisation.*
	Class Weight	*Balanced, None*	*Applies weight to coefficients of C for each class (class weight x C_{class}). None sets all weights to one while Balanced makes them inversely proportional to their frequency in the data.*
Random Forest (RF)	*Number of Trees*	*50, 100, 500, 1000*	*The number of Decision Trees trained. Larger values are more complex but more expensive computationally.*
	Minimum Samples for Split	*5, 8, 10, 20, 40*	*Minimum number of samples allowed to allow another split. Larger values act as a form of regularisation to prevent complex trees.*
	Minimum Samples in Leaf	*1, 4, 8*	*Minimum number of samples allowed in a leaf (terminal) node. Larger values act as a form of regularisation to prevent complex trees.*
	Maximum Number of Features	*All, Square Root of All*	*Defines what features/variables are used to define the optimal split. Square Root of All would use 10 features if we had 100 features.*
	Maximum Tree Depth	*5, 10, 30, NA*	*Limiting the depth of the Decision Tree. NA means no limit is applied.*

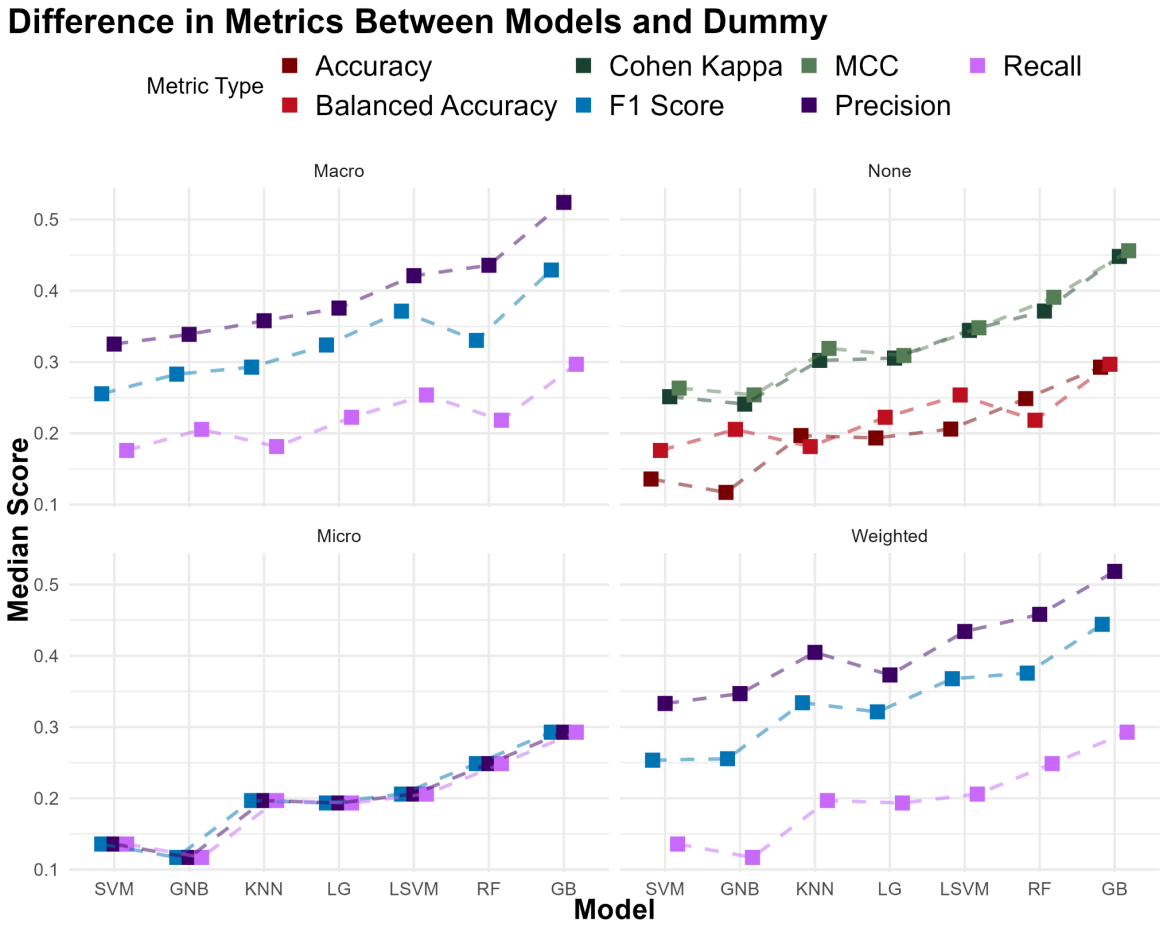
	Splitting Criterion	*Gini impurity, Shannon Entropy*	*The function used to define how best to split the samples into two.*
	Class Weight in Bootstrapping	*Balanced, None*	*Balanced makes the less frequent classes as likely to select as the largest classes. None means the sampling is proportional to frequency.*
Gradient Boosted Trees (GB)	*Learning Rate*	*0.05, 0.1, 0.2*	*The size of influence of the previous tree on the next tree.*
	Number of Trees	*50, 100, 500, 1000*	*The number of Decision Trees trained. Larger values are more complex but more expensive computationally.*
	Minimum Samples for Split	*5, 8, 10, 20, 40*	*Minimum number of samples allowed to allow another split. Larger values act as a form of regularisation to prevent complex trees.*
	Minimum Samples in Leaf	*1, 4, 8*	*Minimum number of samples allowed in a leaf (terminal) node. Larger values act as a form of regularisation to prevent complex trees.*
	Maximum Number of Features	*All, Square Root of All*	*Defines what features/variables are used to define the optimal split. Square Root of All would use 10 features if we had 100 features.*
	Maximum Tree Depth	*5, 10, 30, NA*	*Limiting the depth of the Decision Tree. NA means no limit is applied.*

Appendix B: Model Configuration and Implementation*

**Specifying all non-default values used to set-up the models. The default values are for version 0.24 for Sci-Kit Learn - found Here.*

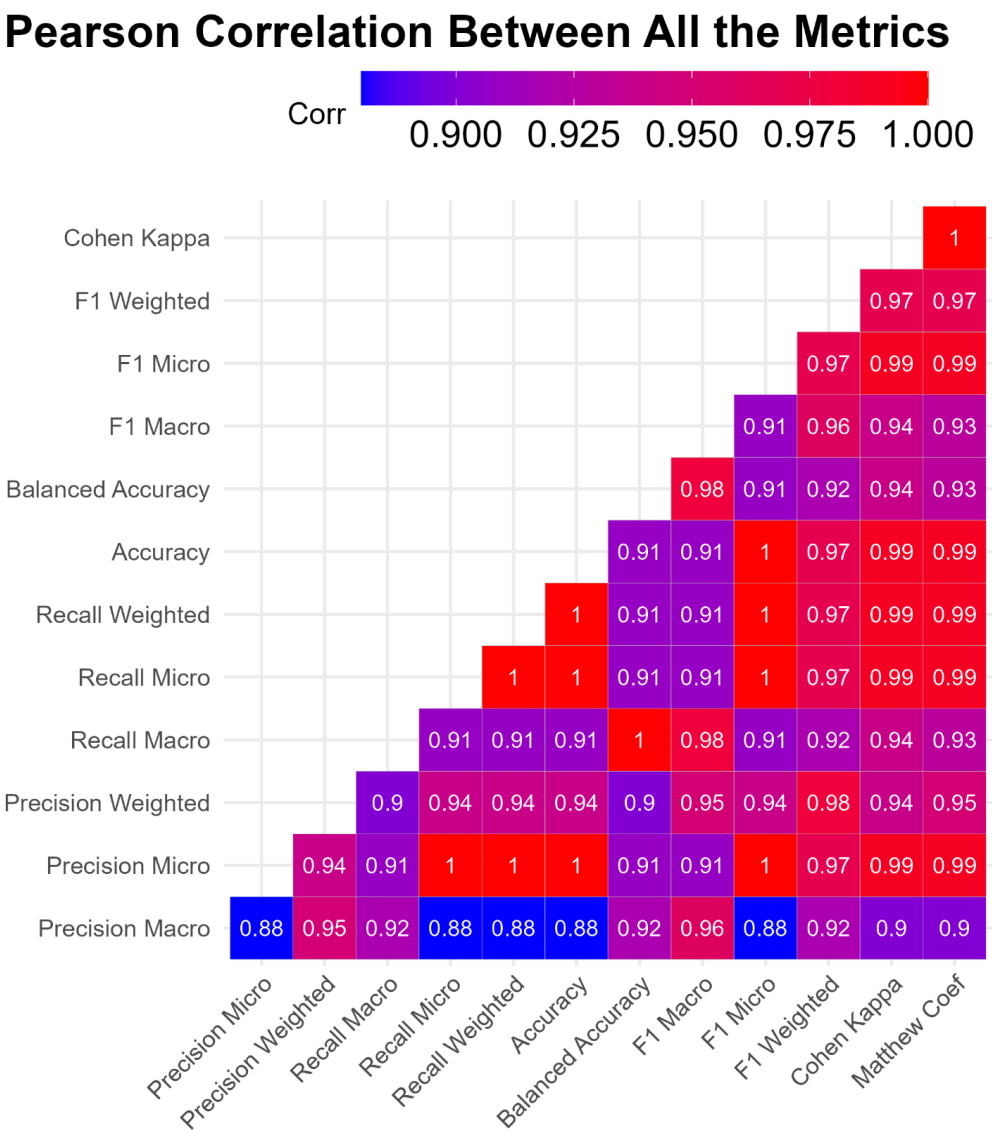
Model	Option	Value	Comment
Logistic Regression (LG)	*Solver*	*Saga*	*Only multi-class classifier which supported L1, L2 and ElasticNet Regularisation*
	Maximum Number of Iterations	*5,000*	*Maximum number of iterations during optimisation before terminating the process.*
Gradient Boosted Trees (GB)	*Number of Iterations to stop after no improvement.*	*10*	*Enables early stopping, if the minimum error in the validation set is not improved after 10 iterations. Generally, it prevents overfitting.*
	Size of Validation Set	*10%*	*(This is the default but it is worth stating explicitly)*
Support Vector Machine (SVM) and (LSVM)	*Maximum Number of Iterations*	*10,000*	*Maximum number of iterations during optimisation before terminating the process.*

Appendix C: Comparison of Models Between Dummy Classifier



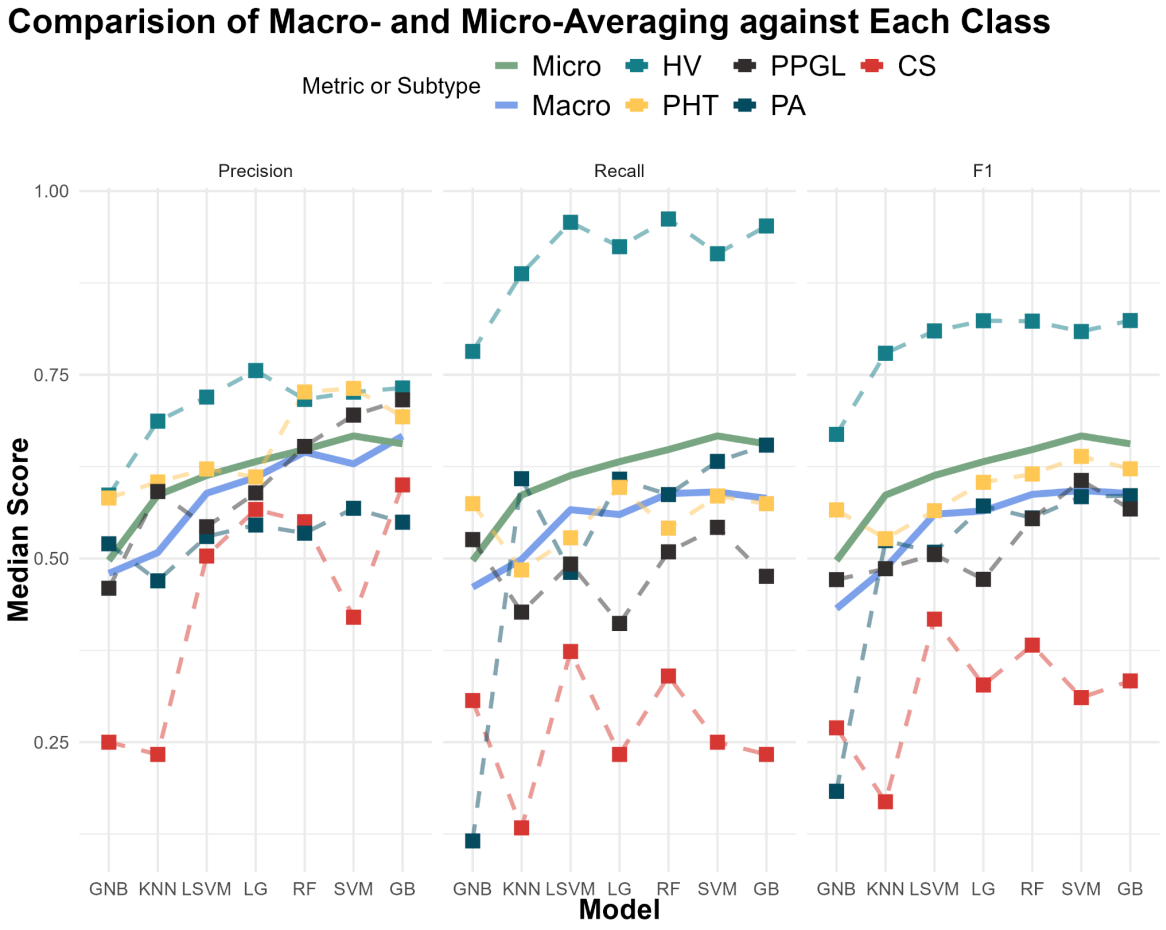
Appendix C: The median difference between the metric values achieved by the model and the Dummy classifier, across the different metrics. The four plots are split by the averaging method, including Macro, Micro, Weighted and None (where no averaging is required). The larger the difference between a genuine model and the Dummy model, the more information about the classes that model has extracted. The difference in Accuracy is much smaller than the difference in MCC or Cohen's Kappa, indicating the absolute value of Accuracy could mainly be described by the data rather than model. It is clear the improvements of a genuine model compared to the Dummy is the Precision rather than Recall, given the gain of Precision is nearly double that of Recall. Micro-averaging had the smallest difference because it obtained the highest scores in the Dummy model (see **Figure 1**). (Models: **GNB** - Gaussian Naive Bayes, **KNN** - K-Nearest Neighbours, **LSVM** - Linear (Kernel) Support Vector Machine, **SVM** - Support Vector Machine, **LG** - Logistic Regression, **RF** - Random Forest, **GB** - Gradient Boosted Trees and the Metric **MCC** - Matthew Correlation Coefficient)

Appendix D: Correlation of All the Metrics



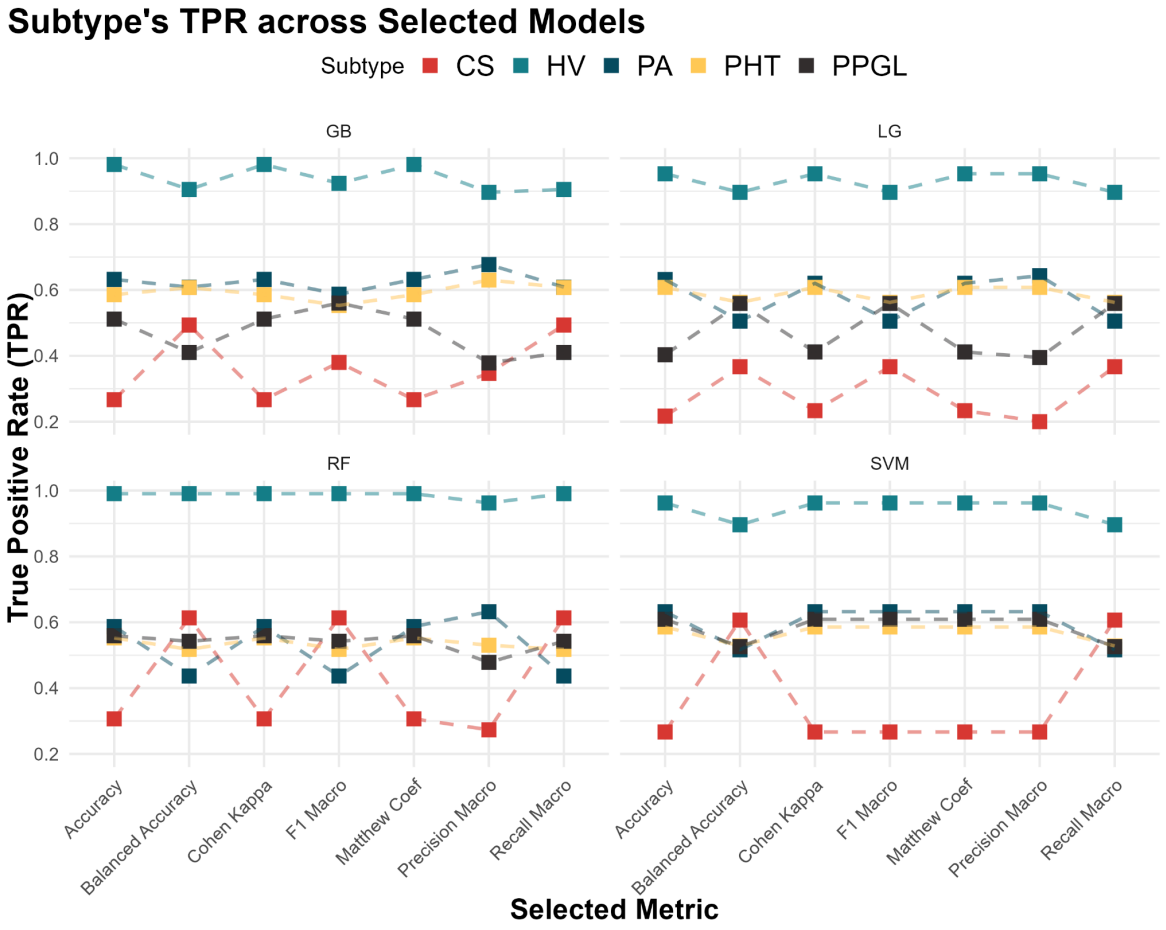
Appendix D: The Pearson Correlation of the metrics for the top ten set of hyperparameters for each model. The colouring ranges from 0.88 (blue) to 1 (red) to emphasise the difference, but the high level of correlations should not be overlooked. Metrics with very high correlations suggest they provide very similar information about the models and there is little to gain from using both simultaneously. As expected, Micro-averaged metrics have perfect correlation with Accuracy and each other. Both MCC and Cohen's Kappa have high correlation with Micro-averaged metrics compared to Macro-averaged. It seems the most distinct metric is Macro Precision, as it differs from Accuracy the most. Generally, Macro-averaging were less correlated compared to Weighted and Micro-averaging, although Weighted averaging sometimes had perfect correlation with Micro-averaged metrics. This is consistent with **Figure 2** and **Appendix C**, as the difference in Recall and Precision is preserved in Macro-averaging and is diminished or absent in Weighted and Micro averaged metrics, respectively.

Appendix E: Duplicate for Figure 3 With Full Dataset



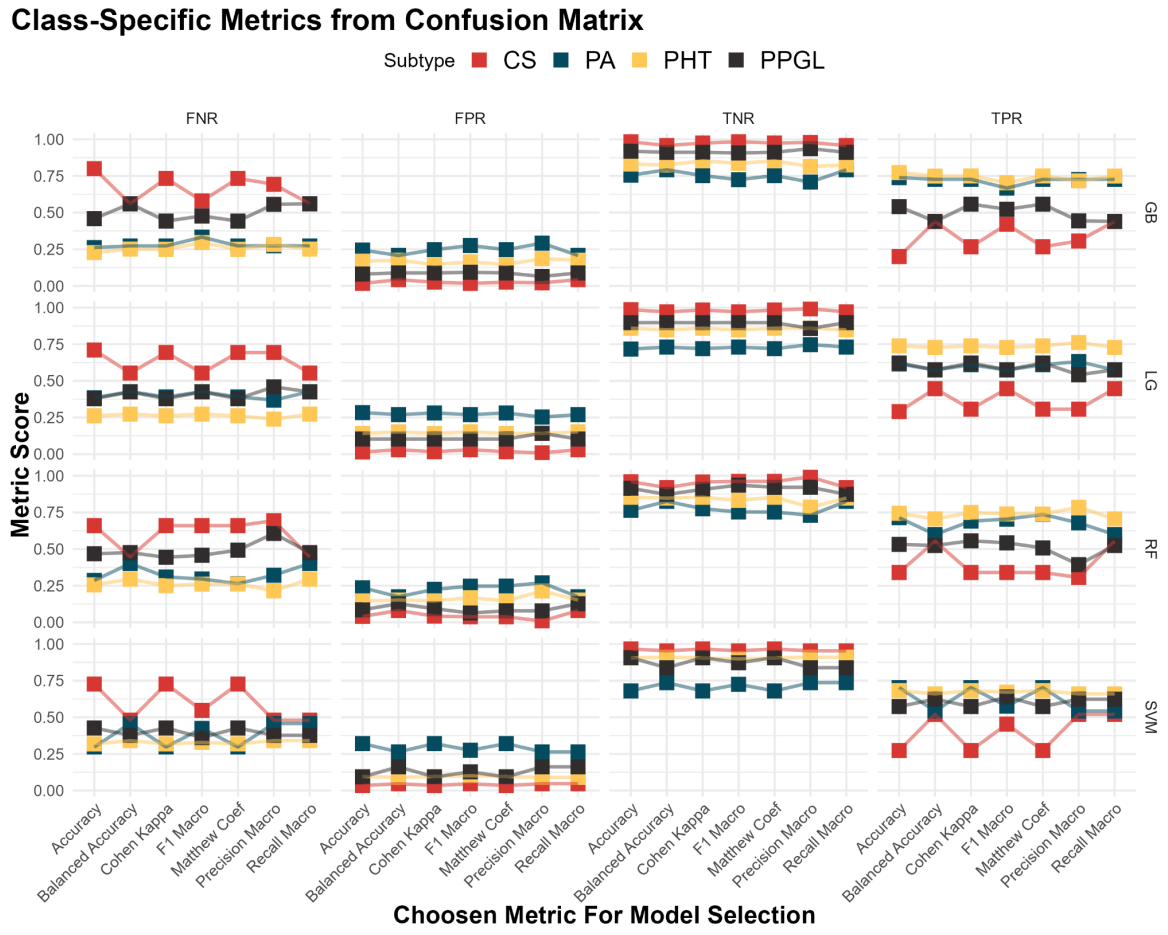
Appendix E: The same type of graph presented in **Figure 3**, but using the complete dataset, instead of removing the healthy volunteers (HV). HV appears to be much easier to classify compared to the hypertensive subtypes which may be due to the size of the class (133). The insensitivity of Micro is consistent for this dataset too, for example the Precision of SVM and the Recall of LG. In both these cases, Macro decreased because there is a drop in performance while Micro increased as the drop in performance occurs for the smaller classes. (Models: **GNB** - Gaussian Naive Bayes, **KNN** - K-Nearest Neighbours, **LSVM** - Linear (Kernel) Support Vector Machine, **SVM** - Support Vector Machine, **LG** - Logistic Regression, **RF** - Random Forest, **GB** - Gradient Boosted Trees and the Subtypes **HV:** Healthy Volunteer, **PHT** - Primary Hypertension, **PA** - Primary Aldosteronism, **PPGL** - Pheochromocytoma/catecholamine-producing Paraganglioma, **CS** - Cushing Syndrome)

Appendix F: Duplicate for Figure 4 With Full Dataset



Appendix F: this is a replication of **Figure 4**, with the full dataset, including the health volunteers (HV). The best set of hyperparameters are determined by an individual metric (shown on the x-axis) and the consequential True Positive Rate (TPR, or Recall) of each class are presented (for the top four classifiers for simplicity). Unsurprisingly the best performance in the majority class (HV) but the very high Recall was unexpected. Similar to **Figure 4**, the selected metric disproportionately impacts the performance in the smaller classes which is shown by the variation of TPR. HV, PHT and PA are minimal variations whereas PPGL and, especially, CS have much more pronounced differences. Specifically, for RF, the change from Accuracy and Balanced Accuracy doubled the recall of CS whilst the Recall of HV was unaffected. (Models: **GB** - Gradient Boosted Trees, **LG** - Logistic Regression, **RF** - Random Forest and **SVM** - Support Vector Machine, Subtypes: **HV:** Healthy Volunteer, **PHT** - Primary Hypertension, **PA** - Primary Aldosteronism, **PPGL** - Pheochromocytoma/ catecholamine-producing Paraganglioma, **CS** - Cushing Syndrome)

Appendix G: Difference in Performance Across Top Four Models - Across the Four Metrics



Appendix G: The class-specific True Positive Rate (TPR), True Negative Rate (TNR), False Positive Rate (FPR) and False Negative Rate (FNR) are shown as columns with the four rows displaying the four models. The x-axis indicates the metric used to determine the best set of hyperparameters to illustrate the potential down-stream effects of using that metric in model selection. The TPR was presented in **Figure 4** because the variation is the most interesting and informative. (Models: **GB** - Gradient Boosted Trees, **LG** - Logistic Regression, **RF** - Random Forest and **SVM** - Support Vector Machine, Subtypes: **HV:** Healthy Volunteer, **PHT** - Primary Hypertension, **PA** - Primary Aldosteronism, **PPGL** - Pheochromocytoma/ catecholamine-producing Paraganglioma, **CS** - Cushing Syndrome)