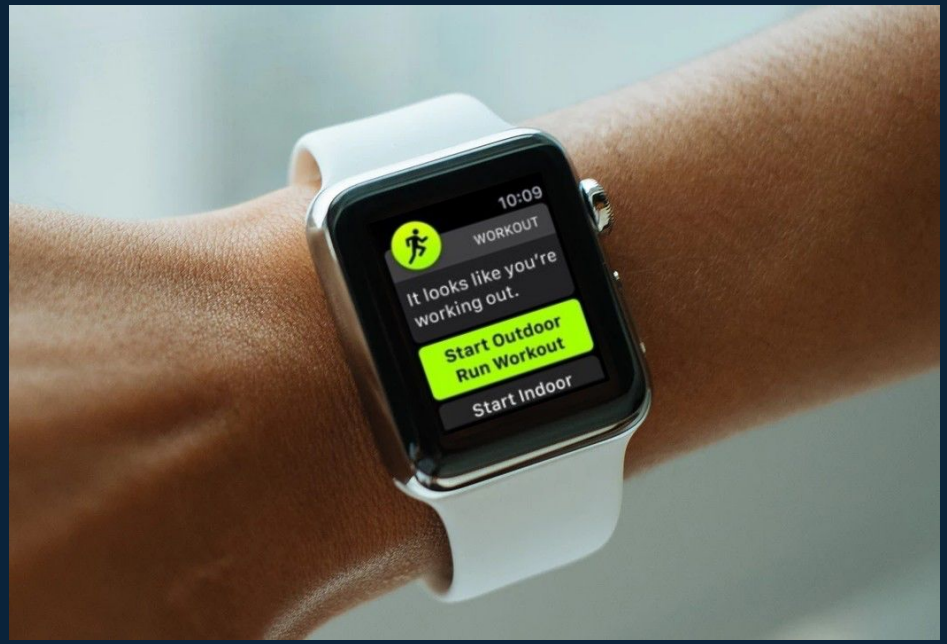
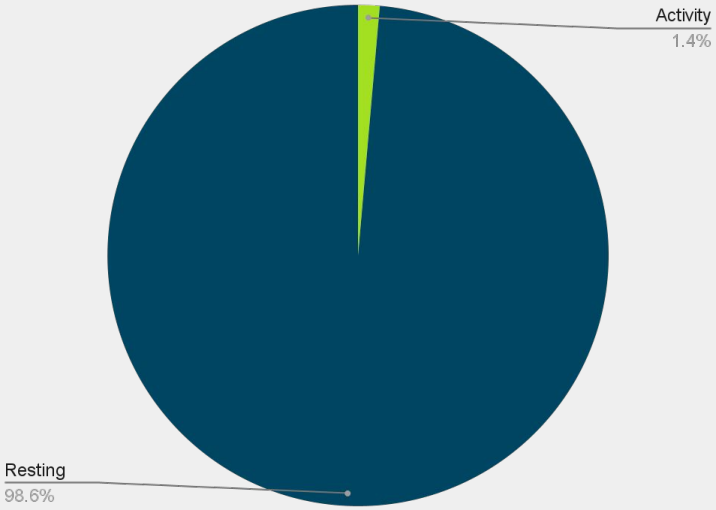
“It looks like you’re working out.”: *Using Machine Learning Classification Techniques to Predict Physical Stress*

Eva Schwartz

Have you ever looked down, and seen this even when you were not working out?





For average adults, the Center for Disease Control recommends “at least 150 minutes of moderate-intensity physical activity a week” but let’s put that into perspective…

150 minutes of recommended exercise in a week * 60 seconds in a minute

---

7 days in a week * 24 hours in a day * 60 minutes every hour * 60 seconds in every minute

… indicates that 1.4% of an average adult’s week should be dedicated to moderate-intensity physical activity

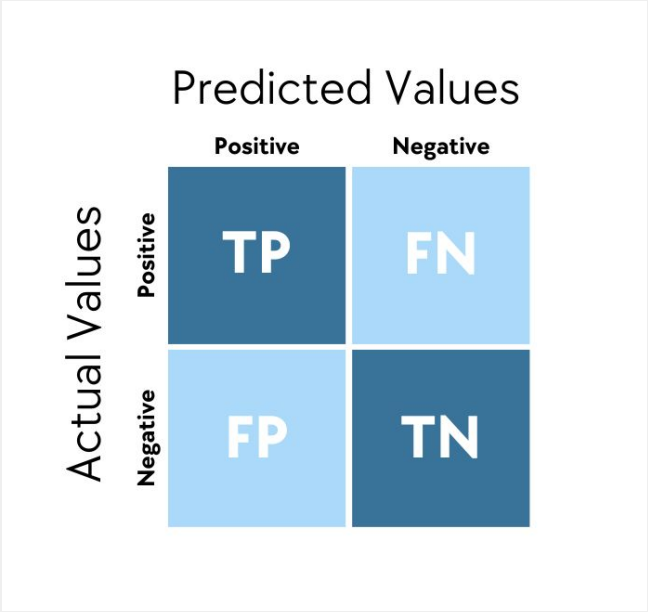
Given such a drastic imbalance in the class sizes of physical stress data, no wonder sometimes fitness wearables think that you are working out, even when you aren't.

As someone who loves training and working out, comparing the performance of machine learning classifiers in imbalanced physical stress datasets quickly emerged as the topic of my Data Science Capstone.

Agenda

1. Review of Machine Learning Model Evaluation Metrics
2. Related Work
3. Methods
4. Implementation
5. Results
6. Limitations and Alternate Market Applications

Review of Machine Learning Model Evaluation Metrics



To classify model evaluation metrics, a confusion matrix is vital in understanding the performance of models in classification tasks

For this project, the true positive category is the majority, resting, non-active class and the true negative class is the physical stress, minority class

$$\text{Accuracy} = \frac{\text{correct classifications}}{\text{total classifications}} = \frac{TP + TN}{TP + TN + FP + FN}$$

Useful, but in imbalanced datasets like the one used here, a false negative is more costly than a false negative, so metrics that are more focused can be more helpful

Focused on positive classifications, in this research, positives are the majority and are not the target class so not as helpful for this project

$$\text{Precision} = \frac{\text{correctly classified actual positives}}{\text{everything classified as positive}} = \frac{TP}{TP + FP}$$

$$\text{Recall (or TPR)} = \frac{\text{correctly classified actual positives}}{\text{all actual positives}} = \frac{TP}{TP + FN}$$

Also focused on positive classification, focused on reducing false negatives but the minority class for this data is the negative class and this metric is not as helpful in evaluating predictions for the negative class

Derives accuracy of negative predictions of models, inverse of precision, vital for this project as goal is predicting negative class

$$\text{Negative Predictive Value} = \frac{\text{correct negative predictions}}{\text{total predicted negatives}} = \frac{TN}{TN + FN}$$

$$\text{Specificity} = \frac{\text{correct negative classifications}}{\text{total actual negatives}} = \frac{TN}{TN + FP}$$

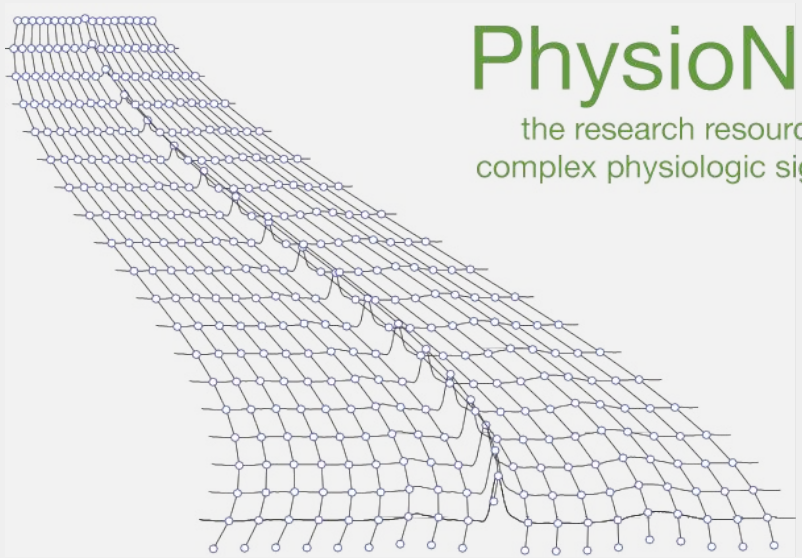
Focused on identifying how well a model identifies actual negative classes, very important in evaluating models in this research given the negative class is the minority in this project

Related Work

- One team working on credit card fraud detection used a dataset with positive, minority class accounting for .2% of their dataset, they recommended assigning larger class weights to fraudulent transactions that they were looking to predict to increase the relative importance of this class in the model training process
- A similar study focused on credit card fraud with .4% of values belonging to minority, fraudulent class used undersampling to balance data, then compared a decision tree, random forest, and logistic regression model with random forest having the strongest performance
- A study focused on emotional and physical stress found that heart rate and electrodermal activity were strong predictors of stress - both mental and physical

Methods

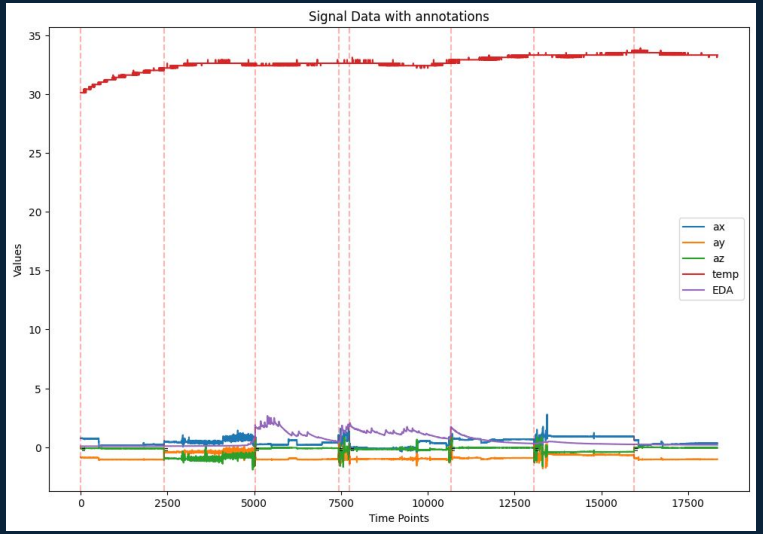
Dataset



- Data courtesy of PhysioNet and is publicly available
- Original features include electrodermal activity, temperature, acceleration, race, heart rate, and arterial oxygen level
    - The data were collected using noninvasive wrist-worn Empatica biosensors.
- Dataset consisted of 20 subjects, with 30% of the sample as female subjects, and a mean overall age of 26.05 years
- Data was collected in 7 stages
    - Stage 2 as physical stress stage, where subjects were instructed to “stand for one minute, walk on a treadmill at one mile per hour for two minutes, then walk/jog on the treadmill at three miles per hour for two minutes.”

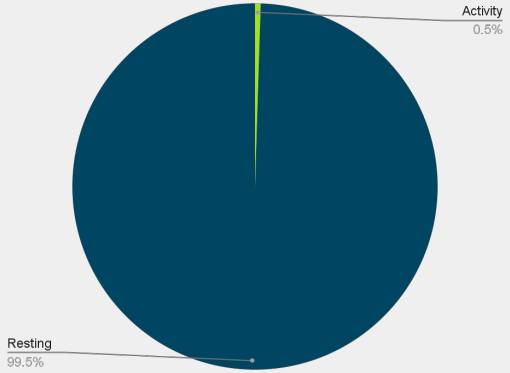
Signal processing and data processing

- Conversion of raw data signals to tabular data
- Synching of dataframes → HR/SP02 and acceleration, temperature, and EDA data
- Isolation forest outlier detection
    - Finds anomalies by detecting how far a given data point is from the rest of the data
    - Outlier detection is primarily exploration focused given the inherent variability of health data



Feature extraction

- Extraction of physical stress variable, with data imbalance seen in pie chart to the top right
- Acceleration magnitude extraction from acceleration variables as seen in formula in the bottom right
- Ratio of a subjects heart rate to their acceleration magnitude to indicate intensity of a subjects physical activity
- Ratio of body temperature to electrodermal activity given strong, complex relationship between these two variables

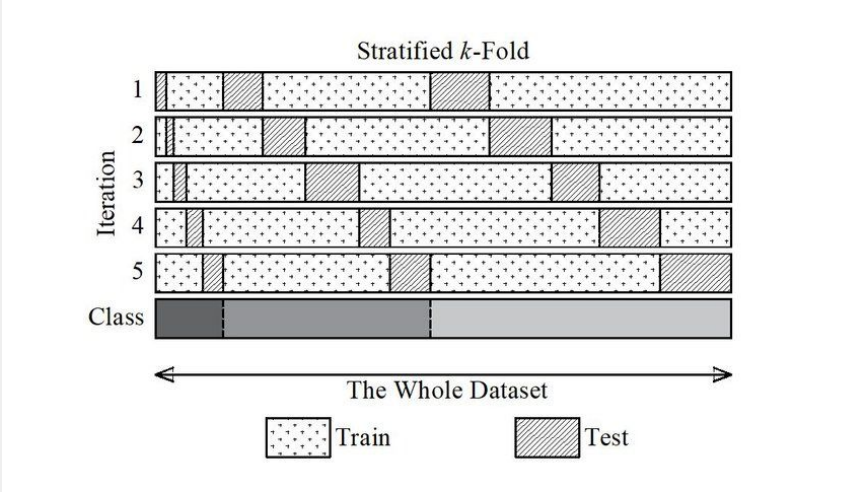


$$\|\vec{a}\| = \sqrt{a_x^2 + a_y^2 + a_z^2}$$

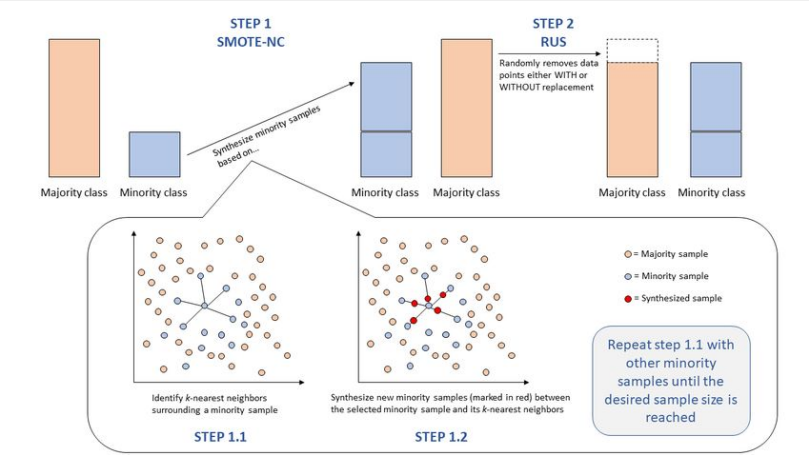
Experiments and Implementation

To perform robust cross-validation while ensuring each training model receives an even distribution of minority class data, I used stratified K-fold cross validation techniques.

K number of folds are initialized, data is split by class as seen in the bottom row of the image to the right, then data is sequentially added to each fold to ensure that cross validation techniques are robust and represent the original data distribution. I used 3 inner folds for tuning hyperparameters, then 5 outer folds for training and testing models.



SMOTETomek Hybrid Resampling

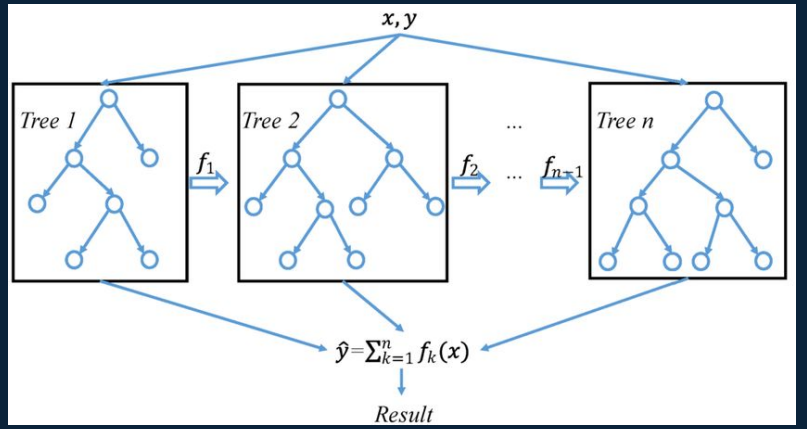


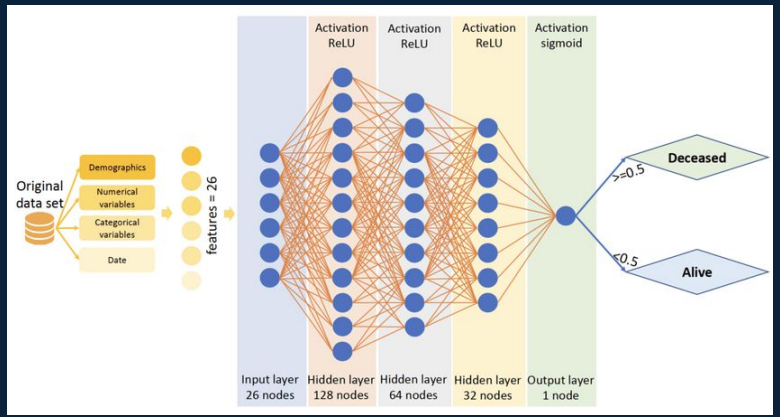
- Data is unbalanced so need to use resampling methods to augment minority class
- SMOTETomek hybrid resampling uses SMOTE resampling to synthetically generate minority class pointswhile using Tomek Link undersampling, as seen in the image to the left, to remove samples of the majority class using nearest neighbor techniques to find excess majority class points
- Following SMOTETomek resampling, MinMaxScaler was implemented in the pipeline to scale data given data was collected in different scales

Models Used

To find the best hyperparameters for each model, RandomizedSearchCV was used in the tuning step. Best performing hyperparameters for each fold of training can be found in the appendix.

- Logistic regression
- Random Forest
- XGBoost - architecture in top right
- Histogram Gradient Boosting
- Multi-Layer Perceptron - architecture in bottom right



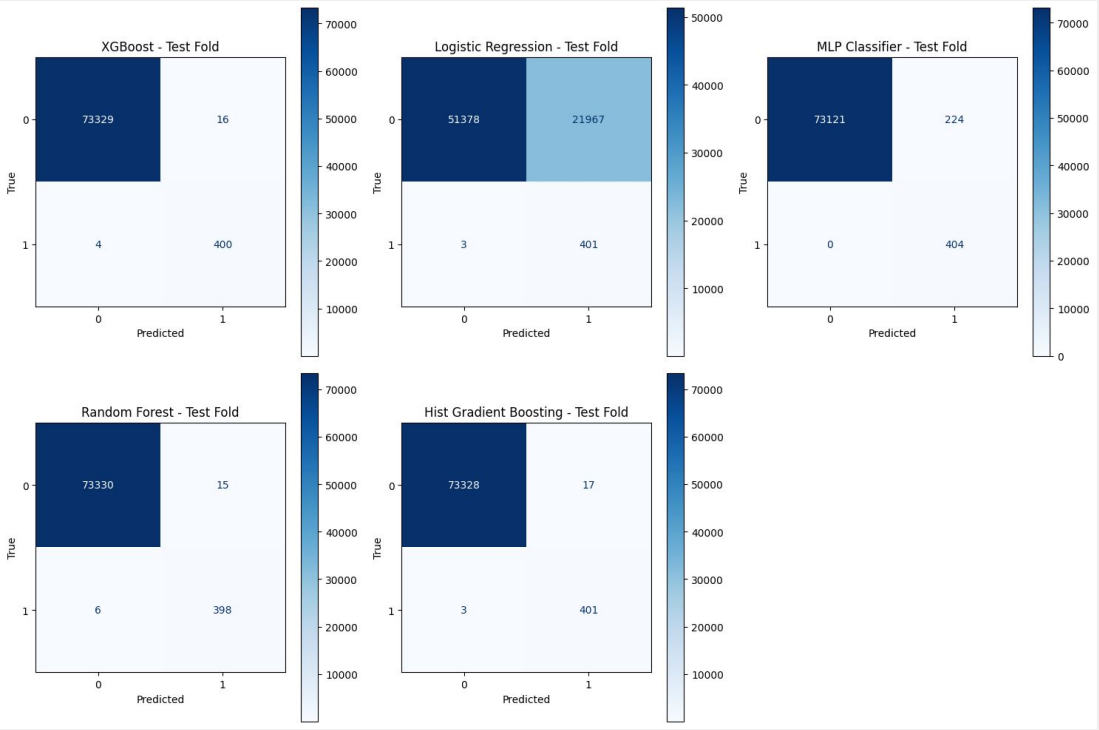


Results

Summary of model results

Model	Accuracy (mean ± std)	Recall (mean ± std)	Precision (mean ± std)	ROC AUC (mean ± std)	PR AUC (mean ± std)	Specificity (mean ± std)	NPV (mean ± std)
XGBoost	0.9934 ± 0.0018	0.9871 ± 0.0036	0.9537 ± 0.0071	1.0000 ± 0.0000	0.9970 ± 0.0013	0.9997 ± 0.0000	0.9999 ± 0.0000
Logistic Regression	0.8476 ± 0.0010	0.9852 ± 0.0056	0.0184 ± 0.0004	0.9196 ± 0.0030	0.0319 ± 0.0004	0.7100 ± 0.0072	0.9999 ± 0.0000
MLP Classifier	0.9981 ± 0.0005	0.9995 ± 0.0010	0.6297 ± 0.0209	0.9996 ± 0.0001	0.9301 ± 0.0250	0.9967 ± 0.0003	1.0000 ± 0.0000
Random Forest	0.9937 ± 0.0026	0.9876 ± 0.0052	0.9551 ± 0.0050	1.0000 ± 0.0000	0.9949 ± 0.0012	0.9997 ± 0.0000	0.9999 ± 0.0000
Hist Gradient Boosting	0.9951 ± 0.0028	0.9906 ± 0.0055	0.9414 ± 0.0104	1.0000 ± 0.0001	0.9971 ± 0.0011	0.9997 ± 0.0001	0.9999 ± 0.0000

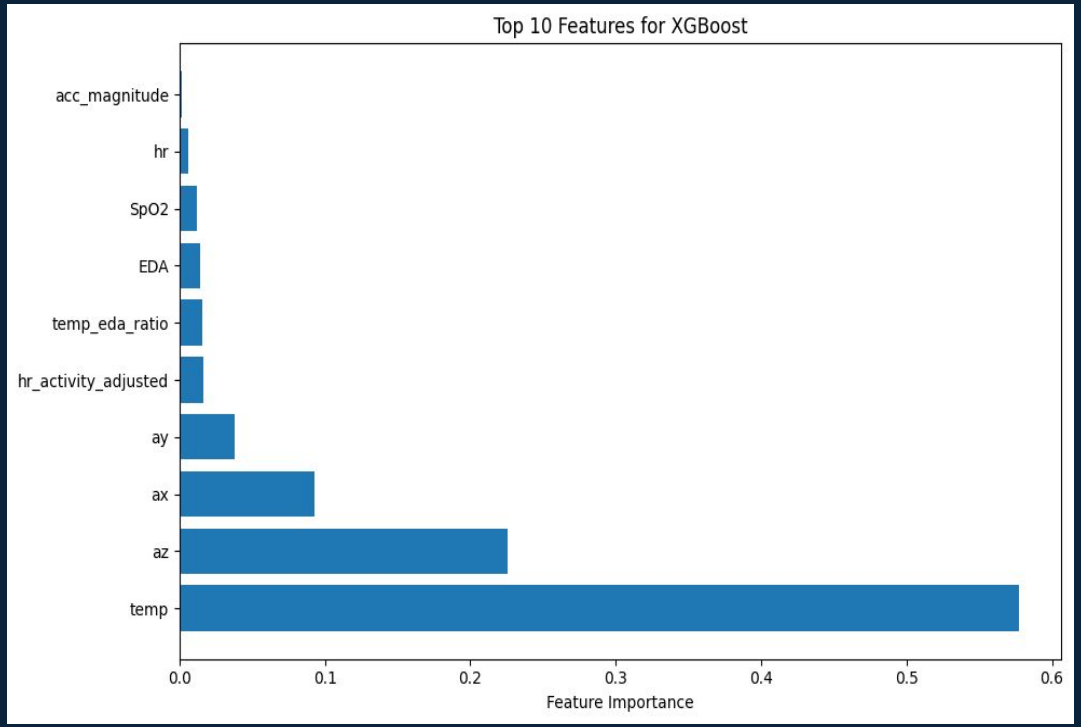
Random Forest and XGBoost both had strong specificity and negative predictive value metrics. The Multi-Layer Perceptron model had the highest negative predictive value, though the difference between models in this metric is very small. The weakness of Logistic Regression when comparing metrics is fairly clear, though this is to be expected given the simplicity of this model.



Darker blue shades indicate higher class counts here.

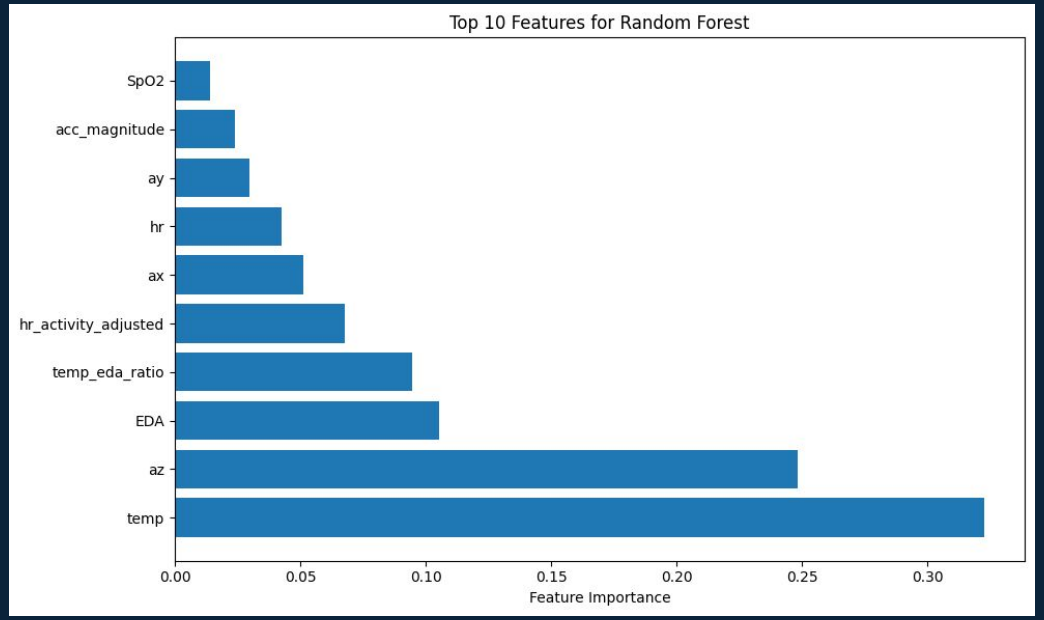
Confusion matrices align with conclusions from model evaluation metrics

Basic models like logistic regression suffer in classifying even the majority class, while models like MLP, Histogram Gradient Boosting, and XGBoost are strong in classifying true positives and true negatives



Body temperature indicates very strong importance in predicting physical stress for the XGBoost model, followed by z-axis acceleration, measuring forward and backward movement

Random Forest feature importance shows similar feature importance variables as XGBoost, though temperature's importance is less than XGBoost and EDA emerges as an additional important feature



Limitations and Alternate Applications

Limitations

- Lack of female representation in the dataset
- Inherent difficulties in predicting extremely imbalance datasets → some false positives sneak through even on very well developed models and products
- Multi-layer perceptron is a good model, but it operates as a “black box,” meaning there is not much information on how the hidden layers work

Alternate Applications

Model comparison research can be used in predicting cyber attacks, especially because cyber attacks also generally suffer from imbalanced data in training

This research could be applied to disease/medical prediction given the health focus of the model development and the focus on optimization of machine learning models for imbalanced datasets

Conclusion

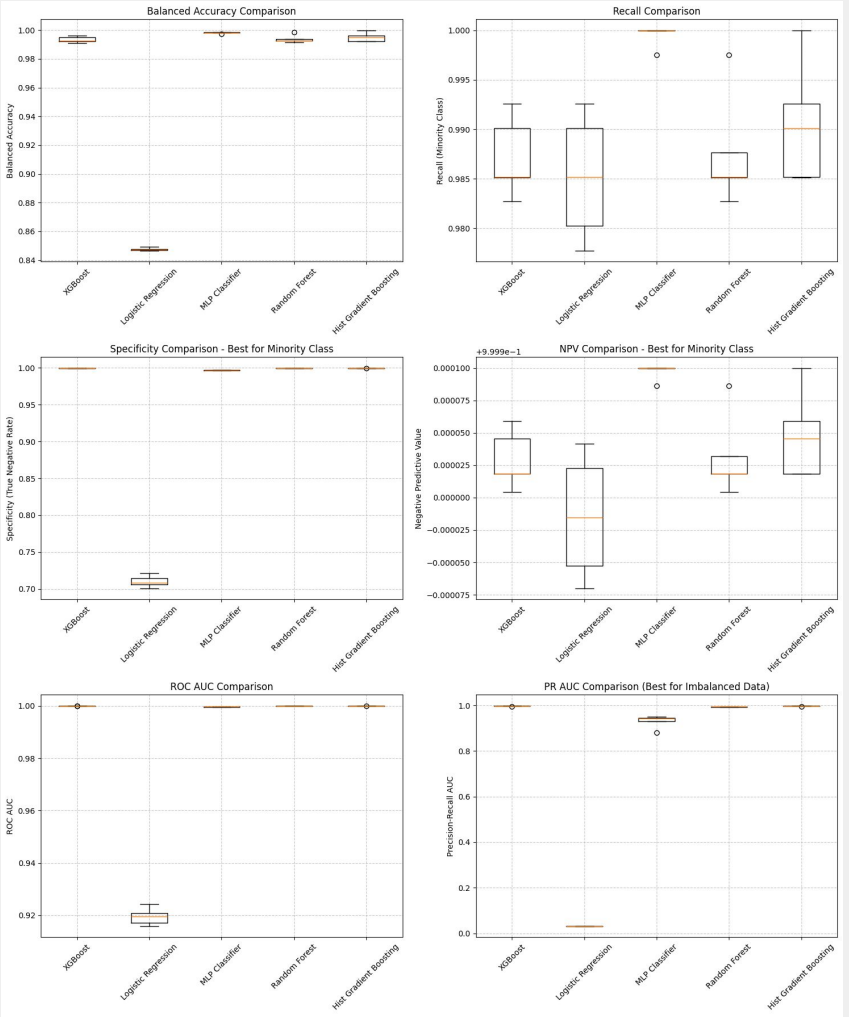
- Multi-layer Perceptron, Random Forest, and XGBoost are all strong models for this task
    - Implementing lighter, faster models like LightGBM for classifying physical stress could be a powerful next step for this body of research
- Temperature emerged as a very important factor in predicting physical stress across models therefore correctly calibrating and developing wearables sensors to accurately predict body temperature is vital in deploying this research to fitness wearables.
- An interesting exploration for this research would be to compare the results of these models to those gathered from a female-dominant biosignal dataset
    - Issue is lack of female representation in biosignal and physiological datasets

Q&A

Any questions?

Thank You!

Appendix 1: training model evaluation metrics



Appendix 2: Model hyperparameters by fold

Fold	Model	Hyperparameter	Best Hyperparameter	Fold				Fold			
1	XGBoost	subsample	1	3	XGBoost	subsample	1	5	XGBoost	subsample	1
1	XGBoost	number of estimators	500	3	XGBoost	number of estimators	500	5	XGBoost	number of estimators	500
1	XGBoost	min_child_weight	3	3	XGBoost	min_child_weight	3	5	XGBoost	min_child_weight	3
1	XGBoost	max_depth	5	3	XGBoost	max_depth	5	5	XGBoost	max_depth	5
1	XGBoost	learning rate	0.3	3	XGBoost	learning rate	0.3	5	XGBoost	learning rate	0.3
1	XGBoost	colsample_bytree	0.8	3	XGBoost	colsample_bytree	0.8	5	XGBoost	colsample_bytree	0.8
1	Logistic Regression	tol	1.00E-05	3	Logistic Regression	tol	1.00E-05	5	Logistic Regression	tol	1.00E-05
1	Logistic Regression	penalty	L2	3	Logistic Regression	penalty	l2	5	Logistic Regression	penalty	L2
1	Logistic Regression	C	100	3	Logistic Regression	C	100	5	Logistic Regression	C	100
1	MLP Classifier	learning rate init	0.001	3	MLP Classifier	learning rate init	0.01	5	MLP Classifier	learning rate init	0.01
1	MLP Classifier	hidden layer sizes	(100,)	3	MLP Classifier	hidden layer sizes	(50,)	5	MLP Classifier	hidden layer sizes	(50,)
1	MLP Classifier	alpha	0.0001	3	MLP Classifier	alpha	0.0001	5	MLP Classifier	alpha	0.0001
1	Random Forest	number of estimators	200	3	Random Forest	number of estimators	200	5	Random Forest	number of estimators	200
1	Random Forest	min_samples_split	5	3	Random Forest	min_samples_split	5	5	Random Forest	min_samples_split	5
1	Random Forest	min_samples_leaf	1	3	Random Forest	min_samples_leaf	1	5	Random Forest	min_samples_leaf	1
1	Random Forest	max_depth	None	3	Random Forest	max_depth	none	5	Random Forest	max_depth	none
1	Histogram Gradient Boosting	max_iterations	100	3	Histogram Gradient Boosting	max_iterations	200	5	Histogram Gradient Boosting	max_iterations	100
1	Histogram Gradient Boosting	max_depth	5	3	Histogram Gradient Boosting	max_depth	5	5	Histogram Gradient Boosting	max_depth	5
1	Histogram Gradient Boosting	learning_rate	0.3	3	Histogram Gradient Boosting	learning_rate	0.1	5	Histogram Gradient Boosting	learning_rate	0.3
2	XGBoost	subsample	1	4	XGBoost	subsample	1				
2	XGBoost	number of estimators	500	4	XGBoost	number of estimators	500				
2	XGBoost	min_child_weight	3	4	XGBoost	min_child_weight	3				
2	XGBoost	max_depth	5	4	XGBoost	max_depth	5				
2	XGBoost	learning rate	0.3	4	XGBoost	learning rate	0.3				
2	XGBoost	colsample_bytree	0.8	4	XGBoost	colsample_bytree	0.8				
2	Logistic Regression	tol	1.00E-05	4	Logistic Regression	tol	0.0001				
2	Logistic Regression	penalty	L2	4	Logistic Regression	penalty	l2				
2	Logistic Regression	C	100	4	Logistic Regression	C	100				
2	MLP Classifier	learning rate init	0.001	4	MLP Classifier	learning rate init	0.01				
2	MLP Classifier	hidden layer sizes	100	4	MLP Classifier	hidden layer sizes	(50,50)				
2	MLP Classifier	alpha	0.0001	4	MLP Classifier	alpha	0.0001				
2	Random Forest	number of estimators	200	4	Random Forest	number of estimators	200				
2	Random Forest	min_samples_split	5	4	Random Forest	min_samples_split	5				
2	Random Forest	min_samples_leaf	1	4	Random Forest	min_samples_leaf	1				
2	Random Forest	max_depth	None	4	Random Forest	max_depth	none				
2	Histogram Gradient Boosting	max_iterations	200	4	Histogram Gradient Boosting	max_iterations	200				
2	Histogram Gradient Boosting	max_depth	3	4	Histogram Gradient Boosting	max_depth	3				
2	Histogram Gradient Boosting	learning_rate	0.3	4	Histogram Gradient Boosting	learning_rate	0.3				

References for images

- Full reference can be accessed through this project's GitHub in the Full Report: https://github.com/GW-datasci/25-spring-ESCHWARTZ/tree/main
- A Study of Fitness Trackers and Wearables
- Hybrid resampling process. | Download Scientific Diagram