WIH 3001 DATA SCIENCE PROJECT

Employing HRV Analysis for Stress Level Detection

Data Product - HeartWise❤️‍🩹

Presented by Cher Jia Wen (U2005415)
Supervised by Assoc. Prof. Dr. Suraya Hamid

Table of Contents

01 Introduction

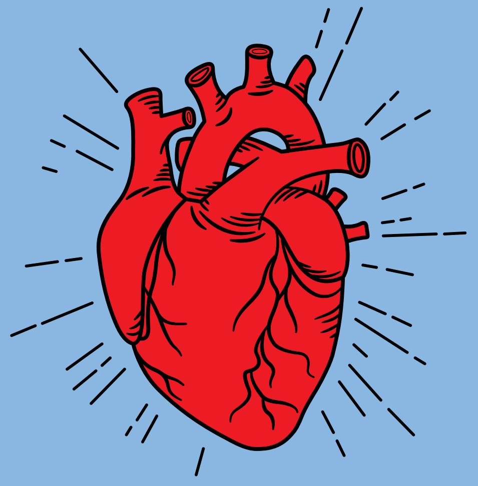
Problem Statements
Objectives

INTRODUCTION

GLOBAL IMPACT:

WHO has recognized stress as "***the health epidemic of the 21st century***" for several years.
(Fink, 2016)

CONSEQUENCES:



Stress

Mental or emotional strain
caused by challenging or demanding situations.

SYMPTOMS:

SELF-AWARENESS:

Recognizing the signs of stress helps in ***taking proactive measures*** such as adopting stress-reduction techniques or seeking professional help.

Stress in Malaysia

In May 2022, Rakuten Insight had conducted a mental health survey in Malaysia.

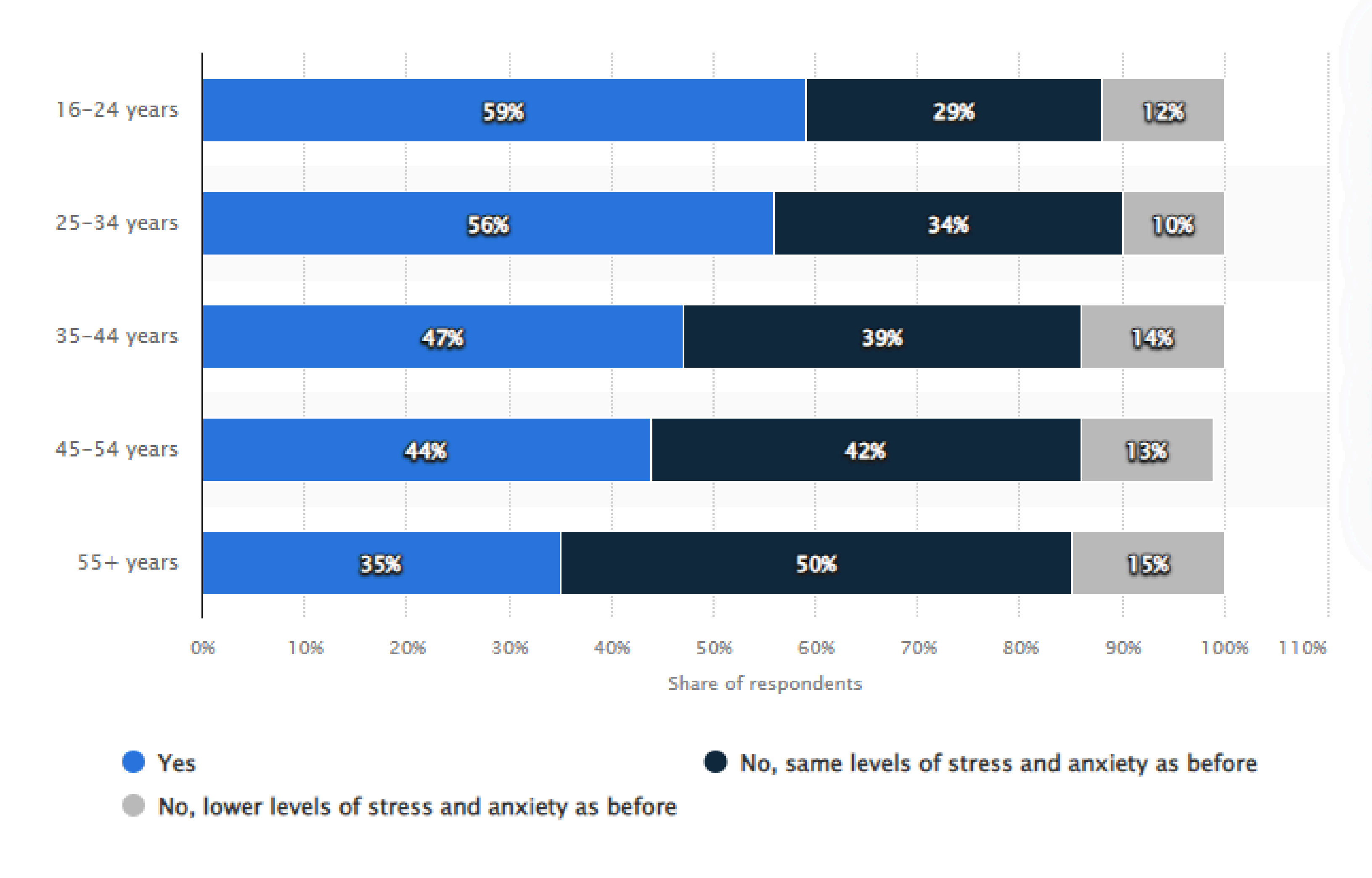


Image 1: Share of people suffering from higher level of stress or anxiety during the past year in Malaysia as of May 2022, by age group.

(Malaysia: Stress and Anxiety Level by Age 2022 | Statista, 2022)

59%

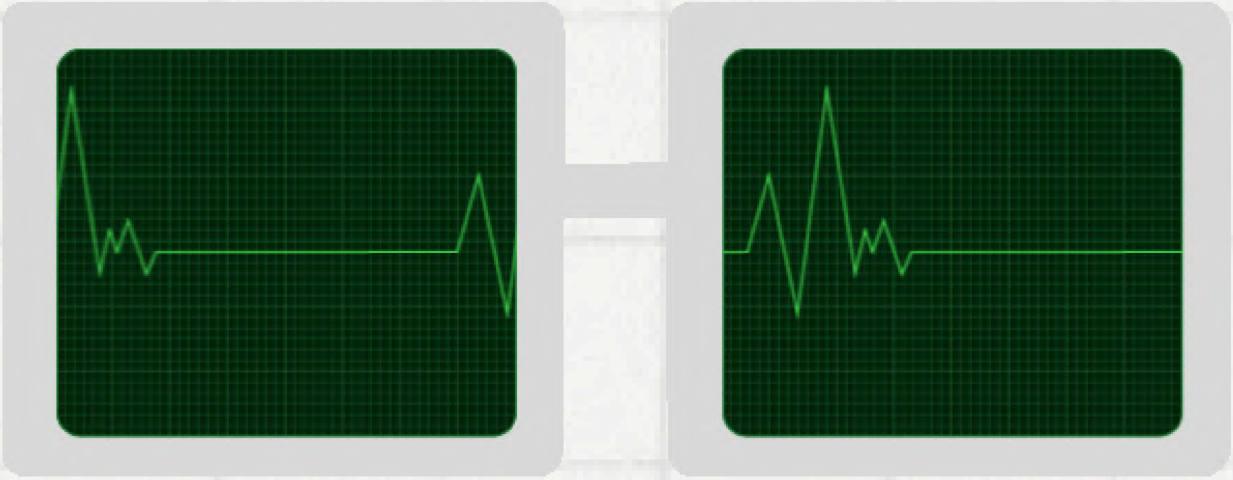
Individuals ***aged 16-24*** experienced ***increased*** in stress or anxiety in 2022.

WHAT CAN WE SEE FROM THE GRAPH?

1. Rising stress levels among young adults
2. Urban lifestyle as a contributing factor
3. Require targeted mental health interventions

Heart Rate Variability (HRV)

- An index of the functional status of the autonomic nervous system (ANS).
- Measures ***the time variance between consecutive heartbeats***, offering valuable insights into the body's response to stress. (Clinic, 2021)



HOW IS HRV LINKED TO STRESS?

- HRV is a non-invasive metric ***reflecting the autonomic nervous system's (ANS)*** influence on heart rhythms.
- ***Lower HRV is associated with higher stress levels***, making it a reliable indicator of physiological stress.
- ***Reduced HRV*** has been observed commonly in individuals indicating a ***poor stress response***. (Cleveland Clinic, n.d.)

WHAT IS THE ROLE OF HRV IN STRESS DETECTION?

By analyzing HRV data:

- Identify stress levels
- Enhancing self-awareness
- Transform stress management to proactive process

WHY IS HRV CHOSEN AS INDICATOR?

- ***Effectiveness*** in detecting stress levels.
  - HRV metrics undergo significant changes during stress episodes. (Kim et al., 2018)
  - Effective in distinguishing between the 'fight-or-flight' stress response and a relaxed state. (WebMD Editorial Contributors, 2021)
- ***Real-time monitoring*** without being over intrusive
- ***Practical and user-friendly*** for daily used.
  - HRV measurement through wearable technology like smartwatches.

PROBLEM STATEMENTS

Rising Global Stress Issue

Stress levels are rising worldwide and are particularly severe in Malaysia. There's a significant focus on addressing stress issues among the younger demographic in Malaysia.

Health Risks

WHO labels stress as a major health challenge linked to heart disease, anxiety, and depression. Poor management of stress might increase the risks of having physically and psychological issues.

Impractical Existing Methods

Existing methods for stress detection and management are often not real-time and may be invasive or impractical.

OBJECTIVES

- To ***identify the best machine learning algorithm*** to detect stress levels using HRV analysis.

- To ***develop a machine learning model*** capable of analyzing HRV data to detect stress levels.

- To ***deploy a user-friendly web application*** of the HRV-based stress level detection system.

02
Data Science Methodology

Analysis
Modelling
Codes
Tools

Project Workflow

DS Methodology

01 Problem Understanding

02 Data Acquisition

03 Data Exploration & Preprocessing

04 Exploratory Data Analysis (EDA)

05 Model Development

06 Model Evaluation

07 Deployment

Guided by the CRISP-DM (Cross-Industry Standard Process for Data Mining) model

01 PROBLEM UNDERSTANDING

- **Access current situation**: Stress impact and research on HRV.
- Define the **problem statements** and **objectives.**
- Identify **potential ML techniques** by reviewing related work paper.

ARTICLE SUMMARY TABLE

Author(s), year, title	Aim of Study	ML Algorithms used for Stress Detection	Results
Yekta Said Can, Arnrich, B., & Ersoy, C. (2019). Stress detection in daily life scenarios using smart phones and wearable sensors: A survey. *Journal of Biomedical Informatics*, *92*, 103139–103139. https://doi.org/10.1016/j.jbi.2019.103139	• To review stress detection research in different environments, including daily life. • To discuss devices, features, and stress alleviation methods. • To identify research directions and open problems in stress detection.	1. Support Vector Machine (SVM) 2. k-Nearest Neighbors (kNN) 3. Linear Discriminant Analysis (LDA)	The studies showed varying degrees of success, with some achieving high accuracy in detecting stress using physiological and behavioral data. However, real-world application outside controlled environments remains challenging.
Mouna Benchekroun, Pedro Elkind Velmovitsky, Istrate, D., Zalc, V., Plinio Pelegrini Morita, & Lenné, D. (2023). Cross Dataset Analysis for Generalizability of HRV-Based Stress Detection Models. *Sensors*, *23*(4), 1807–1807. https://doi.org/10.3390/s23041807	To evaluate the generalizability of Heart Rate Variability (HRV)-based machine learning models for stress detection using two different datasets.	1. Logistic Regression 2. Random Forest - Includes both independent dataset analysis and cross-dataset validation	The results indicate that while both models perform well, the Random Forest model demonstrates better generalization capabilities with a stable F1 score of 61%, suggesting its effectiveness in HRV-based stress detection models for various applications. The paper's focus is on the adaptability of these models to different datasets and real-world scenarios, contributing to mental health and medical research.
Yekta Said Can, Dilara Gokay, Dilruba Reyyan Kılıç, Deniz Ekiz, Niaz Chalabianloo, & Ersoy, C. (2020). How Laboratory Experiments Can Be Exploited for Monitoring Stress in the Wild: A Bridge Between Laboratory and Daily Life. *Sensors*, *20*(3), 838–838. https://doi.org/10.3390/s20030838	To develope more effective stress detection system that can function reliably in real-world settings and investigate the impact of different data labeling techniques on model accuracy.	The study utilizes machine learning algorithms, but specific models are not detailed in the visible portion of the paper.	The paper discusses the challenges and potential solutions in stress detection, especially regarding the accuracy and reliability of models in real-life scenarios compared to controlled lab environments. However, specific results or comparative analysis of different models are not detailed in the visible section. The best model and its specific results are not discussed in the portion of the paper that is visible.

02 DATA ACQUISITION

A Kaggle dataset with **HRV features under 3 levels of stress levels** has been used for this project.

Dataset	**SWELL dataset**
Source	Kaggle
Usage	1. S. Koldijk, M. A. Neerincx, and W. Kraaij, "Detecting Work Stress in Offices by Combining Unobtrusive Sensors," IEEE Trans. Affect. Comput., vol. 9, no. 2, pp. 227–239, 2018. 2. S. Koldijk, M. Sappelli, S. Verberne, M. A. Neerincx, and W. Kraaij, "The SWELL Knowledge Work Dataset for Stress and User Modeling Research," Proc. 16th Int. Conf. Multimodal Interact. - ICMI '14, pp. 291–298, 2014.

Details	• *36 columns (Only 10 columns are used)* • *369290 rows for train dataset;* • *41034 rows for test dataset (9:1)*
Variables	• *MEAN_RR , MEDIAN_RR, SDRR, RMSSD, SDSD, SDRR_RMSSD, HR, pNN25, pNN50, condition*

	MEAN_RR	MEDIAN_RR	SDRR	RMSSD	SDSD	SDRR_RMSSD	HR	pNN25	pNN50	condition
0	885.157845	853.763730	140.972741	15.554505	15.553371	9.063146	69.499952	11.133333	0.533333	Normal Stress Level
1	939.425371	948.357865	81.317742	12.964439	12.964195	6.272369	64.363150	5.600000	0.000000	Overload
2	898.186047	907.006860	84.497236	16.305279	16.305274	5.182201	67.450066	13.066667	0.200000	Overload
3	881.757865	893.460030	90.370537	15.720468	15.720068	5.748591	68.809562	11.800000	0.133333	Normal Stress Level
4	809.625331	811.184865	62.766242	19.213819	19.213657	3.266724	74.565728	20.200000	0.200000	Normal Stress Level
5	923.283866	617.794160	517.536544	9.965976	9.933933	51.930344	81.342254	1.200000	0.600000	Normal Stress Level

DATA DICTIONARY:

	Feature	Description
0	MEAN_RR	Mean of RR intervals (milliseconds)
1	MEDIAN_RR	Median of RR intervals (milliseconds)
2	SDRR	Standard deviation of RR intervals (milliseconds)
3	RMSSD	Root Mean Square of Successive Differences between RR intervals (milliseconds)
4	SDSD	Standard deviation of Successive Differences between RR intervals (milliseconds)
5	SDRR_RMSSD	Ratio of SDRR to RMSSD
6	HR	Heart rate (beats per minute)
7	pNN25	Percentage of RR intervals >25 ms different from the previous (short-term HRV)
8	pNN50	Percentage of RR intervals >50 ms different from the previous (short-term HRV)
9	RR	Time between consecutive heartbeats, derived from ECG signal (milliseconds)

	Level	Condition
0	Level 1	Normal Stress Level: Normal or relaxed state
1	Level 2	Attention Needed: Moderate stress level
2	Level 3	Overload: High stress level, immediate intervention may be necessary

03 DATA EXPLORATION & PRE-PROCESSING

- **Data exploration**: Analyze the properties of the data: distribution of HRV values.

	MEAN_RR	MEDIAN_RR	SDRR	RMSSD	SDSD	SDRR_RMSSD	HR	pNN25	pNN50
count	369289.000000	369289.000000	369289.000000	369289.000000	369289.000000	369289.000000	369289.000000	369289.000000	369289.000000
mean	846.650104	841.965890	109.352531	14.977498	14.976767	7.396597	73.941824	9.841143	0.866001
std	124.603984	132.321005	77.117025	4.120766	4.120768	5.143834	10.337453	8.195574	0.990189
min	547.492221	517.293295	27.233947	5.529742	5.529630	2.660381	48.737243	0.000000	0.000000
25%	760.228533	755.750735	64.205641	11.830959	11.830671	4.541896	66.715776	3.666667	0.000000
50%	822.951438	819.689595	82.608243	14.415918	14.415388	5.952112	74.217809	7.600000	0.466667
75%	924.117422	916.821570	118.237002	17.927144	17.924839	7.919841	80.334937	13.333333	1.466667
max	1322.016957	1653.122250	563.486949	26.629477	26.629392	54.523950	113.752309	39.400000	5.466667

```
Data columns (total 10 columns):
 #   Column      Non-Null Count   Dtype
---  ------      --------------   -----
 0   MEAN_RR     369289 non-null  float64
 1   MEDIAN_RR   369289 non-null  float64
 2   SDRR        369289 non-null  float64
 3   RMSSD       369289 non-null  float64
 4   SDSD        369289 non-null  float64
 5   SDRR_RMSSD  369289 non-null  float64
 6   HR          369289 non-null  float64
 7   pNN25       369289 non-null  float64
 8   pNN50       369289 non-null  float64
 9   condition   369289 non-null  object
dtypes: float64(9), object(1)
```

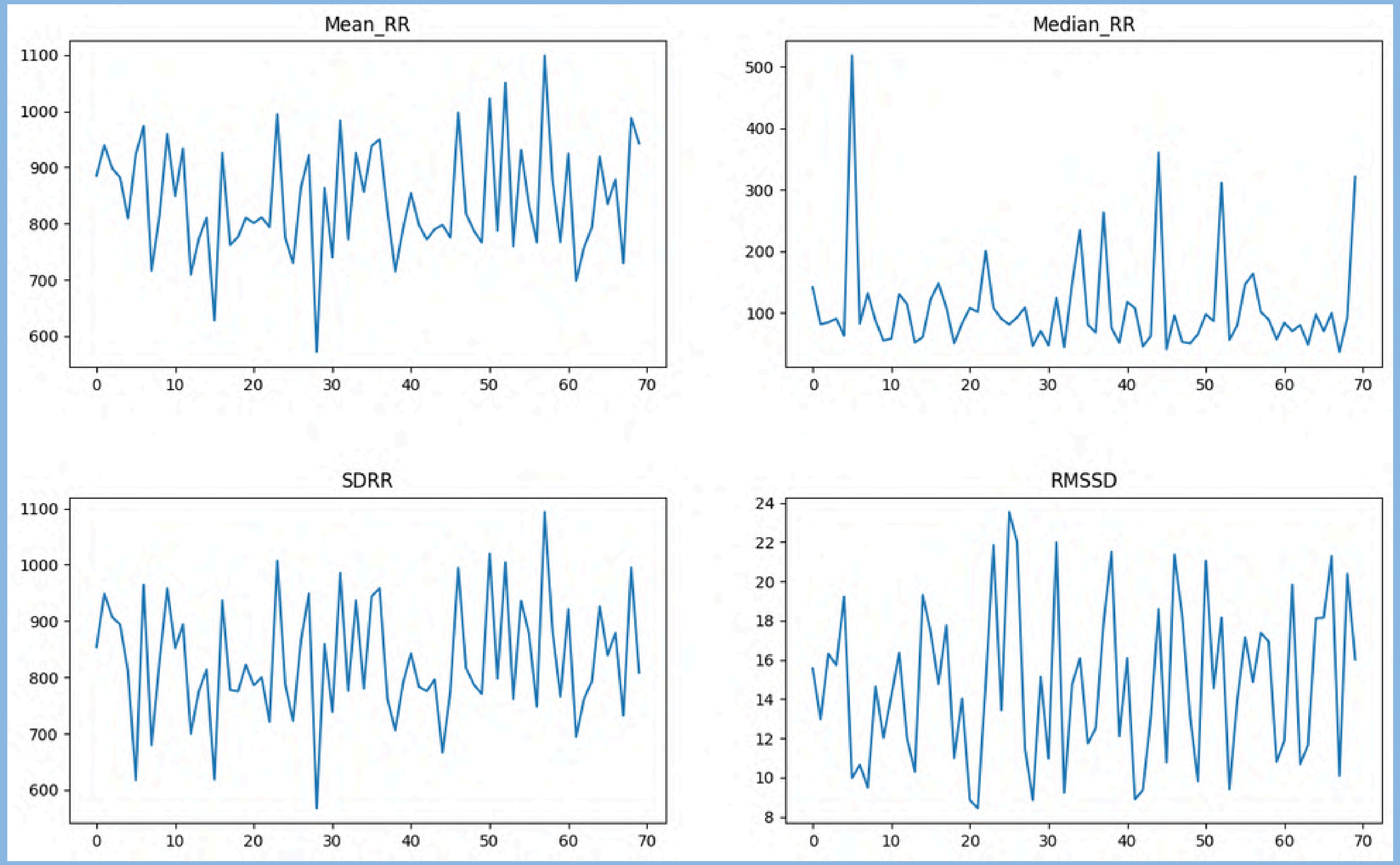
- **Data cleaning**: Handle missing data
- **Data transformation:** Scale HRV data, encode data

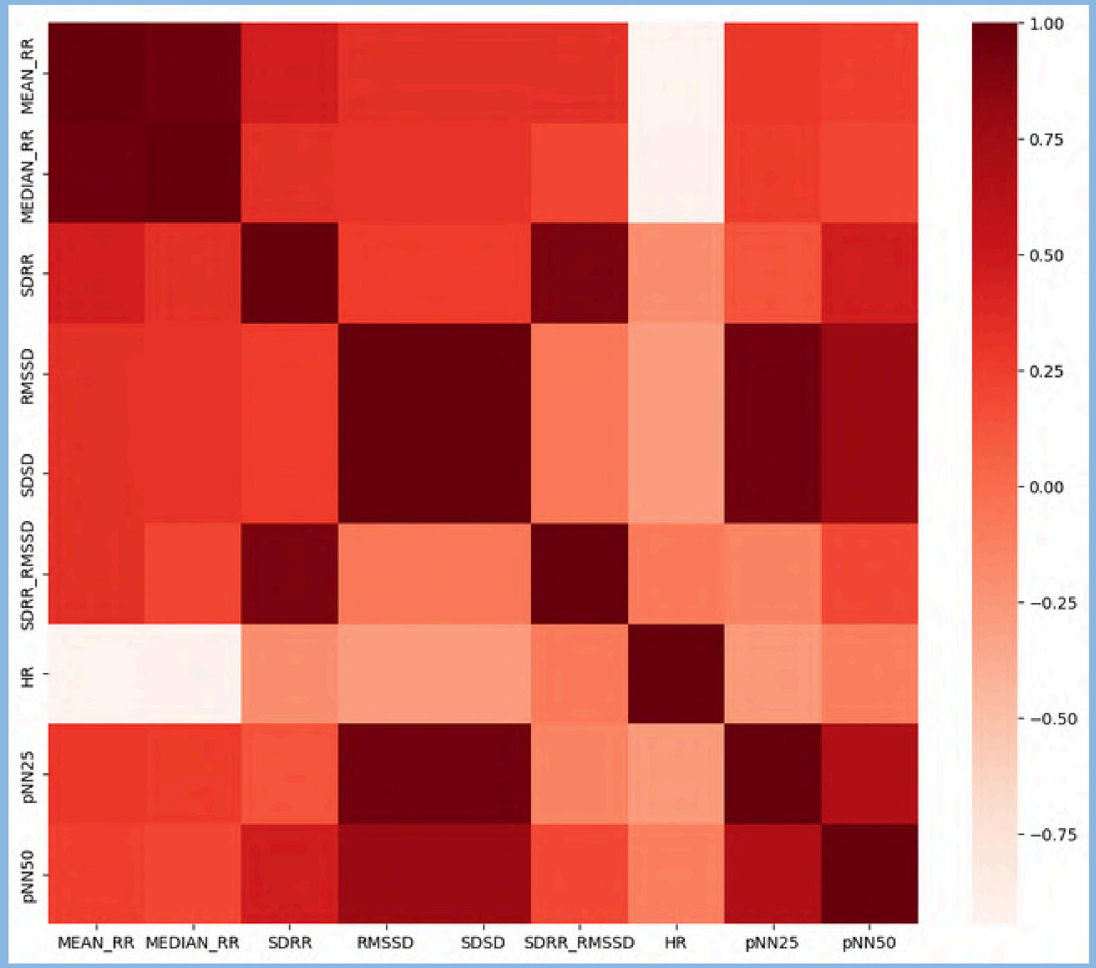
```python
# For normalizing/scaling
scaler = StandardScaler()

# For encoding
labelencoder = LabelEncoder()
```

```python
# For train dataset
# Encode the 'condition' column
train_encoded_condition = labelencoder.fit_transform(train['condition'])

# Handle missing values
train.fillna(train.mean(), inplace=True)

# Normalize the data except 'condition' column
train_features = train.drop('condition', axis=1)
train_scaled_features = scaler.fit_transform(train_features)

# Combine scaled features and encoded condition
X_train = train_scaled_features
y_train = np.array(train_encoded_condition)
```

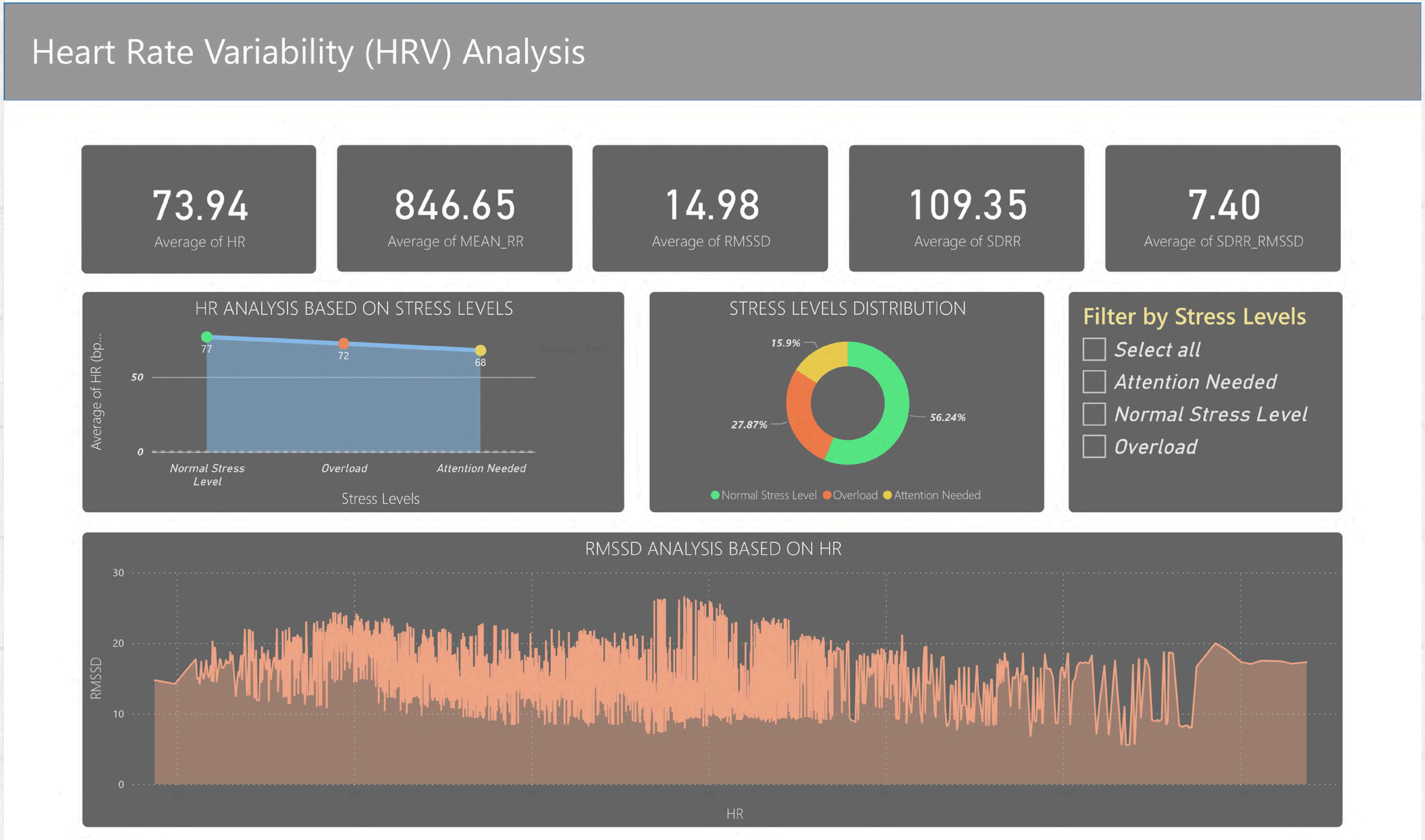
04 EXPLORATORY DATA ANALYSIS (EDA)

- **Statistical analysis**: Perform statistical tests to understand relationships and patterns in the data.
- **Visualization**: Create plots (histograms, scatter plots) to visualize trends and patterns in HRV related to stress levels.





Dashboard: HRV Analysis



https://app.powerbi.com/groups/me/reports/2a698d48-a389-42be-9cc6-fff2e8a67ae5/ReportSection?experience=power-bi

05 MODEL DEVELOPMENT

Classification

- Supervised ML method where the model tries to predict the correct label of a given input data.

After reviewing several related work research paper, these are some of the **ML algorithms (classifier)** used :

- Random Forest
- Logistic Regression
- K-Nearest Neighbors (KNN)
- Support Vector Machine(SVM)
- Decision Tree
- Gradient Boosting
- Neural Network (1-layer)

Target Variables	*condition*
Input Variables	*MEAN_RR* *MEDIAN_RR* *SDRR* *RMSSD* *SDSD* *SDRR_RMSSD* *HR* *pNN25* *pNN50*

06 MODEL EVALUATION

- **Evaluation Metrics:** Precision, Recall, F1-Score, Accuracy

Precision	Ratio of correctly predicted positive observations to the total predicted positives. High precision relates to a low false positive rate.
Recall	Ratio of correctly predicted positive observations to all actual positives. High recall indicates most positive cases are caught.
F1-Score	The weighted average of Precision and Recall. This score takes both false positives and false negatives into account.
Accuracy	A metric that measures how often a machine learning model correctly predicts the outcome

Random Forest:

```
# Create the Random Forest model
rf_def_model = RandomForestClassifier(random_state=42)

# Train the Random Forest model
rf_def_model.fit(X_train, y_train)

# Predict on the test set
rf_def_predictions = rf_def_model.predict(X_test)

# Random Forest Model Evaluation
print("Random Forest Classifier (Default):")
print(classification_report(y_test, rf_def_predictions))
print("Accuracy:", accuracy_score(y_test, rf_def_predictions))
```

```
Random Forest Classifier (Default):
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     11782
           1       1.00      1.00      1.00     22158
           2       1.00      1.00      1.00      7093

    accuracy                           1.00     41033
   macro avg       1.00      1.00      1.00     41033
weighted avg       1.00      1.00      1.00     41033

Accuracy: 0.9999756293714815
```

OVERFITTING ?

Logistic Regression:

```
# Create the Logistic Regression model
logistic_model = LogisticRegression(random_state=42)

# Train Logistic Regression model
logistic_model.fit(X_train, y_train)

# Predict on the test set
logistic_predictions = logistic_model.predict(X_test)

# Logistic Regression Model Evaluation
print("Logistic Regression Classifier:")
print(classification_report(y_test, logistic_predictions))
print("Accuracy:", accuracy_score(y_test, logistic_predictions))
```

```
Logistic Regression Classifier:
              precision    recall  f1-score   support

           0       0.57      0.31      0.40     11782
           1       0.61      0.87      0.72     22158
           2       0.46      0.21      0.29      7093

    accuracy                           0.59     41033
   macro avg       0.55      0.46      0.47     41033
weighted avg       0.57      0.59      0.55     41033

Accuracy: 0.5933273219116321
```

K-Nearest Neighbors:

```
# Create the KNN model with adjusted number of neighbors
knn_model = KNeighborsClassifier(n_neighbors=20, weights='distance')

# Train KNN model
knn_model.fit(X_train_scaled, y_train)

# Predict on the test set
knn_predictions = knn_model.predict(X_test_scaled)

# KNN Model Evaluation
print("Adjusted K-Nearest Neighbors Classifier:")
print(classification_report(y_test, knn_predictions))
print("Accuracy:", accuracy_score(y_test, knn_predictions))

# Optional: Evaluate using cross-validation
cv_scores = cross_val_score(knn_model, X_train_scaled, y_train, cv=5)
print("Average cross-validation score:", np.mean(cv_scores))
```

```
Adjusted K-Nearest Neighbors Classifier:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     11782
           1       1.00      1.00      1.00     22158
           2       1.00      1.00      1.00      7093

    accuracy                           1.00     41033
   macro avg       1.00      1.00      1.00     41033
weighted avg       1.00      1.00      1.00     41033

Accuracy: 0.9998537762288889
Average cross-validation score: 0.999834817651289
```

OVERFITTING ?

SVM:

```
from sklearn.svm import LinearSVC

# Create SVM model with LinearSVC for faster execution
svm_model = LinearSVC(random_state=42, tol=1e-4, max_iter=1000)

# Train SVM model
svm_model.fit(X_train, y_train)

# SVM Evaluation
svm_predictions = svm_model.predict(X_test)
print("Support Vector Machine Classifier:")
print(classification_report(y_test, svm_predictions))
print("Accuracy:", accuracy_score(y_test, svm_predictions))
```

```
Support Vector Machine Classifier:
              precision    recall  f1-score   support

           0       0.57      0.32      0.41     11782
           1       0.62      0.89      0.73     22158
           2       0.52      0.19      0.28      7093

    accuracy                           0.60     41033
   macro avg       0.57      0.47      0.47     41033
weighted avg       0.59      0.60      0.56     41033

Accuracy: 0.6031486852045914
```

Decision Tree

```
# Decision Tree Classifier
decision_tree_model = DecisionTreeClassifier(random_state=42)
decision_tree_model.fit(X_train, y_train)
decision_tree_predictions = decision_tree_model.predict(X_test)

# Model Evaluation
print("Decision Tree Classifier:")
print(classification_report(y_test, decision_tree_predictions))
print("Accuracy:", accuracy_score(y_test, decision_tree_predictions))
```

Decision Tree Classifier:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	11782
1	1.00	1.00	1.00	22158
2	1.00	1.00	1.00	7093
accuracy			1.00	41033
macro avg	1.00	1.00	1.00	41033
weighted avg	1.00	1.00	1.00	41033

Accuracy: 0.9994394755440743

OVERFITTING ?

Gradient Boosting

```
# Gradient Boosting Classifier
gradient_boost_model = GradientBoostingClassifier(random_state=42)
gradient_boost_model.fit(X_train, y_train)
gradient_boost_predictions = gradient_boost_model.predict(X_test)


print("\nGradient Boosting Classifier:")
print(classification_report(y_test, gradient_boost_predictions))
print("Accuracy:", accuracy_score(y_test, gradient_boost_predictions))
```

Gradient Boosting Classifier:

	precision	recall	f1-score	support
0	0.87	0.73	0.80	11782
1	0.83	0.94	0.88	22158
2	0.83	0.70	0.76	7093
accuracy			0.84	41033
macro avg	0.84	0.79	0.81	41033
weighted avg	0.84	0.84	0.83	41033

Accuracy: 0.8372529427533936

Lightweight Neural Network (1 layer)

```
# Lightweight Neural Network
# Assuming a small network with one hidden layer of 100 neurons
neural_network_model = MLPClassifier(hidden_layer_sizes=(100,), random_state=42)
neural_network_model.fit(X_train, y_train)
neural_network_predictions = neural_network_model.predict(X_test)

print("\nLightweight Neural Network:")
print(classification_report(y_test, neural_network_predictions))
print("Accuracy:", accuracy_score(y_test, neural_network_predictions))
```

Lightweight Neural Network:

	precision	recall	f1-score	support
0	0.97	0.98	0.98	11782
1	0.99	0.98	0.98	22158
2	0.97	0.97	0.97	7093
accuracy			0.98	41033
macro avg	0.98	0.98	0.98	41033
weighted avg	0.98	0.98	0.98	41033

Accuracy: 0.9789194063314893

OVERFITTING ?

INTERPRET THE MODELS

High Risk of Overfitting :

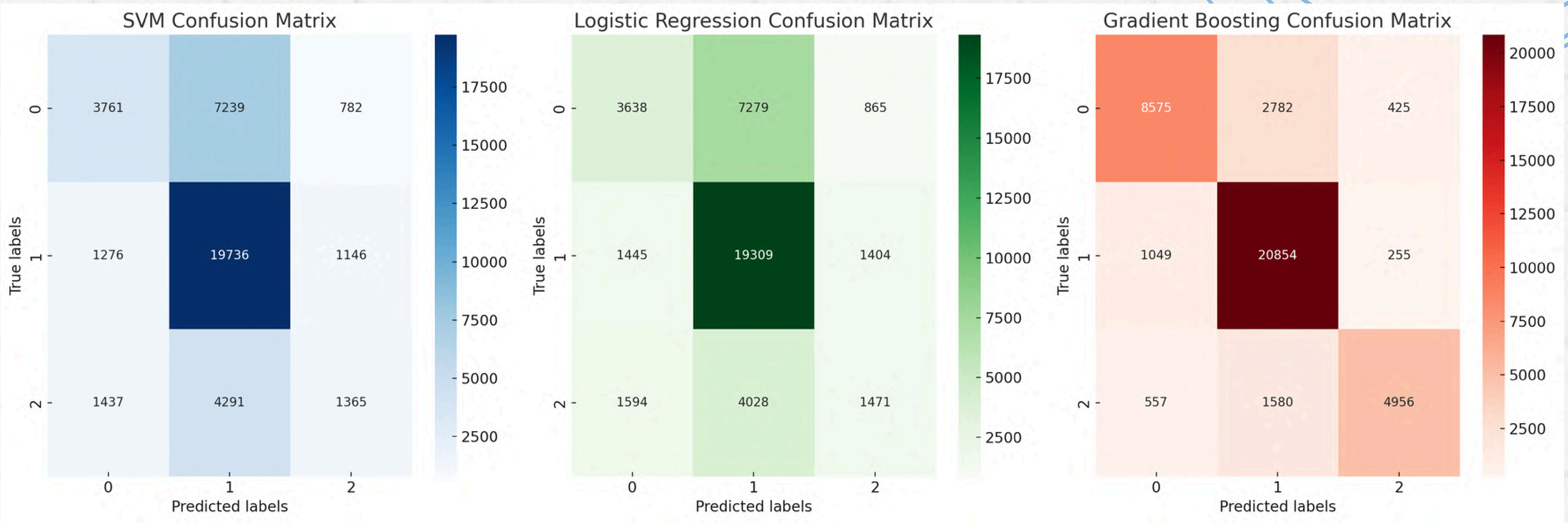
- Occurs when a model learns the training data too well, including its noise and outliers, and fails to generalize to new, unseen data.
- Models with extremely high accuracy and F1-scores:
  - KNN
  - Decision Tree
  - Neural Network Models
  - Random Forest

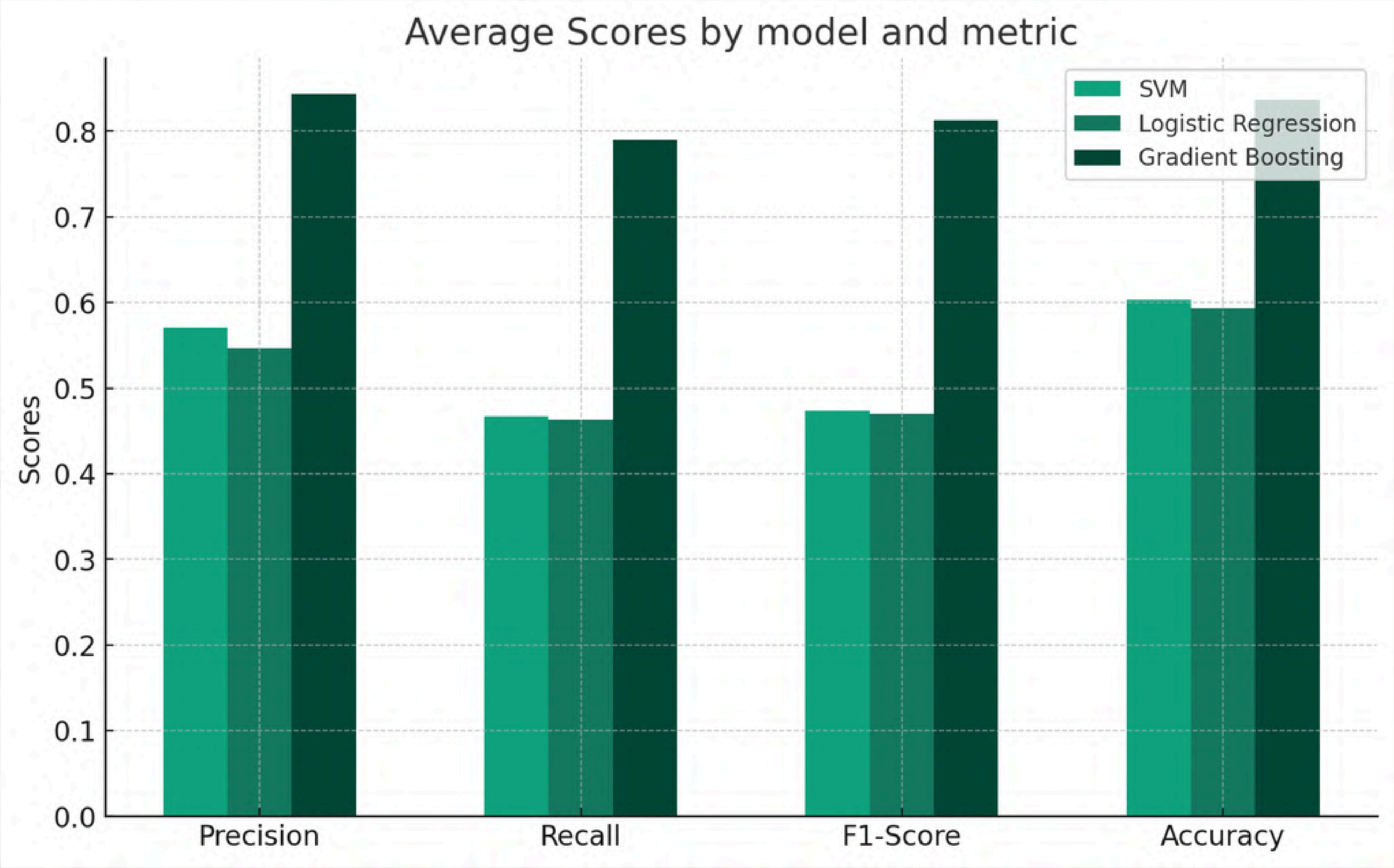
Optimum Models:

Compare the performance for non-overfitting models:

1. Logistic Regression
2. SVM
3. Gradient Boosting

- Visualize the scores with bar chart
- Utilize confusion matrix - to access the performance of models in imbalanced datasets if there is class dominating over others.





Gradient Boosting

- High accuracy
- Outperforms in terms of precision, recall and F1-score across all classes.

07 MODEL DEPLOYMENT

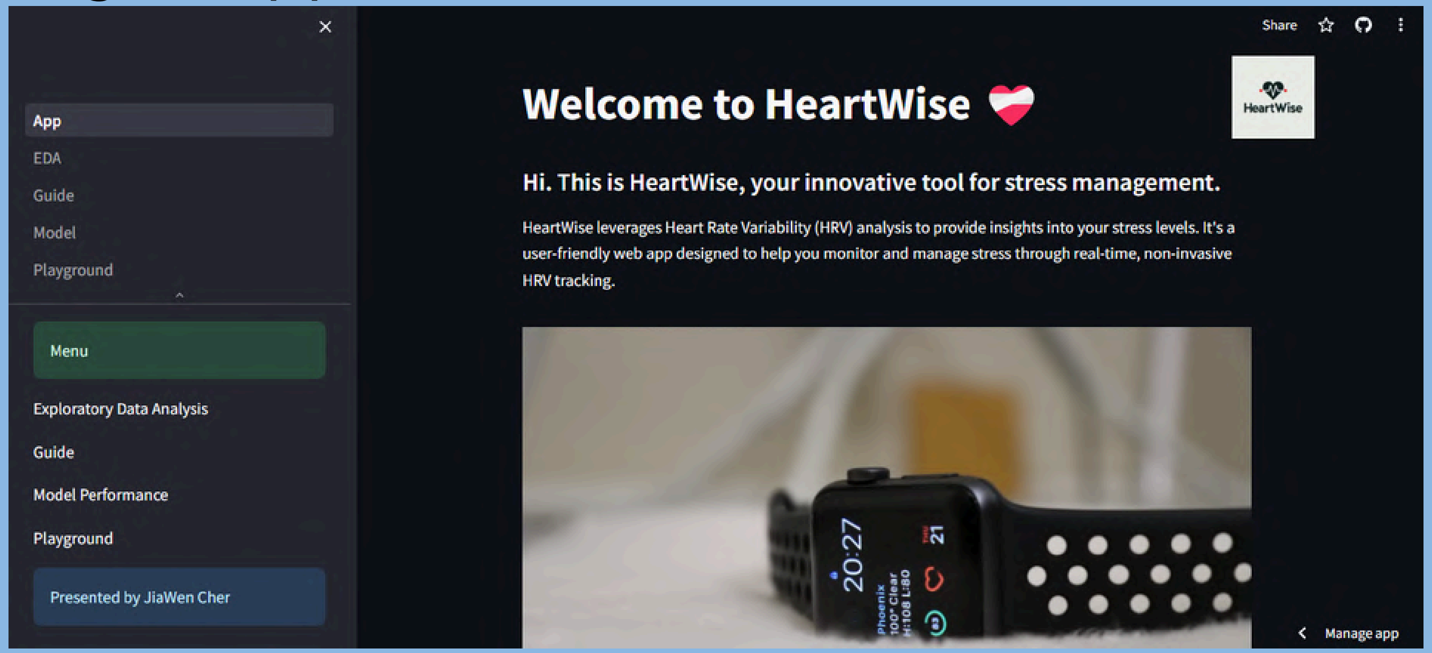
Hosted using: Streamlit

Web App: [HeartWise]

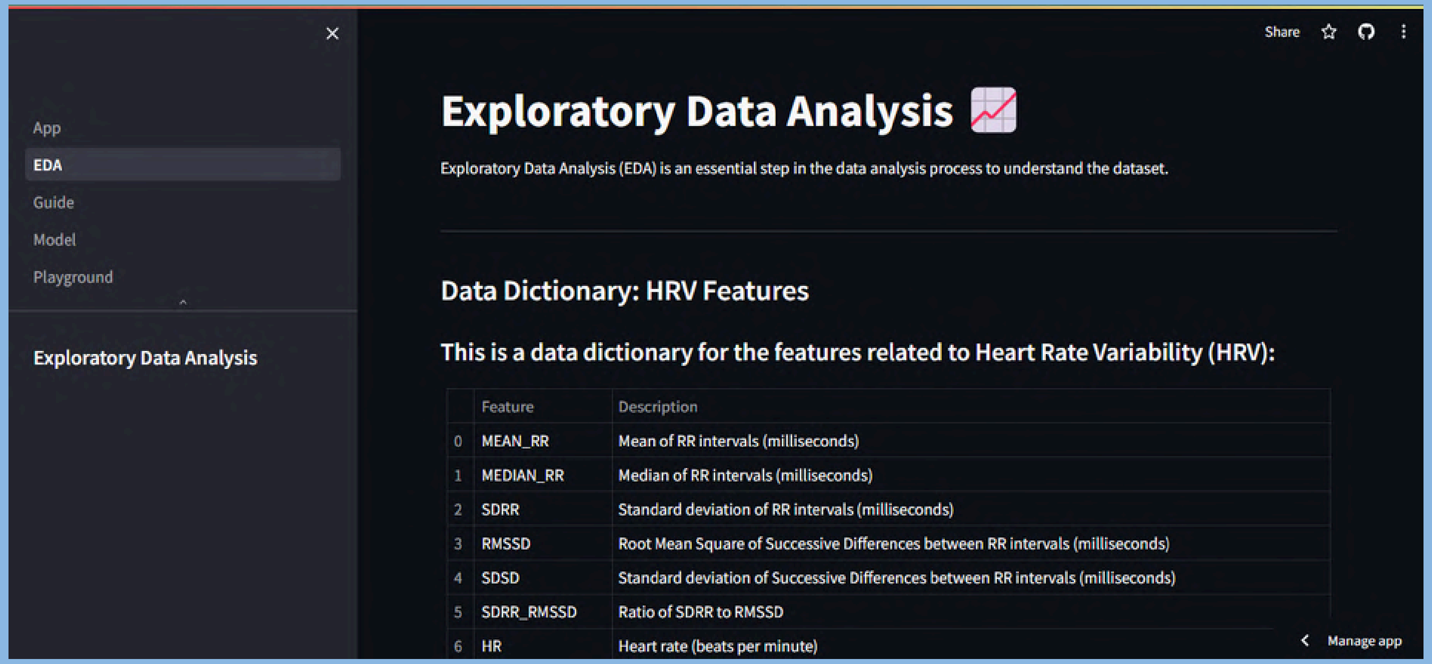
HeartWise

Pages include:
- App/Menu
- EDA
- Guide
- Model
- Playground

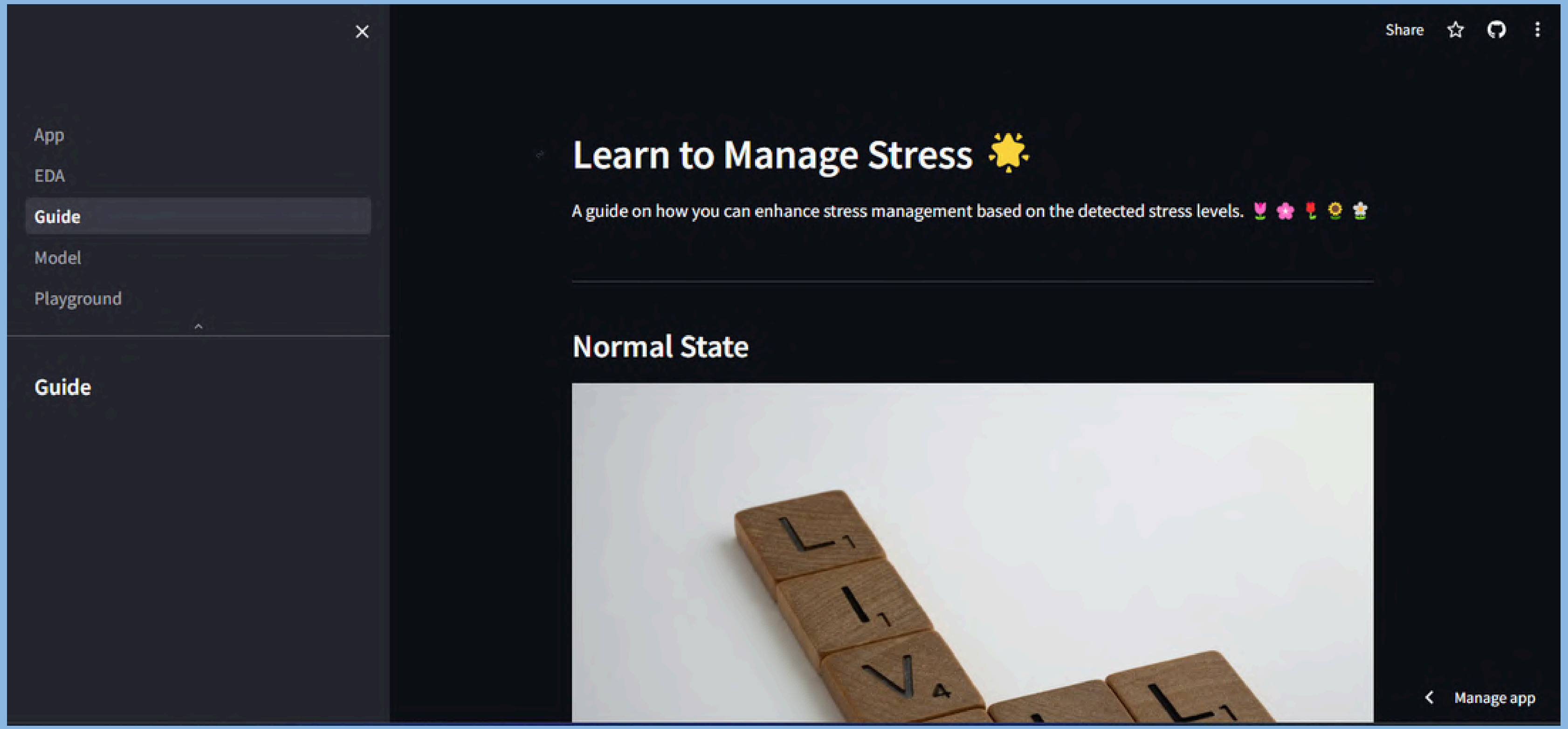
Page – App/Menu:



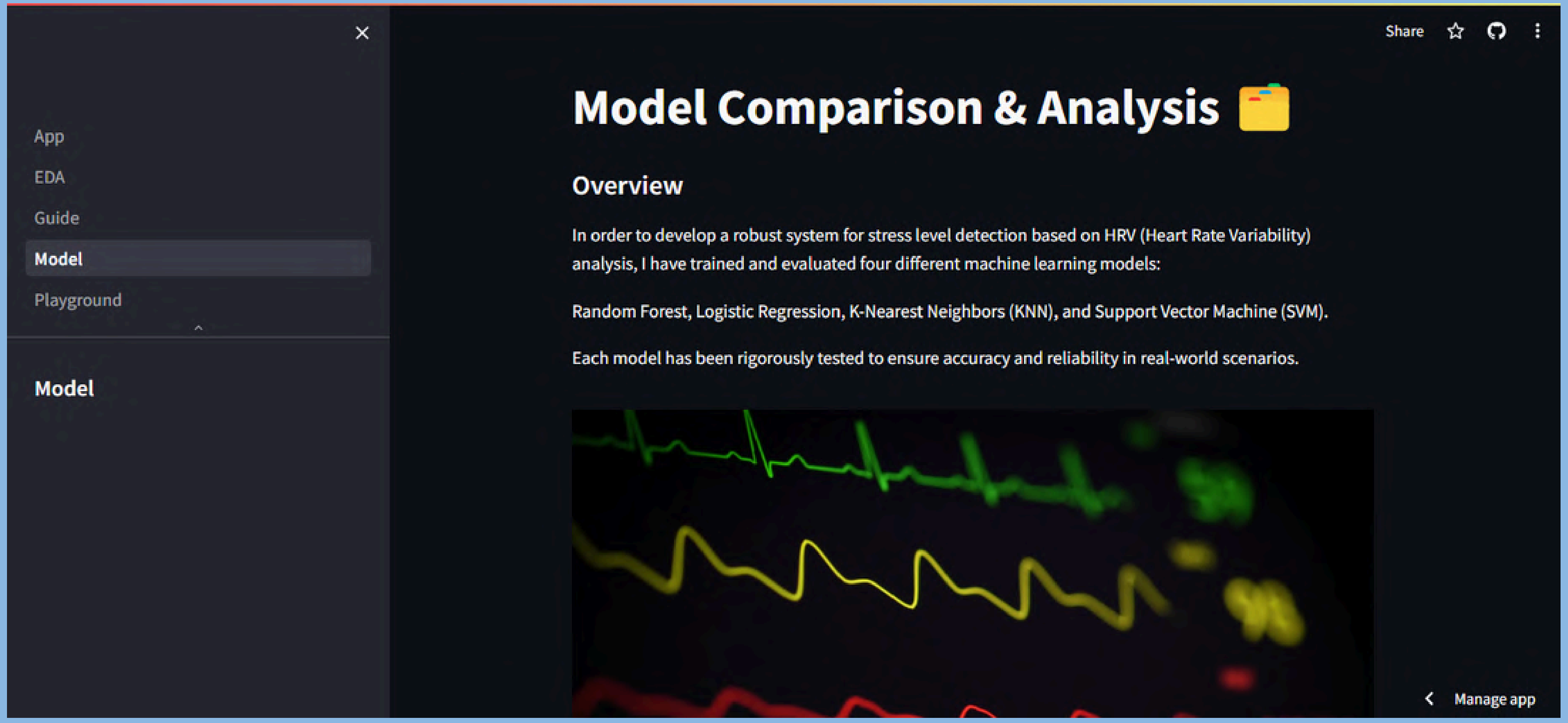
Page – EDA:



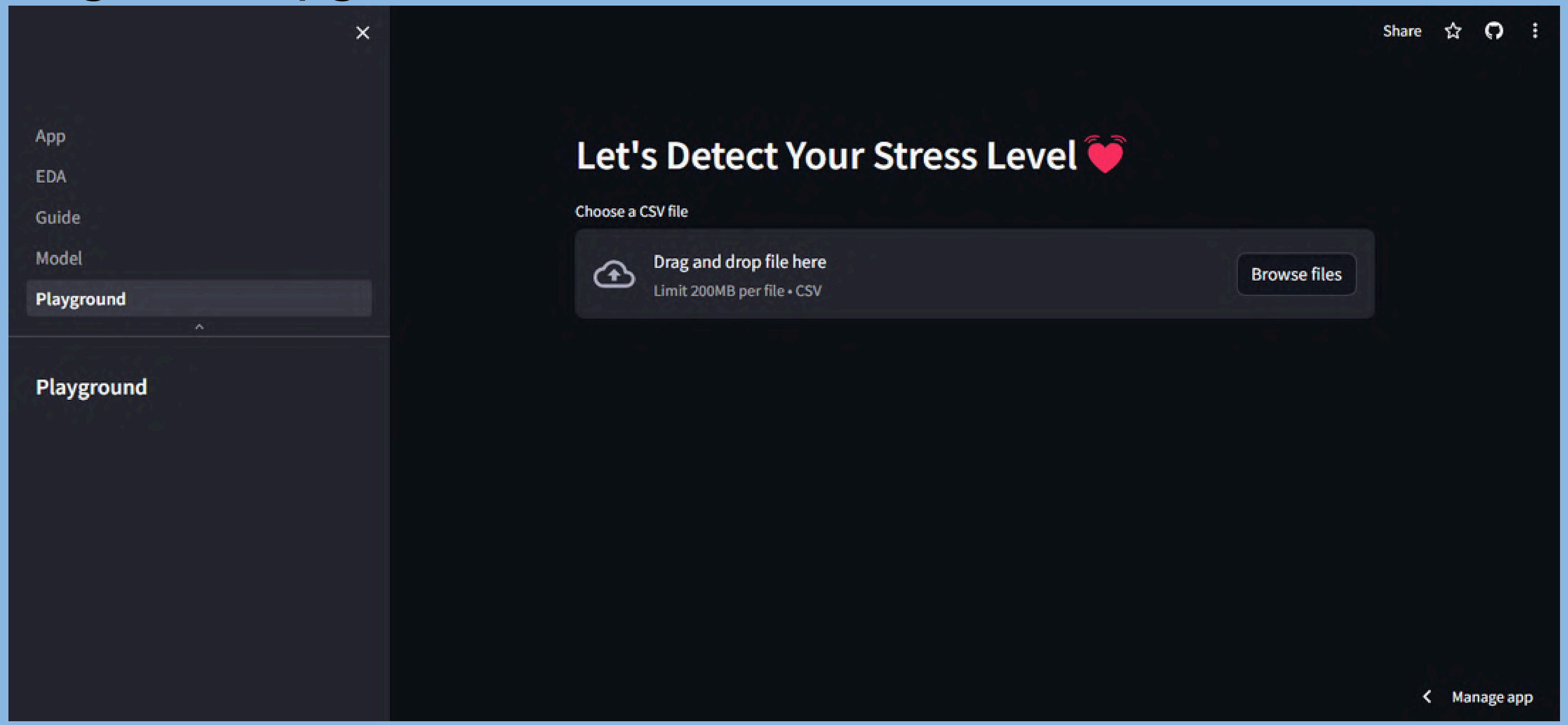
Page - Guide:



Page - Model:



Page - Playground



Tools

PROGRAMMING ENVIRONMENT & LANGUAGE

WEB APP FRAMEWORK

Streamlit

DATA PREPARATION & EXPLORATION

pandas NumPy

MODEL DEVELOPMENT & EVALUATION

scikit learn

Joblib

DATA VISUALIZATION

matplotlib

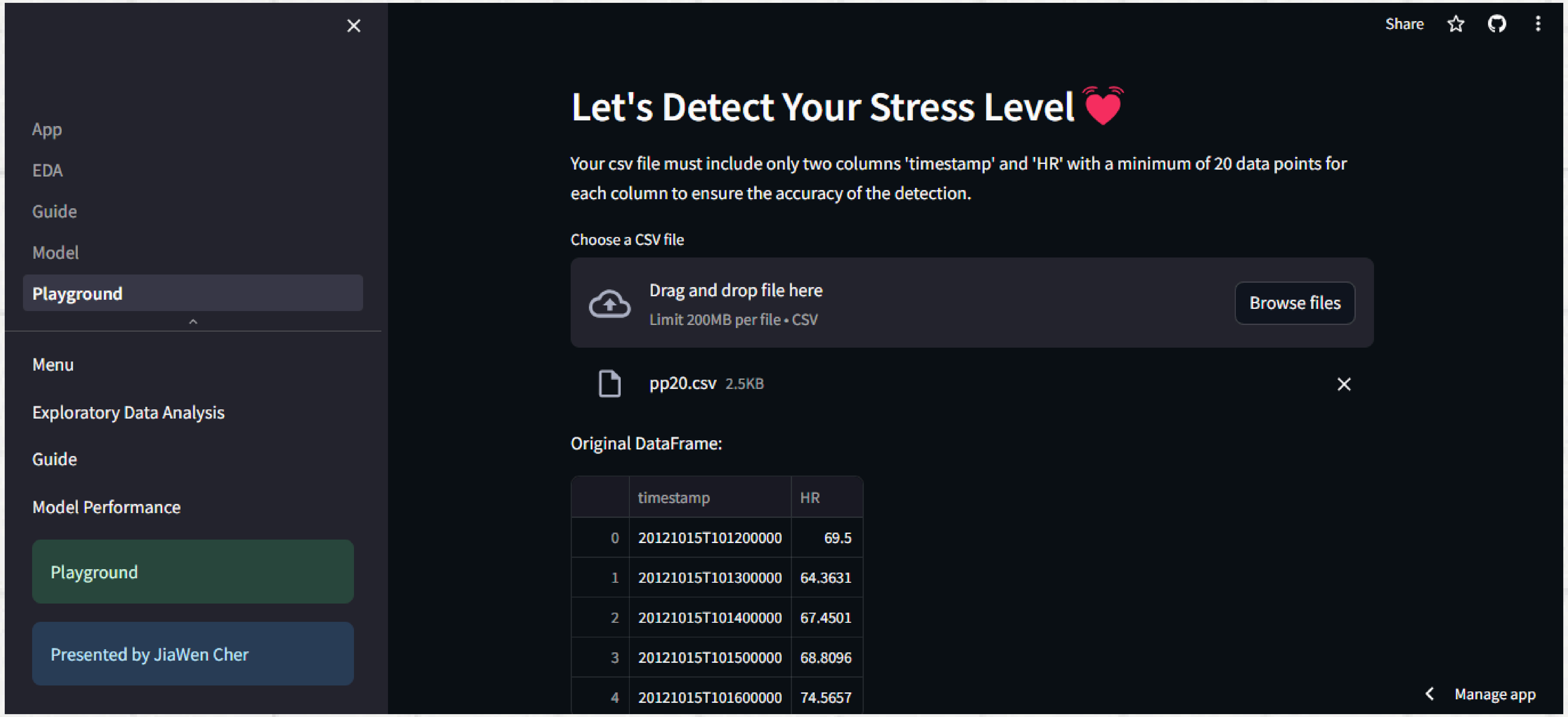
Power BI

plotly

seaborn

03
Demonstration of
HeartWise

Demonstration Video:



https://drive.google.com/file/d/1AEwhyUFsMB4xSc1nRJ9oQpkWyUz3Zelh/view?usp=sharing

User Manual:

User Manual

HeartWise

Empower your heart, master your stress.

A place where every heartbeat tells a story.

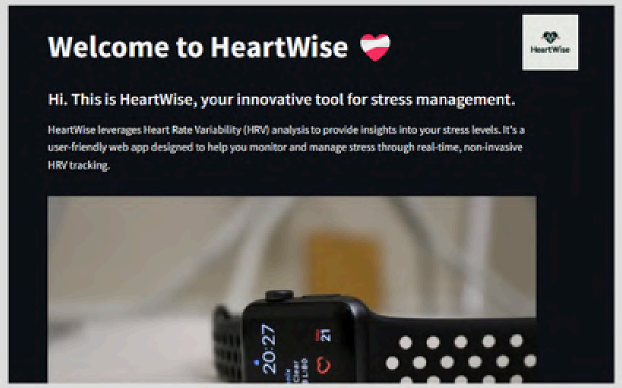
Scan to access HeartWise.

https://heartwise.streamlit.app/

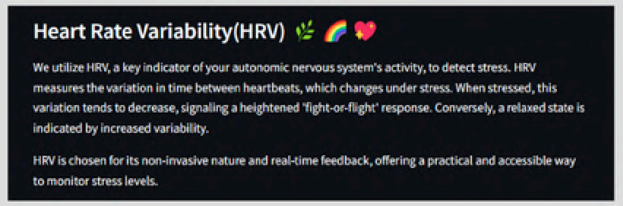
App/Menu Page:

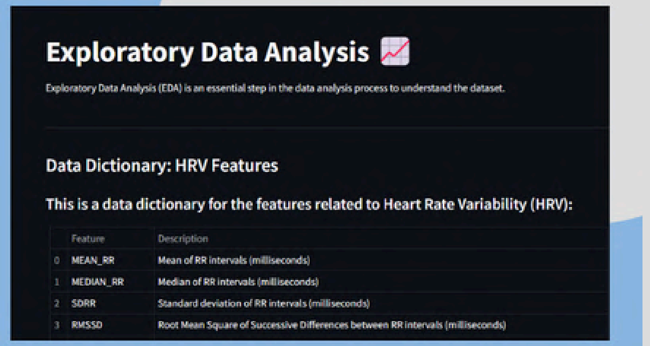
Discover HeartWise

Welcome to HeartWise!
- Discover how Heart Rate Variability (HRV) is linked to your stress levels.
- Find out what makes HeartWise your go-to app for stress management.



Quick Tips:
- Navigate through the menu to explore various features.
- Tap on any section for detailed insights about your heart health and stress.





EDA Page:

Dive into Data

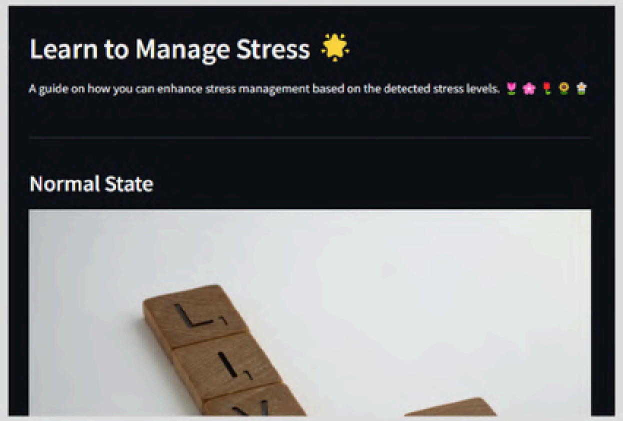
Explore Our Data Universe
- Check out the data that powers HeartWise, including HRV features and our comprehensive data dictionary.
- Experience our interactive dashboard for a hands-on look at dataset analysis.

How to Use:
- Scroll through the dataset for an overview.
- Use the interactive tools to visualize different aspects of the data.

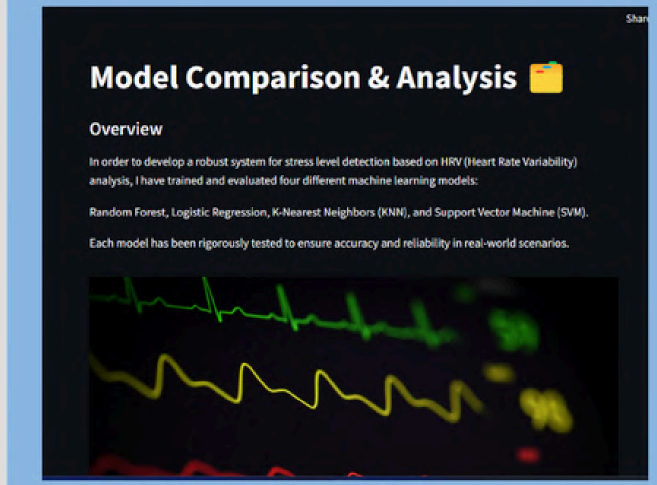
Guide Page:

Understand Your Stress Level

Your Stress Levels
- Learn about the 3 levels of stress detected by HeartWise.
- Discover practical tips and techniques to manage each stress level.



Step-by-Step:
- Identify your current stress level using our easy-to-read indicators.
- Follow our tailored advice to manage and reduce your stress effectively.



Model Page:

Insight into Our Model

Behind the Scenes: The ML Model
- Understand why Random Forest classifier is chosen as our trusted model for accurate detections.

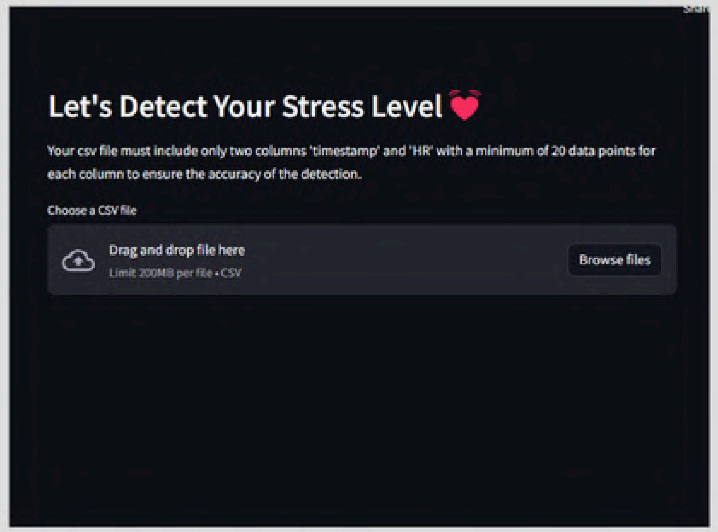
Model Insights:
- Read about the mechanics of our model in simple terms.
- Find out how our model analyzes and interprets your HRV data.

Playground Page:

Your Personal Analysis

Interactive Stress-Level Detection

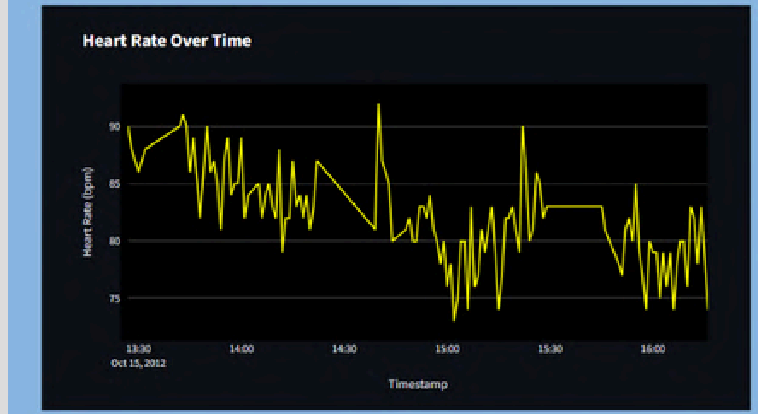
- Upload your own CSV file with 'timestamp' and 'hr' to see HeartWise in action.
- Ensure your file has a minimum of 20 data points for each attribute for precise analysis.

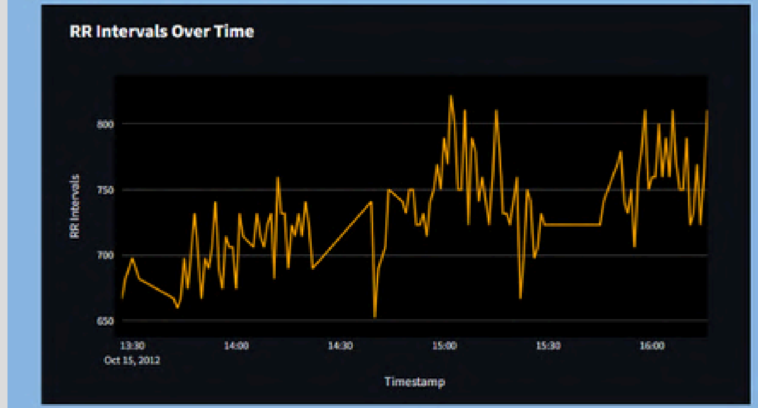


How to Play:

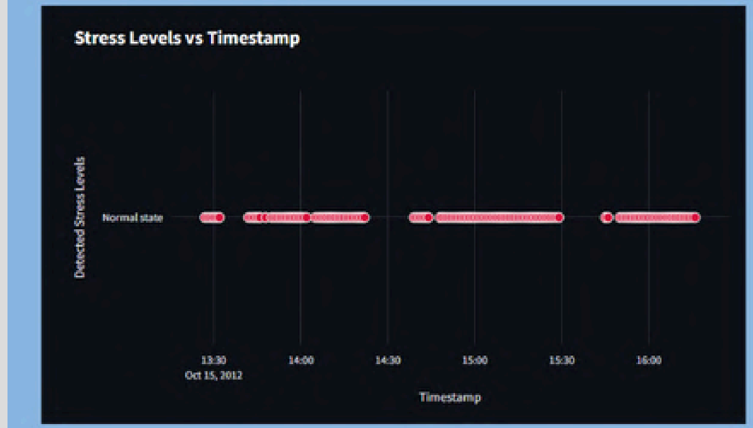
- Upload your csv file.
- Watch as HeartWise processes your data, calculating HRV features and determining your stress level.
- Interact with the plots to understand your stress level in relation to time and heart rate.

HRV Analysis:





Stress Level Detection:



Stress Level Detections:

	timestamp	HR	detected stress levels
0	2012-10-15 13:27:00	90	Normal state
1	2012-10-15 13:28:00	88	Normal state
2	2012-10-15 13:29:00	87	Normal state
3	2012-10-15 13:30:00	86	Normal state
4	2012-10-15 13:31:00	87	Normal state

All the analysis plots and results can be export as file after detection completed.

Your heart's story unfolds each day – let HeartWise be the narrator.

Transform stress management into a daily ritual with HeartWise.

Seamlessly monitor your heart rate variability (HRV) to understand your stress levels in real-time.

With our intuitive dashboard and personalized insights, tracking your heart's health becomes an effortless part of your daily routine.

Use HeartWise to detect stress patterns, learn coping strategies, and maintain a balanced, healthier lifestyle.

Thanks

HeartWise

https://heartwise.streamlit.app/

04
Conclusion

Limitations
Future Improvements

Conclusion

This project successfully **achieved the objectives** in developing a stress level detection model using HRV analysis.

- **To identify the best machine learning algorithm to detect stress levels using HRV analysis.**

Gradient Boosting - the most effective algorithm for detecting stress levels using HRV data. This model demonstrated superior performance in terms of precision, recall, and F1-score, making it the most suitable choice for this application.

- **To develop a machine learning model capable of analyzing HRV data to detect stress levels.**

A ML model that leverages HRV data to detect stress levels has been successfully developed. This model trained and validated on a dataset comprising HRV features, effectively interpret HRV data patterns associated with different stress levels.

- **To deploy a user-friendly web application of the HRV-based stress level detection system.**

"**HeartWise**", a Streamlit-based web application has been deployed. This platform allows users to easily upload their heart rate CSV files for analysis and enhance the self-awareness of stress levels

Limitations

- Lack of Domain Knowledge
- Computational Constraints
- Time Constraints

Future Improvements

- Enhance UI/UX by adding personalized feedback such as advise by domain expert and educational content regarding stress and HRV.
- Optimize in terms of privacy concerns
- Implement stress monitoring/detecting features into wearable devices, thus user can detect heart rate and stress levels at the same time.
- Expand the capabilities by integrating additional physiological markers alongside HRV

05
References

Woo Hee Jung, Jang, K.-I., & Lee, S.-H. (2019). Heart and Brain Interaction of Psychiatric Illness: A Review Focused on Heart Rate Variability, Cognitive Function, and Quantitative Electroencephalography. Clinical Psychopharmacology and Neuroscience : The Official Scientific Journal of the Korean College of Neuropsychopharmacology, 17(4), 459–474. https://doi.org/10.9758/cpn.2019.17.4.459

Heart Rate Variability - How to Analyze ECG Data - iMotions. (2019, July 19). IMotions. https://imotions.com/blog/learning/best-practice/heart-rate-variability/#:~:text=The%20single%20most%20common%20way,of%20HRV%20and%20parasympathetic%20activity

Pedro Elkind Velmovitsky, Alencar, P., Leatherdale, S. T., Cowan, D., & Plinio Pelegrini Morita. (2022). Using apple watch ECG data for heart rate variability monitoring and stress prediction: A pilot study. Frontiers in Digital Health, 4. https://doi.org/10.3389/fdgth.2022.1058826

Malaysia: stress and anxiety level by age 2022 | Statista. (2022). Statista; Statista. https://www.statista.com/statistics/1322323/malaysia-share-of-people-feeling-more-stressed-or-anxious-by-gender/

Dalmeida, K. M., & Masala, G. L. (2021). HRV Features as Viable Physiological Markers for Stress Detection Using Wearable Devices. Sensors, 21(8), 2873–2873. https://doi.org/10.3390/s21082873

Hogan Assessments. (2020, April 8). (“Managing Stress With Strategic Self-Awareness,” 2020, #)Managing Stress with Strategic Self-Awareness | Hogan Assessments. Hogan Assessments. https://www.hoganassessments.com/blog/managing-stress-with-strategic-self-awareness/

Harvard. (2017, November 22). Heart rate variability: How it might indicate well-being - Harvard Health. Harvard Health; Harvard Health. https://www.health.harvard.edu/blog/heart-rate-variability-new-way-track-well-2017112212789

S. Srakocic. (2023, August 18). Heart Rate Variability (HRV): What It Means and How to Find Yours. Healthline; Healthline Media. https://www.healthline.com/health/heart-health/heart-rate-variability-chart#hrv-guidelines
Clinic, C. (2021). Heart Rate Variability (HRV): What It Is and How You Can Track It. Cleveland Clinic. https://my.clevelandclinic.org/health/symptoms/21773-heart-rate-variability-hrv

Dalmeida, K. M., & Masala, G. L. (2021). HRV Features as Viable Physiological Markers for Stress Detection Using Wearable Devices. Sensors, 21(8), 2873–2873. https://doi.org/10.3390/s21082873

Christos Goumopoulos, & Στεργιόπουλος, Ν. Γ. (2022). Mental stress detection using a wearable device and heart rate variability monitoring. Elsevier EBooks, 261–290. https://doi.org/10.1016/b978-0-323-90585-5.00011-4

Hats, M. (2023, November 3). *The Role of Self-Awareness in Stress Management - The Woman Who Wears Many Hats - Medium*. Medium; Medium. https://medium.com/@thewomanwhowearsmanyhats/the-role-of-self-awareness-in-stress-management-a11324419d92

Fink, G. (2016). Stress: The Health Epidemic of the 21st Century | SciTech Connect. Elsevier.com. https://scitechconnect.elsevier.com/stress-health-epidemic-21st-century/

WebMD Editorial Contributors. (2021, April 20). What Is Heart Rate Variability? WebMD; WebMD. https://www.webmd.com/heart/what-is-heart-rate-variability

Thank you very much!