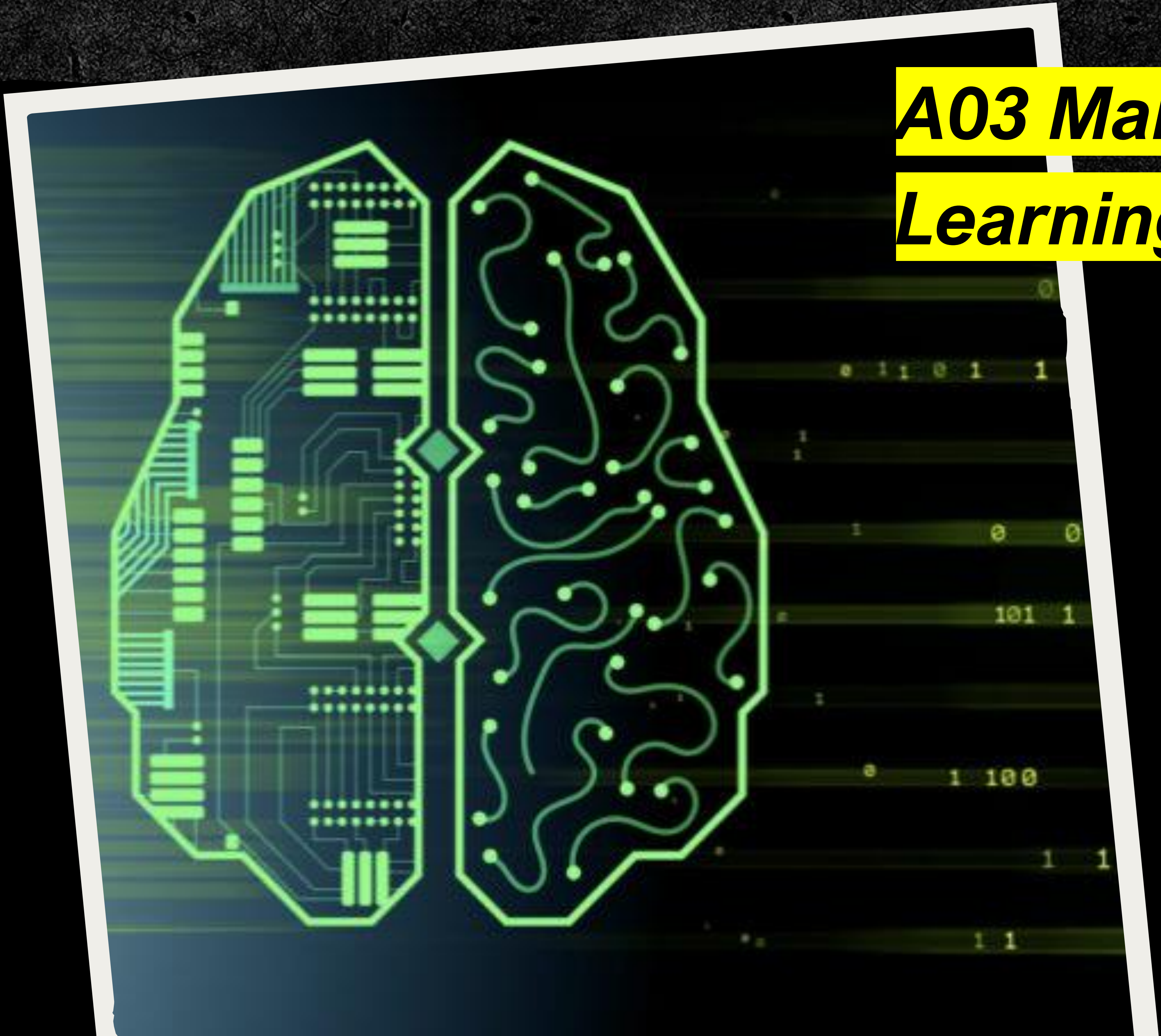


A03 Manual Machine Learning Model Design

Andrew Badzioch

Kolby Boyd

Florentin Degbo

Natalia Solorzano

Jeralynn Sparks

Prof. Patricia McManus

ITAI 1378

06 February 2024

Fitness and Health Tracker

Data Collector: Florentin Degbo

Algorithm Designer: Andrew Badzioch

Model Trainer: Natalia Solorzano and Kolby Boyd

Application Specialist: Florentin Degbo

Introduction

- Expanding array of sensors and data collectors enable powerful applications in personal health and fitness tracking.
- Monitors heart rate, calorie intake, step counts, sleep duration, and mood for holistic health insights.
- Foundation: Data collection gathers information for model training and testing.
- Data processing involves algorithms, ranging from supervised to unsupervised or reinforcement learning based on use cases.
- Trained model is fine-tuned through the training process, minimizing error and boosting rewards.
- Integration of applications ensures the model functions seamlessly, providing valuable insights and recommendations.

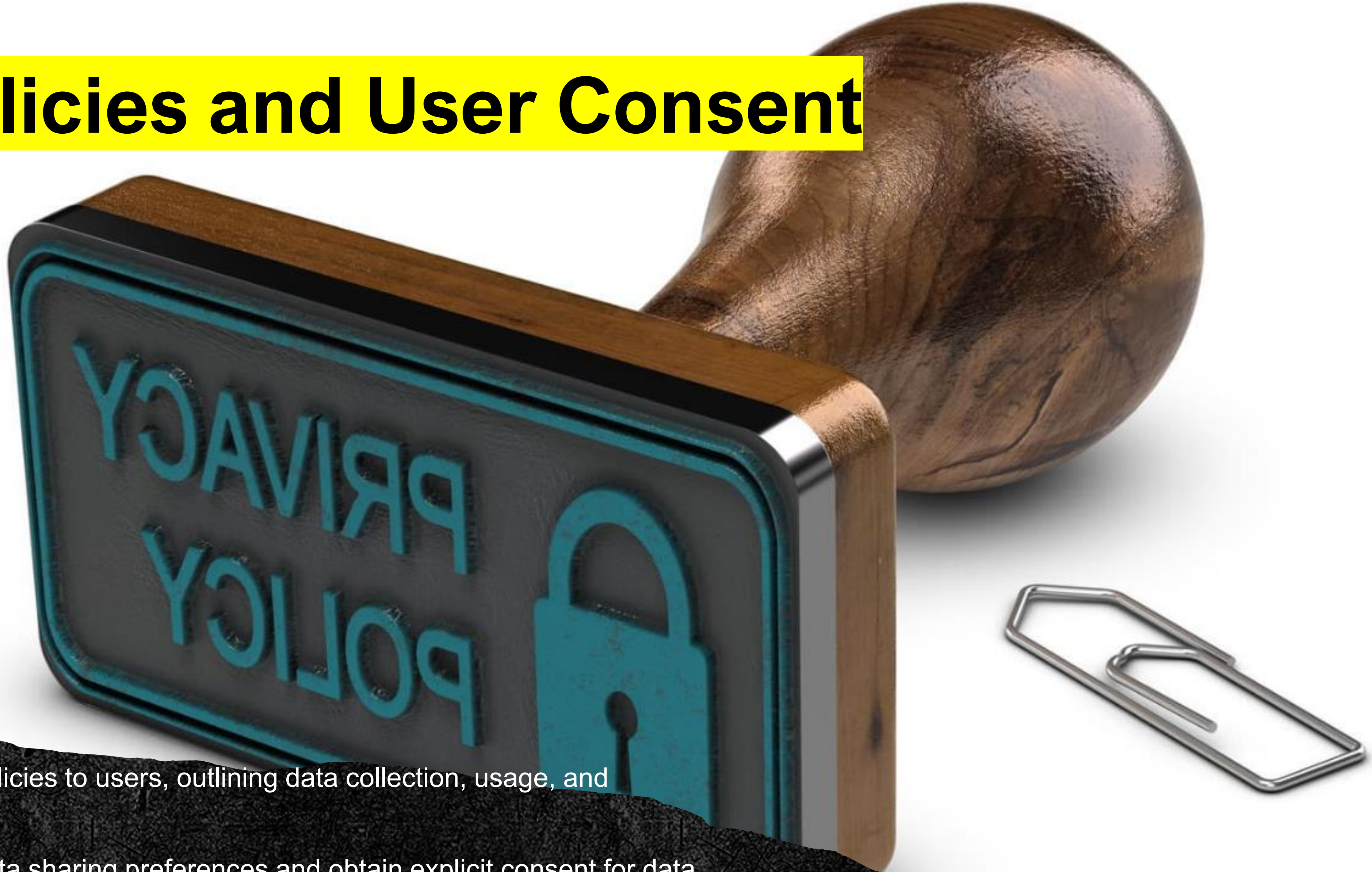
Data collection

- Data Sources
- Collection Methods
- Data Processing

Data Security Measures

- Employ robust data encryption techniques to safeguard user information.
- Utilize secure communication protocols like HTTPS for data transmission.
- Implement strict access controls to prevent unauthorized access to sensitive health data.

Privacy Policies and User Consent



- Clearly communicate privacy policies to users, outlining data collection, usage, and protection measures.
- Provide granular control over data sharing preferences and obtain explicit consent for data collection.
- Allow users to revoke consent or delete their data at any time to ensure transparency and user autonomy.

Algorithms

1. # Personalized workout plan generation from sklearn.ensemble
import RandomForestRegressor
# Training the model with historical workout data
model = RandomForestRegressor() model.fit(training_data, target_data)
# Predicting the next workout based on current progress
predicted_workout = model.predict(new_data)

2. # Predicting nutritional needs using linear regression from sklearn.linear_model
import LinearRegression
# Training the model with dietary data
model = LinearRegression() model.fit(training_diet_data, target_calories_data)
# Predicting optimal daily caloric intake
predicted_calories = model.predict(new_diet_data)

Algorithms Cont.

3. # Activity recognition using a neural network

```
import tensorflow as tf
# Building a neural network for activity classification
model = tf.keras.models.Sequential([ # Define layers for sensor data processing and classification ])
# Training the model with labeled activity data
model.fit(training_sensor_data, training_activity_labels)
```

4. # Injury risk detection using support vector machines

```
from sklearn.svm import SVC
# Training the model with biomechanical data
model = SVC() model.fit(training_biomechanical_data, target_injury_labels)
# Predicting injury risk during a workout session
predicted_injury_risk = model.predict(new_biomechanical_data)
```

Model Training

-Data Preparation: training, validation and test sets

-Machine Learning Model

-Hyperparameter fine-tuning

-Model's performance is assessed and modified if needed

-Validation Methods are implemented

-Final adjustment of threshold values or class weights

A03_OpticMinds_ITAI1378.ipynb

File Edit View Insert Runtime Tools Help Last saved at 4:59 PM

+ Code + Text

```
#Install libraries
!pip install scikit-learn
```

```
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (1.2.2)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.23.5)
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.11.4)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.3.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (3.2.0)
```

```
#Import RandomForestRegressor
from sklearn.ensemble import RandomForestRegressor
print(RandomForestRegressor)
```

```
<class 'sklearn.ensemble._forest.RandomForestRegressor'>
```

```
# Training the model with historical workout data
training_data = [[1,2,3],[4,5,6],[7,8,9]]  # Replace with actual training data
target_data = [10,20,30]  # Replace with actual target data
model = RandomForestRegressor()
model.fit(training_data, target_data)
```

```
RandomForestRegressor
RandomForestRegressor()
```

Conclusion

The integration of machine learning into person health and fitness tracking offers tremendous potential for improving individual well-being. By collecting and analyzing various types of data from the expanding array of sensors and collectors, using the proper algorithms to train the model, and the integration into real-world applications, these models can learn from the user's behavior and preferences, offering personalized feedback to support their goals. As technology continues to evolve, the potential for machine learning in personal health and fitness tracking remains open to new thoughts and ideas, promising further growth in the management of personal well-being.

References

- ❑Linkedin: harnessing-power-machine-learning-python-fitness-austin-stewart
- ❑https://scikit-learn.org