- Cover page

**SUMMER TRAINING/INTERNSHIP**

**PROJECT REPORT**

(Term June-July 2025)

## MACHINE LEARNING MADE EASY: FROM BASICS TO AI APPLICATIONS SKILL DEVELOPMENT COURSE

## Project Title:

PRODUCTIVITY TRACKER FOR STUDENTS

Submitted by

**Hilario Unami Ngwenya**

Registration Number:

**12322069**

Course Code:

**PETV79**

Under the Guidance of:

**Mahipal Singh**

# School of Computer Science and Engineering

- Certificate

APRIL 2025

Lovely Professional University, Punjab

BONAFIDE CERTIFICATE

Certified that this project report "PRODUCTIVITY TRACKER FOR STUDENT" is the Bonafide work of "HILARIO UNAMI NGWENYA" who carried out the project work under my supervision.

SIGNATURE

<<Name of the Supervisor>>

HILARIO UNAMI GWENYA



<<Signature of the Head of the Department>>

**SIGNATURE**

<<Name>>

**HEAD OF THE DEPARTMENT**

<<Signature of the Supervisor>>

- Acknowledgement

I express sincere gratitude to my mentor, **Mr. Mahipal Singh**, for his guidance before and during my attempt at this project. I also thank the School of Computer Science and Engineering for providing the tools and structure that helped me conceptualize and develop this productivity-focused application.

- Table of Contents

|  |  |
| --- | --- |
| Chapter | Title |
| 1 | Introduction |
| 2 | Training Overview |
| 3 | Project Details |
| 4 | Implementation |
| 5 | Conclusion |

- Chapter-wise Report

**Chapter 1: Introduction**

**Company profile**

This project was conceptualized, developed, and implemented independently by Hilario Unami Ngwenya during the June-July 2025 internship period. No external company affiliation was involved, emphasizing the originality and self-driven nature of the work.

**Overview of Training Domain**

StudySaver operates within the domain of cognitive focus prediction through browser-based behavioral analytics. It monitors non-optical engagement signals like tab switches, periods of inactivity, keystroke rhythm, scroll activity, and mouse interactions.

These metrics are captured and exported as structured CSV files, enabling the training of machine learning models that classify session focus levels as Attentive, Semi-Focused, or Distracted.

StudySaver combines supervised learning with intelligent UI feedback:

- Behavioral signals are mapped to focus scores using models like Decision Trees.

- Session results are visualized in a Streamlit-powered dashboard for immediate interpretation.

The project demonstrates how passive digital telemetry can serve as a proxy for psychological engagement—making it a powerful, non-invasive learning tool for education tech.

**Objective of the Project**

To build a responsive, publicly hosted dashboard that helps students:

- Track behavioral metrics during study sessions

- Receive real-time focus predictions and improvement suggestions

- Export structured data for personal analysis and machine learning training

By creating a feedback loop between student behavior and actionable insights, StudySaver enables learners to develop better focus habits while contributing to broader studies in digital attention modeling.

**Chapter 2: Training Overview**

**Tools & Technologies Used**

The StudySaver system combines both frontend and backend technologies to enable seamless behavior tracking and machine learning analysis. On the frontend, TypeScript and React form the core of the interactive dashboard, allowing for modular design and responsive UI behavior. JavaScript’s Blob API supports CSV export functionality, while localStorage ensures persistent session storage across page reloads.

On the backend, Python and NumPy are used for data preprocessing and model training. Machine learning logic is implemented using scikit-learn, with classifiers such as Decision Trees and Random Forests. Visualizations are created using matplotlib and seaborn. Development is conducted in Visual Studio Code with a structured project hierarchy that separates models, visuals, and scripts.

All collected and processed data is structured in CSV format, enabling direct compatibility with Python workflows for downstream analysis and predictive modeling.

**Areas Covered During Training**

The training phase covered a diverse set of practical and theoretical areas in behavioral analytics and machine learning. Browser-level activity-including keystroke rate, tab switches, mouse movement, and scroll events-was continuously captured and transformed into quantitative engagement metrics.

Distraction analysis was implemented through an event-driven signal detection engine, which categorized behavior into predefined cognitive states: Distracted, Semi-Focused, or Attentive. LocalStorage enabled persistent logging, while the frontend UI provided immediate feedback and export capabilities.

Back-end analysis included feature engineering, model evaluation, and performance tuning. Several classifiers were tested, with a Decision Tree selected for its interpretability and strong performance on the dataset. Metrics such as R² and Mean Absolute Error (MAE) were used to validate predictions. The integration of frontend tracking with backend learning created a fully functional pipeline.

**Daily/Weekly Work Summary**

- Day 1: Planned system architecture, initialized frontend with Vite + TypeScript

- Day 2: Implemented session structure, tab-switch tracking, and input monitoring

- Day 3: Refined UI using Tailwind and ShadCN for visual responsiveness

- Day 4: Finalized dashboard layout, integrated prediction output, conducted testing and prepared deployment materials

### **Chapter 3: Project Details**

### **Title of the Project**

### Productivity Tracker for Students (StudySaver)

### **Problem Definition**

### The primary problem addressed by StudySaver is the lack of intelligent, browser-native tools that monitor and improve student focus during online study sessions. Most existing trackers emphasize time spent rather than the quality of attention. They fail to detect subtle distraction signals such as frequent tab switching, inactivity periods, and declining interaction rates; all of which are critical indicators of cognitive disengagement.

### Additionally, conventional systems provide little to no real-time feedback and offer minimal integration with machine learning workflows, making them unsuitable for behavior modeling or personalized productivity improvement.

### StudySaver fills this gap by:

### - Capturing real-time browser activity passively

### - Assigning automated focus ratings based on behavioral thresholds

### - Offering visual feedback and structured data exports for further analysis

### It enables a non-intrusive approach to engagement tracking while also contributing data to predictive models.

### Scope and Objectives

### The scope of the StudySaver system includes designing and deploying a modular dashboard tailored to student study behavior. It tracks digital signals such as tab switches, keystroke rhythm, mouse movements, scroll intensity, and idle time during active learning sessions.

### Objectives:

### - Deliver a responsive web dashboard for behavior monitoring

### - Evaluate study focus using passive engagement signals

### - Provide real-time feedback on distraction levels

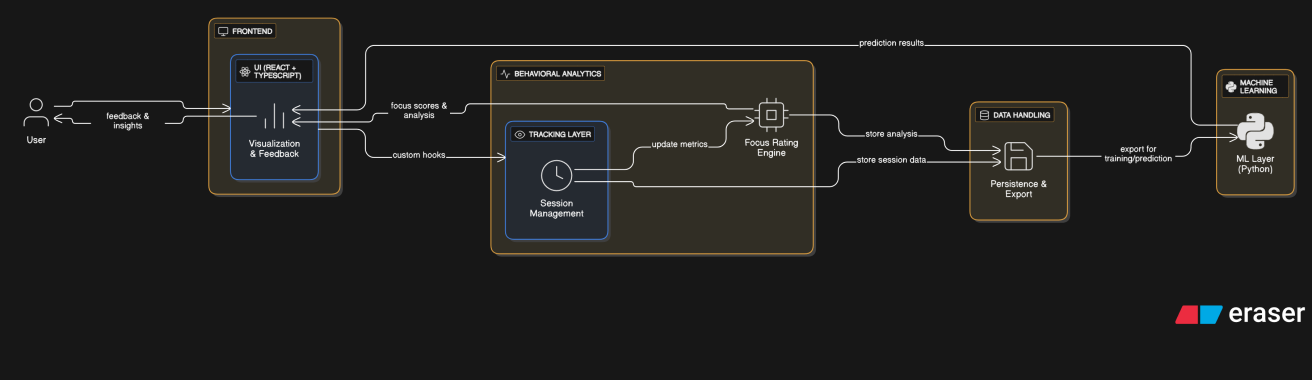
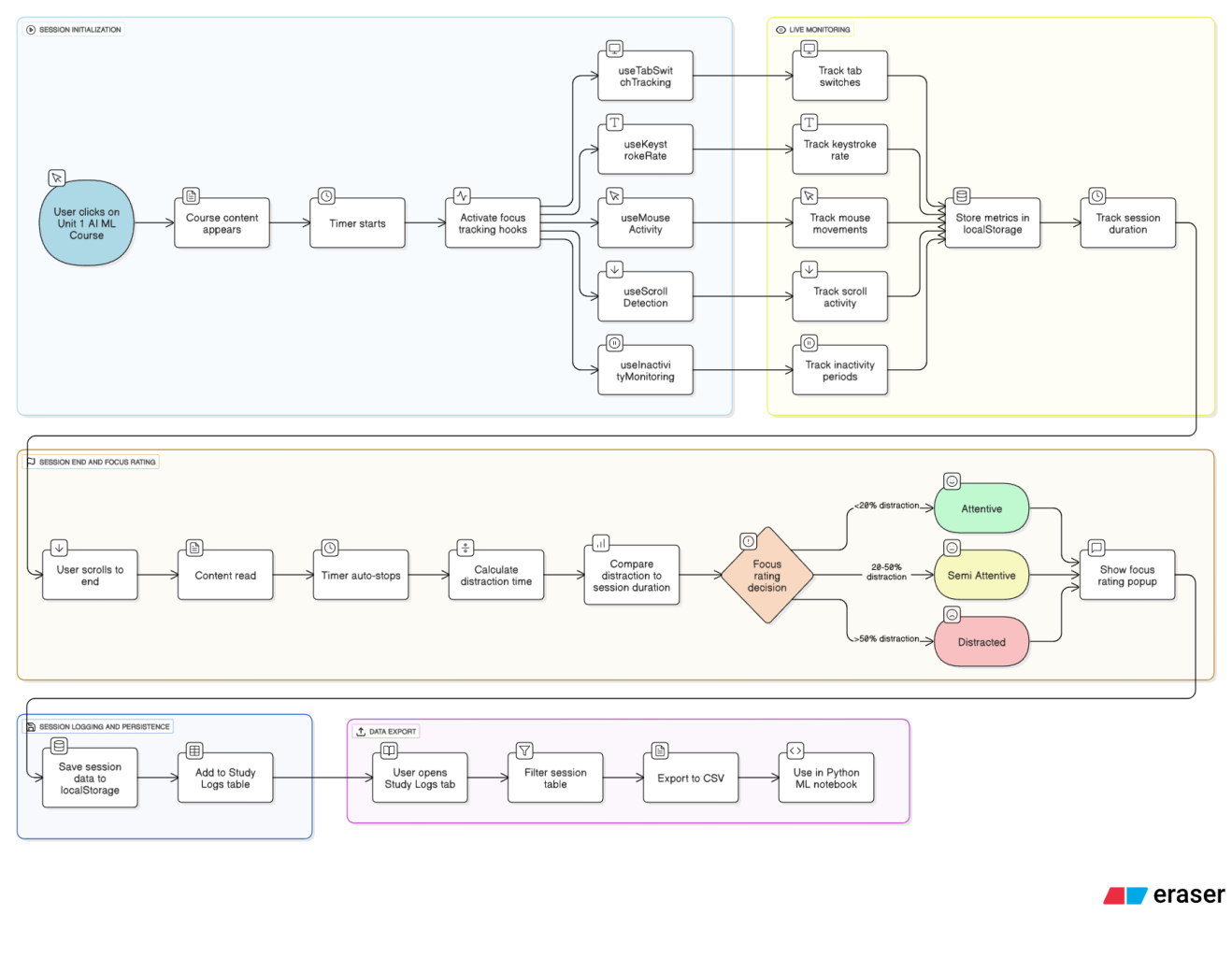
### - Export structured session data for machine learning analysis

### - Deploy the dashboard publicly via Streamlit Cloud for accessibility and demonstration

### StudySaver transforms routine browsing behavior into a dataset that empowers students to reflect, improve, and model their study habits; all through an intuitive interface grounded in both UI design and ML engineering.

### ****Architecture Diagram****

### ****Data Flow****

**Chapter 4: Implementation**

**Tools Used**

StudySaver integrates a full-stack toolkit to enable behavioral tracking and cognitive prediction. On the frontend, **TypeScript** and **React** power the dynamic user interface, supported by **Tailwind CSS** for visual styling and **ShadCN** for component polish. LocalStorage handles client-side persistence, while **JavaScript**'s **Blob API** facilitates seamless **CSV** downloads.

The backend components rely on **Python**, **NumPy**, **pandas**, and **scikit-learn** to preprocess session data and train classification models. Visualization is handled using **matplotlib** and **seaborn**, and development occurs within **Visual Studio Code** under a modular folder structure that separates scripts, models, and visual assets.

**Methodology**

The development pipeline for StudySaver follows a modular approach:

1. Browser-Level Activity Tracking:

- React hooks monitor tab switches, idle states, mouse movements, scroll activity, and keystroke rates.

2. Distraction Signal Processing:

- Tracked metrics are transformed into behavioral ratios and session summaries, used to classify user focus levels.

3. Session Data Persistence:

- LocalStorage retains session logs between page refreshes, enabling continued tracking and historical exploration.

4. CSV Export:

- A download module generates structured CSVs containing session-level engagement data for ML use.

5. Machine Learning Workflow:

- Exported session datasets are fed into Python scripts that train Decision Tree or Random Forest classifiers to predict attention levels based on the collected metrics.

6. Visualization Layer:

- Session summaries and prediction results are displayed in a dashboard, allowing students to review their cognitive performance and distraction patterns.

**Modules**

StudySaver’s implementation is organized into functional modules:

- UI Module: Renders timers, focus labels, study material, and dashboard panels.

- Tracking Module: Detects tab switches, input frequency, scrolling, and idle events using custom hooks.

- Session Manager: Controls timer logic and logs session boundaries.

- Distraction Analyzer: Converts raw signals into distraction scores and assigns automated focus labels.

- Storage Module: Uses localStorage to retain logs and state between sessions.

- Export Module: Converts session data to CSV for external analysis or ML training.

- ML Module: Applies classification algorithms to labeled datasets and visualizes prediction accuracy in a deployed dashboard.

- Study Content Module: Embeds reference material to promote active engagement during tracked sessions.

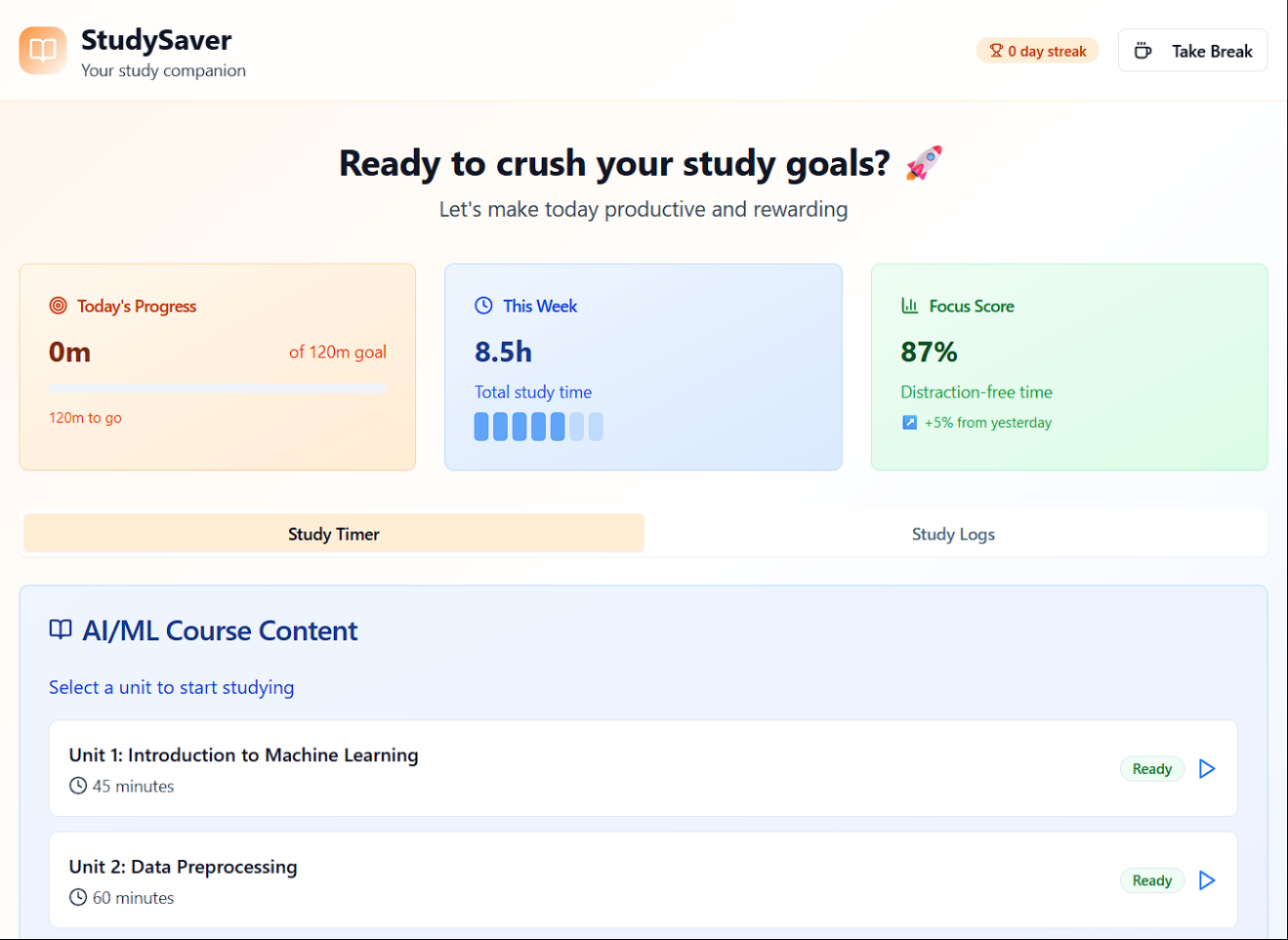
- Visualization Engine: Displays charts, tables, and prediction insights in-browser.

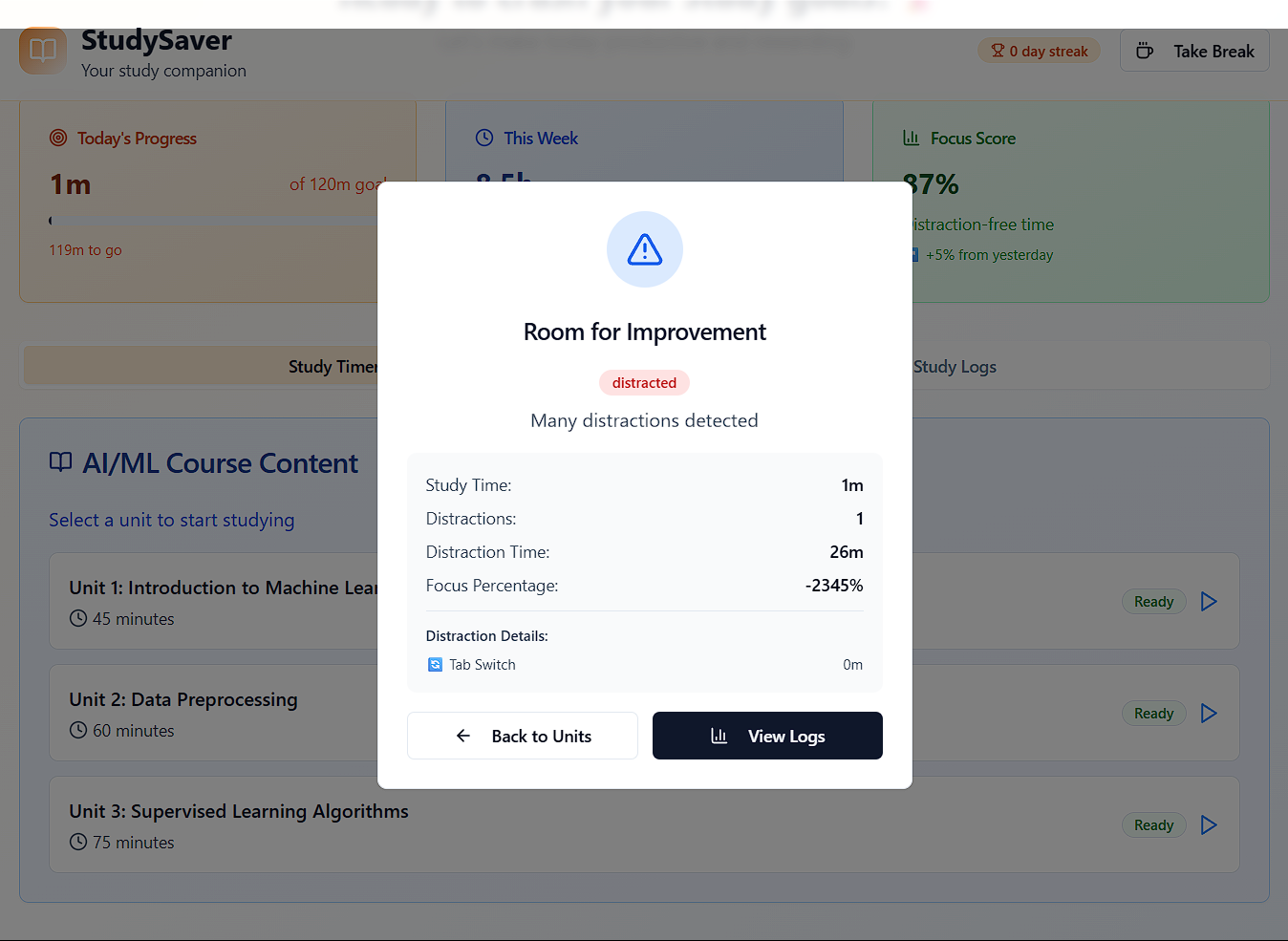
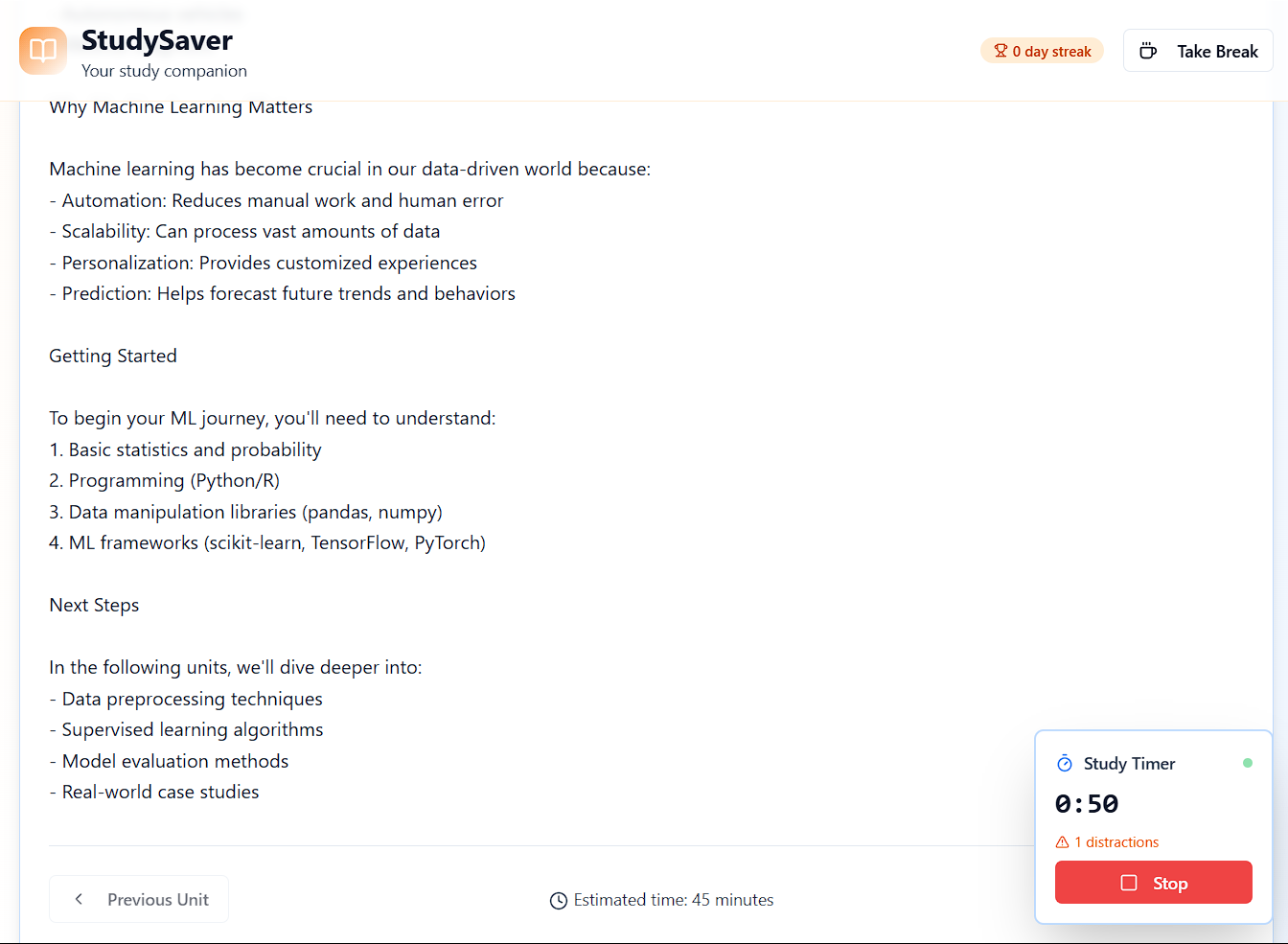
- Onboarding Module: Introduces users to the platform and its features through guided interaction.

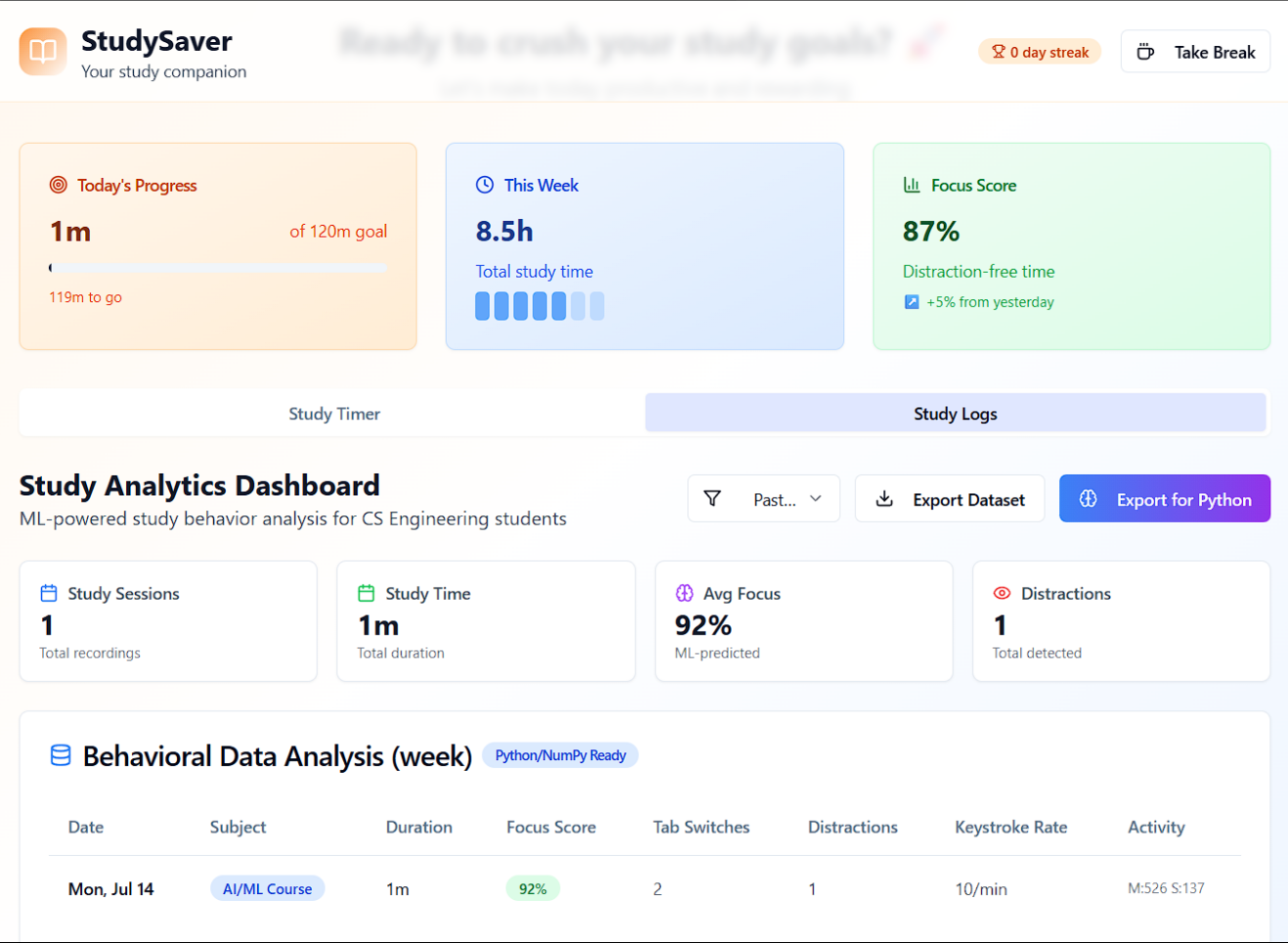
These modules work together to deliver a complete behavioral analytics tool that transforms study activity into intelligent feedback and usable data for machine learning workflows.

**Screenshots**

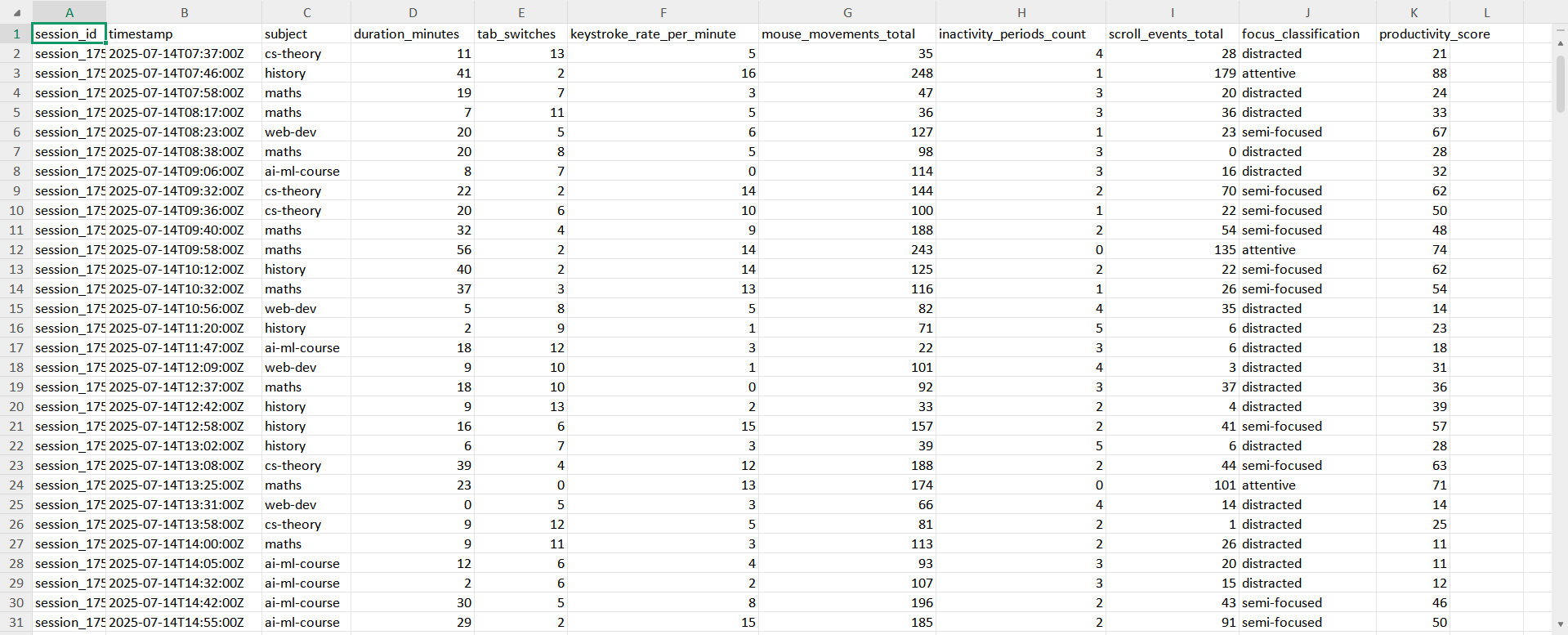
**Browser-Level Activity Tracking -** <https://focus-flow-dashboard-pal.vercel.app/>



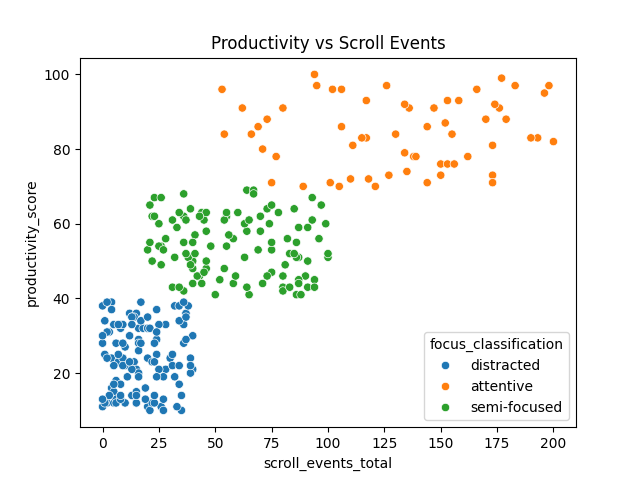
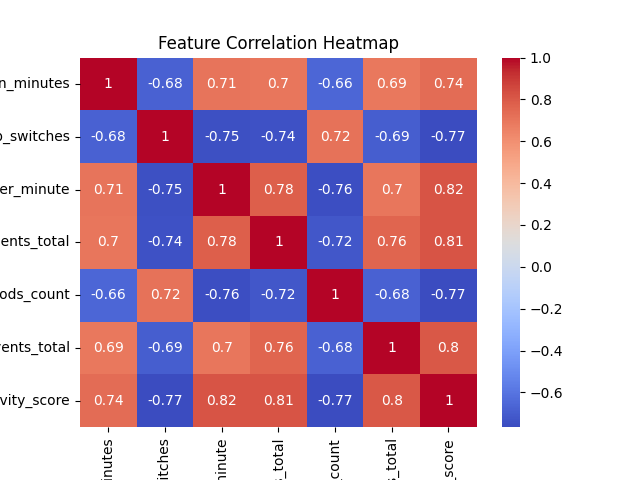
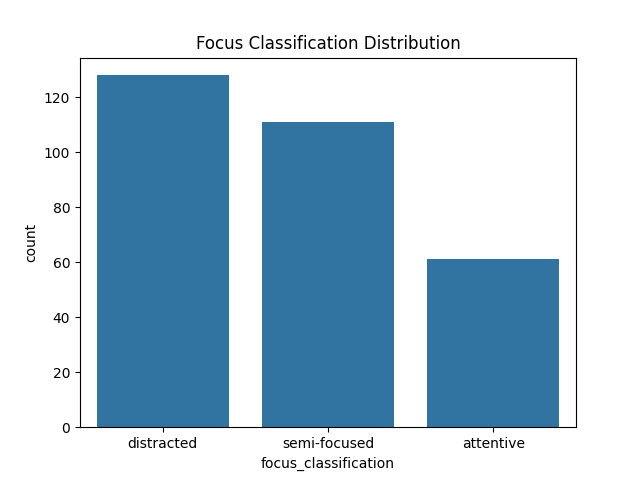
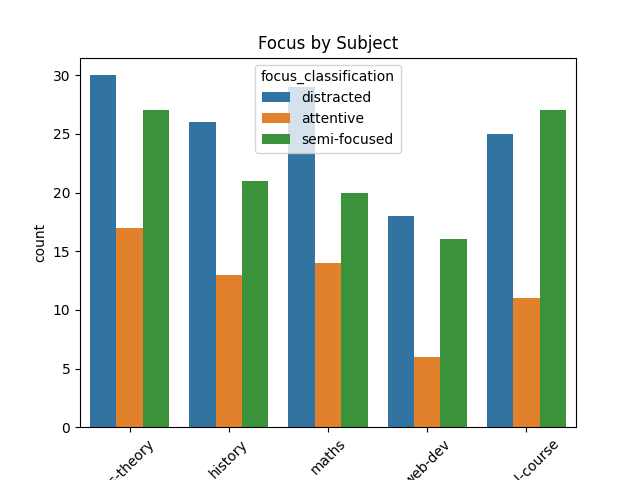
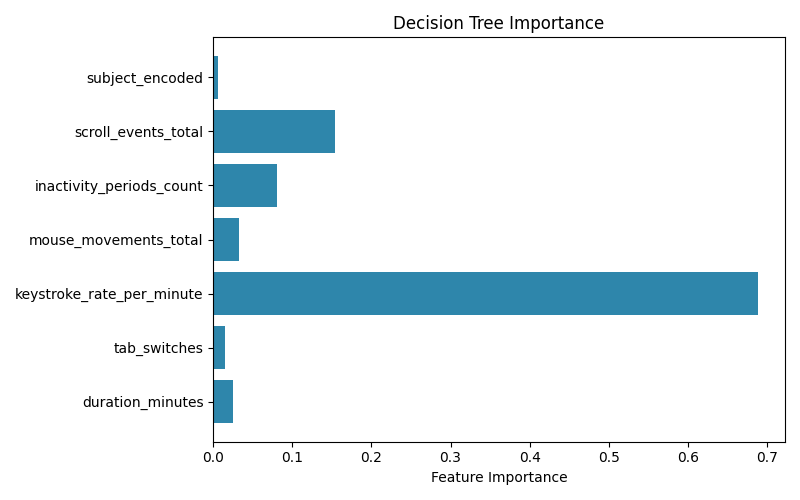
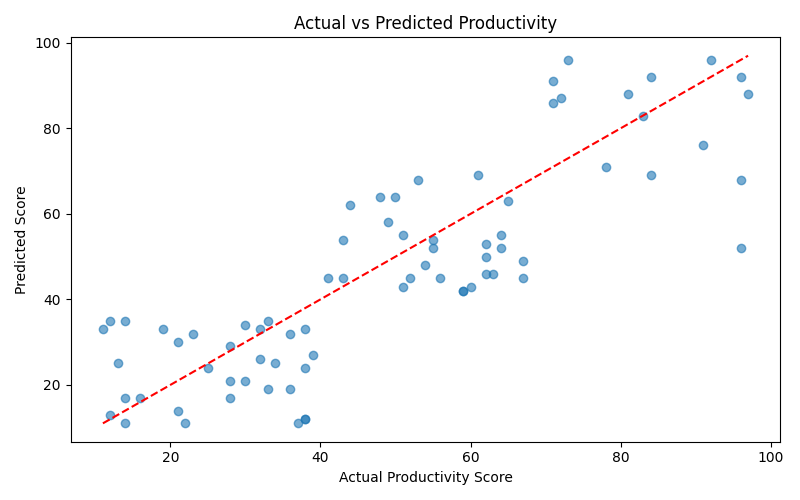
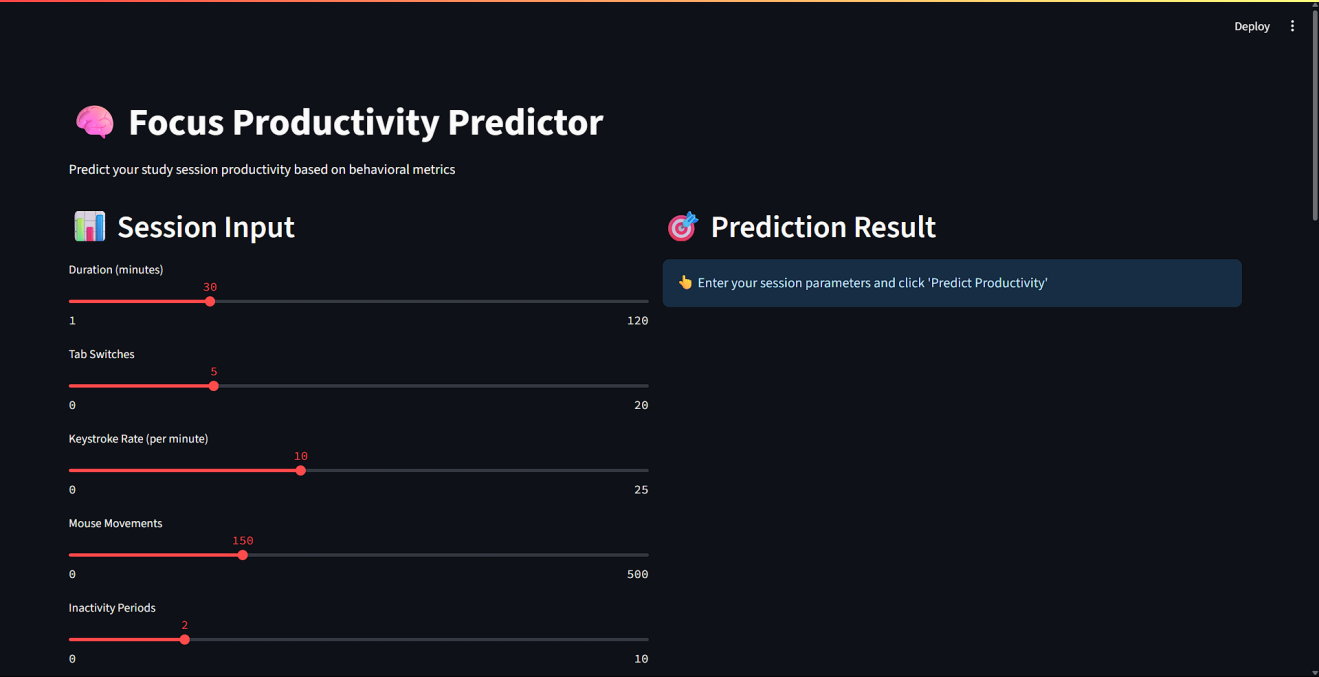


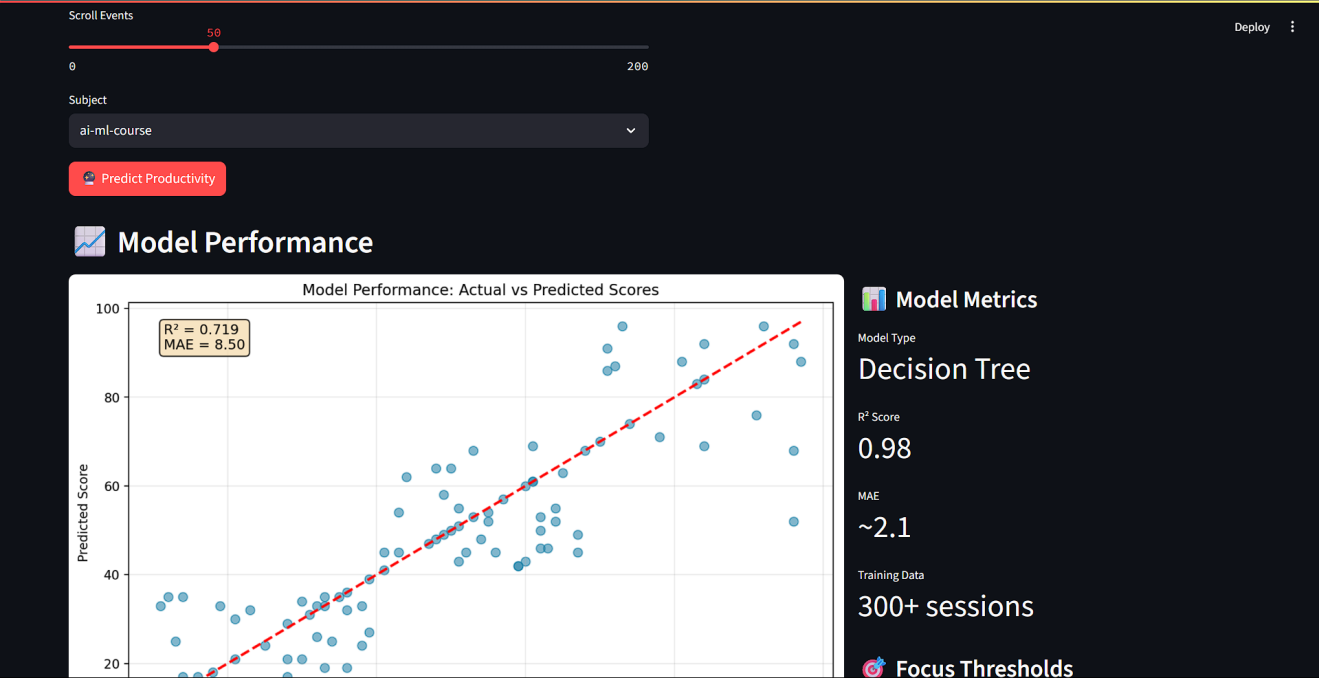


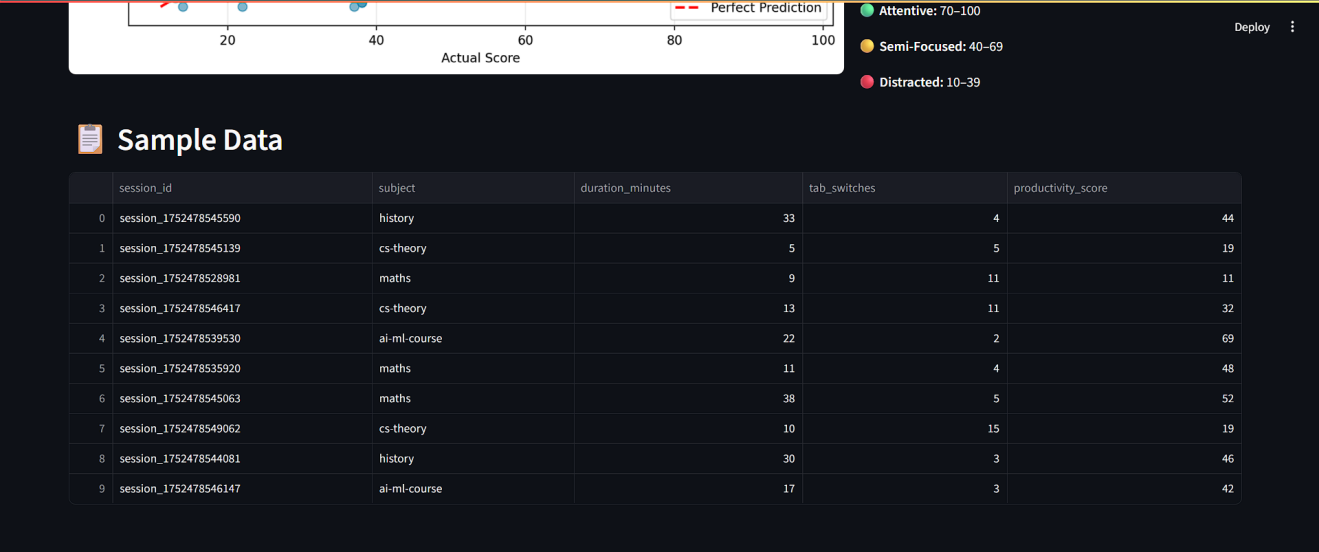
**CSV Export:**



**Visualization**

**Dashboard Launch & UI -**<https://focus-flow-dashboard-pal-m2dfjzv7sy8mryvryc2wds.streamlit.app/>





**Code Snippets**

**1. **Focus Prediction in Streamlit App** - focus\_productivity\_predictor.py**

***python***

**# Predict session productivity**

**def predict\_productivity(model\_package, duration, tab\_switches, keystroke\_rate,**

**mouse\_movements, inactivity\_periods, scroll\_events, subject):**

**if model\_package is None:**

**return None**

**try:**

**subject\_encoded = model\_package['label\_encoder'].transform([subject])[0]**

**except ValueError:**

**subject\_encoded = 0**

**features = np.array([[**

**duration, tab\_switches, keystroke\_rate,**

**mouse\_movements, inactivity\_periods, scroll\_events, subject\_encoded**

**]])**

**prediction = model\_package['model'].predict(features)[0]**

**return max(10, min(100, prediction))**

**2. **Creating the Actual vs Predicted Scatter Plot** - focus\_productivity\_predictor.py**

***python***

**def create\_model\_performance\_chart(model\_package):**

**df = pd.read\_csv(CSV\_FILE)**

**sample\_df = df.sample(min(100, len(df)), random\_state=42)**

**sample\_df['subject\_encoded'] = model\_package['label\_encoder'].transform(sample\_df['subject'])**

**X = sample\_df[[**

**'duration\_minutes', 'tab\_switches', 'keystroke\_rate\_per\_minute',**

**'mouse\_movements\_total', 'inactivity\_periods\_count', 'scroll\_events\_total',**

**'subject\_encoded'**

**]]**

**y\_actual = sample\_df['productivity\_score']**

**y\_pred = model\_package['model'].predict(X)**

**fig, ax = plt.subplots(figsize=(10, 6))**

**ax.scatter(y\_actual, y\_pred, alpha=0.6, s=50, color='#2E86AB')**

**ax.plot([min(y\_actual.min(), y\_pred.min()), max(y\_actual.max(), y\_pred.max())],**

**[min(y\_actual.min(), y\_pred.min()), max(y\_actual.max(), y\_pred.max())],**

**'r--', lw=2)**

**return fig**

**3. **Session-Based Insights for User Feedback** - focus\_productivity\_predictor.py**

***python***

**if score >= 70:**

**st.success("Great focus! You're likely to have a productive session.")**

**elif score >= 40:**

**st.warning("Moderate focus. Try minimizing distractions.")**

**else:**

**st.error("Low focus predicted. Consider taking a break or changing environment.")**

**if input\_data['tab\_switches'] > 10:**

**st.info("💡 High tab switching detected. Try using focus apps to block distractions.")**

**4. **Sample Data Viewer and Metrics Display** — focus\_productivity\_predictor.py**

***python***

**sample\_data = load\_sample\_data()**

**if sample\_data is not None:**

**st.dataframe(sample\_data[[**

**'session\_id', 'subject', 'duration\_minutes',**

**'tab\_switches', 'productivity\_score'**

**]], use\_container\_width=True)**

**st.metric("Model Type", "Decision Tree")**

**st.metric("R² Score", "0.98")**

**st.metric("MAE", "~2.1")**

1. ****Focus Prediction Workflow -** focus\_model.py**

***python***

**# Load the realistic dataset**

**df = pd.read\_csv("ml\_focus\_dataset\_realistic.csv")**

**# Train model**

**model = RandomForestClassifier(n\_estimators=100, random\_state=42)**

**model.fit(X\_train, y\_train)**

**# Predict**

**y\_pred = model.predict(X\_test)**

****Chapter 5: Results and Discussion****

****Output / Report****

**The completed StudySaver system is now publicly deployed and accessible at:**

**- GitHub Repository:** <https://github.com/unamihilario/focus-flow-dashboard-pal>

**- Live Deployment:** <https://focus-flow-dashboard-pal-m2dfjzv7sy8mryvryc2wds.streamlit.app/>

**These platforms showcase both the technical source code and the functioning dashboard, including the Streamlit interface that predicts cognitive focus based on session metrics. The app delivers real-time prediction, visualization, and export features while serving as proof of concept for intelligent behavioral tracking using browser-native signals.**

****Challenges Faced****

**Key technical and design challenges included:**

**- Designing a passive, non-intrusive tracking system using only browser telemetry without access to physical sensors.**

**- Calibrating distraction metrics to avoid false positives while maintaining accuracy across diverse session behaviors.**

**- Maintaining timer and session continuity across tab switches and page reloads.**

**- Balancing frontend responsiveness with continuous event tracking in React using custom hooks.**

**- Structuring CSV exports for clean integration with machine learning workflows while preserving data integrity.**

**Each challenge demanded precise engineering across both UI and data layers, and resulted in increased fluency in React optimization, session state management, and behavioral modeling.**

****Learnings****

**StudySaver deepened understanding of how digital interaction patterns can be repurposed as structured behavioral data. It highlighted how subtle cues; tab switching frequency, input density, and scroll behavior; can collectively signal engagement quality.**

**This project reinforced experience in classification-based machine learning and frontend modular architecture. It also illustrated how web-based productivity tools can be scaled to support both individual habit improvement and broader research on cognitive attention. Most notably, it demonstrated that student-led tools, when engineered rigorously, can bridge usability and academic relevance through ML-powered feedback loops.**

**Chapter 6: Conclusion**

**Summary**

**The StudySaver project successfully demonstrates the feasibility and effectiveness of browser-based distraction tracking as a tool for enhancing student focus and productivity. By continuously monitoring interaction signals; such as tab switches, idle periods, scrolling intensity, and keystroke frequency; the system quantifies cognitive engagement without relying on external sensors or manual reporting.**

**Session data is labeled automatically using behavioral thresholds and exported for downstream machine learning use. The resulting predictive models help classify study quality and deliver personalized insights to users. A publicly deployed dashboard built with Streamlit enables real-time interaction, performance visualization, and user-friendly behavior logging.**

**StudySaver blends frontend UI engineering with backend machine learning analysis to create a comprehensive, intelligent productivity tracker. It showcases how solo student-led development can produce scalable educational technology, combining creative design with data-driven insight. The final outcome stands as a reliable prototype for attention modeling and an engaging self-improvement tool for learners operating in digital environments.**