ML-BASED PERSONALIZED USER EXPERIENCE USING USABILITY HEURISTICS AND HEATMAP

Bachelor Thesis Project | 2024

Netaji Subhas University of Technology

PRESENTED BY

Rohan Mittal

Sujal Nimbalkar

Navneet Kumar

Agenda

1. Problem Statement
2. The Challenges
3. Objective
4. Timeline
5. Heuristic Evaluation
6. ML Model
7. Why HITL is better?
8. Result
9. Conclusion
10. Future Scope

Problem Statement

In modern web and mobile applications, delivering a personalized user experience (UX) is key to enhancing user satisfaction and engagement. However, most UX designs often rely on predefined layouts and generic interactions that do not consider the individual preferences, behaviors, and contexts of users. To address this, integrating machine learning (ML) techniques with usability heuristics and real-time interaction data (such as heatmaps) can help create a more adaptive and personalized UX.

Back to Agenda

The Challenges

01 Collects and Analyzes User Interaction Data

02 Implements Usability Heuristics

03 Maintains Usability and Performance

04 Evaluates Effectiveness

Back to Agenda

OBJECTIVE

- Capture and Interpret User Behavior
- Apply Usability Principles
- Deliver Personalized Experiences
- Balance Adaptability with Stability
- Evaluate the System's Impact

Back to Agenda

Timeline

IDEA INCEPTION → HEATMAPPER TOOL → USABILITY HEURISTICS (10) → ML MODEL → PYTHON SCRIPTS FOR DATASET

IDEA INCEPTION

Brainstorming and finalizing the project idea. Define objectives, scope, and deliverables.

HEATMAPPER TOOL

- Research and select/create a heatmapper tool to analyze user interactions.
- Collect heatmap data using this tool (e.g., tracking mouse movement, clicks, or scroll behavior).
- Process heatmap data for further use.

USABILITY HEURISTICS (10)

- Apply the 10 usability heuristics (e.g., visibility of system status, user control) to the heatmap data.
- Label the data based on heuristics as 'YES' (optimal design) or 'NO' (suboptimal design).

ML MODEL

- Train a machine learning model on the labeled dataset.
- Use techniques like hyperparameter tuning and validation for model optimization.
- Evaluate model performance using accuracy, confusion matrix, F1-score, etc.

PYTHON SCRIPTS FOR DATASET

- Develop Python scripts for handling datasets.
- Automate data preprocessing, prediction, and recommendation generation.

Methodology

DATA COLLECTION →

Track user interactions such as clicks, mouse movements, and scrolls using custom-built heatmapping tools.

Heatmap Analysis →

Input heatmap data into a machine learning model
Analyze user behavior patterns from heatmaps

MACHINE LEARNING MODEL TRAINING →

Train the CNN+RF model on the collected interaction data to predict UI adjustments
Continuously refine the model with more data

User Interface (UI) Adaptation →

Modify UI elements
Ensure changes adhere to usability heuristics such as consistency, visibility of system status, and user control.

Enable Heatmap Saving →

Check that all UI modifications maintain usability standards
Avoid overwhelming users with drastic changes

Evaluation

Measure the effectiveness of UI adaptations by tracking user satisfaction, task completion rates, and engagement metrics.
Iterate on the design based on feedback and evaluation data.

Back to Agenda

Introduction to Heuristic Evaluation

- Definition: Heuristic evaluation is a usability inspection method where a group of evaluators examines the interface and judges its compliance with recognized usability principles (the "heuristics").

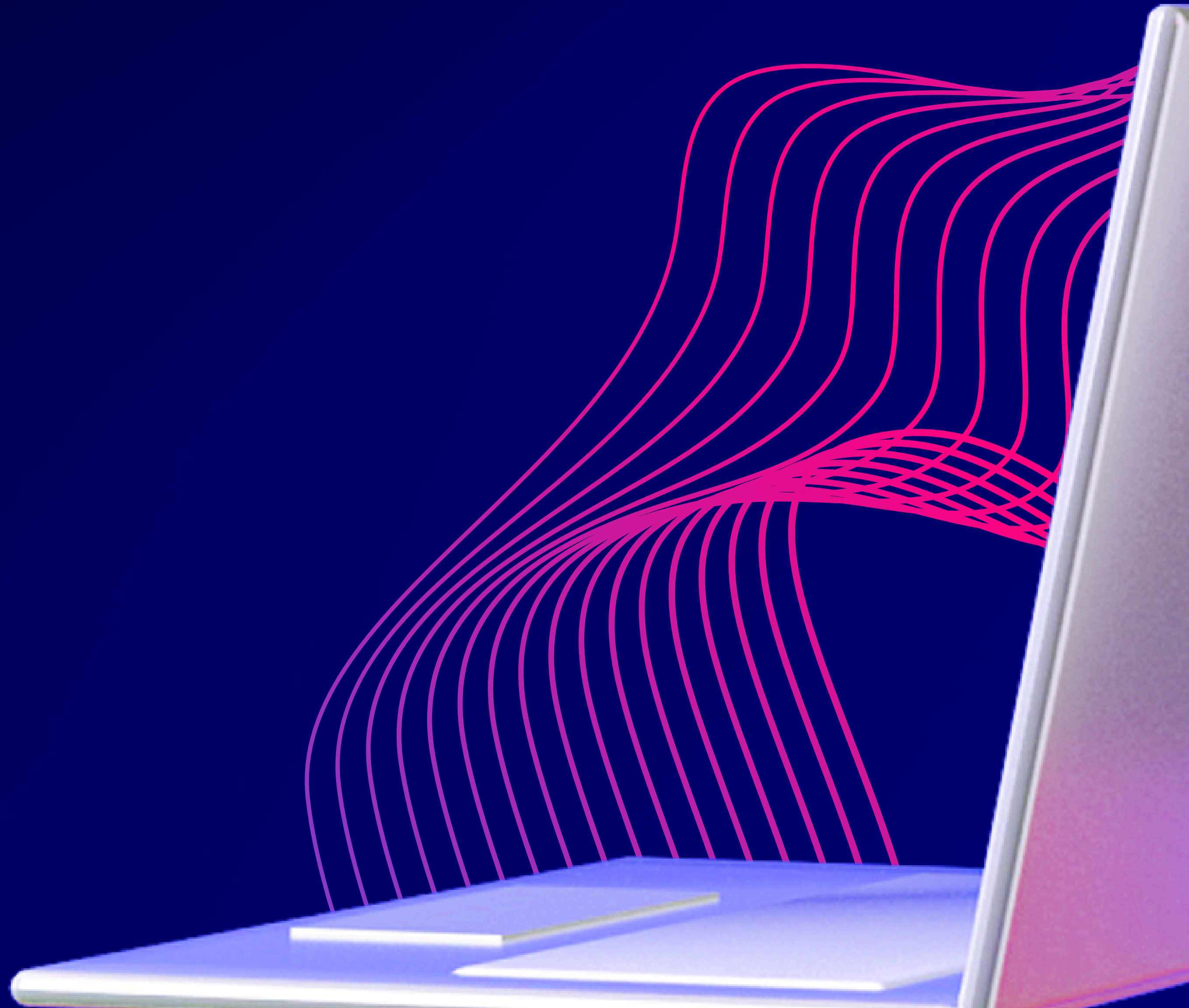
- Purpose: The primary goal is to identify usability problems in a user interface design so they can be addressed as part of an iterative design process.

Back to Agenda

Back to Agenda

10 Usability Heuristics

- Visibility of System Status
- Match Between System and the Real World
- User Control and Freedom
- Consistency and Standards
- Error Prevention
- Recognition Rather Than Recall
- Flexibility and Efficiency of Use
- Aesthetic and Minimalist Design
- Help Users Recognize, Diagnose, and Recover from Errors
- Help and Documentation



Back to Agenda

Division of Heuristics based on type of User

NOVICE USERS

1. Visibility of System Status
2. Match Between System and the Real World
3. User Control and Freedom
4. Consistency and Standards
5. Error Prevention
6. Recognition Rather Than Recall
7. Flexibility and Efficiency of Use
8. Aesthetic and Minimalist Design
9. Help Users Recognize, Diagnose, and Recover from Errors
10. Help and Documentation

INTERMITTENT USERS

1. Visibility of System Status
2. Match Between System and the Real World
3. User Control and Freedom
4. Consistency and Standards
5. Error Prevention
6. Recognition Rather Than Recall
7. Flexibility and Efficiency of Use
8. Aesthetic and Minimalist Design

EXPERT USERS

1. Flexibility and Efficiency of Use
2. Aesthetic and Minimalist Design
3. Consistency and Standards

Back to Agenda

INTEGRATE THIS TO ANY SITE

INCLUDE THE HEATMAPPING SCRIPT

```
<script src="path/to/heatmapper.js"></script>
```

Add Heatmap Generation Code

```
window.onload = function() {
    generateCoordMap();
};
```

Create a Button to Display the Heatmap

```
<button id="downloadHeatMap"
style="display:none;"
onclick="downloadHeatmap();">Download Heat
Map</button>
```

Enable Heatmap Saving (Optional)

```
function saveHeatmap(dataUrl) {
    fetch('/save-heatmap', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json'
        },
        body: JSON.stringify({ heatmap: dataUrl })
    });
```

Server-Side Handling

```
app.post('/save-heatmap', (req, res) => {
    const dataUrl = req.body.heatmap;
    const base64Data =
dataUrl.replace(/^data:image\/png;base64,/, "");
    fs.writeFile("heatmap.png", base64Data,
'base64', (err) => {
        if (err) console.log(err);
        res.sendStatus(200);
    });
});
```

Heatmap Analysis ML Model

The model can distinguish between correct and incorrect patterns based on the heatmap data, which visualizes user behavior such as mouse movements and clicks.

1	Input Data: Heatmaps, Labels
2	Data Preprocessing: Normalization, Augmentation, Feature Extraction
3	Model Architecture: Convolutional Neural Networks (CNNs), Fully Connected Layers, Activation Functions
4	Training: Dataset, Loss Function, Optimizer, Validation
5	Reasoning Behind Correct Pattern Detection: Correct Patterns, Incorrect Patterns
6	Validation & Performance: Accuracy, Precision, Recall, F1-score,

Back to Agenda

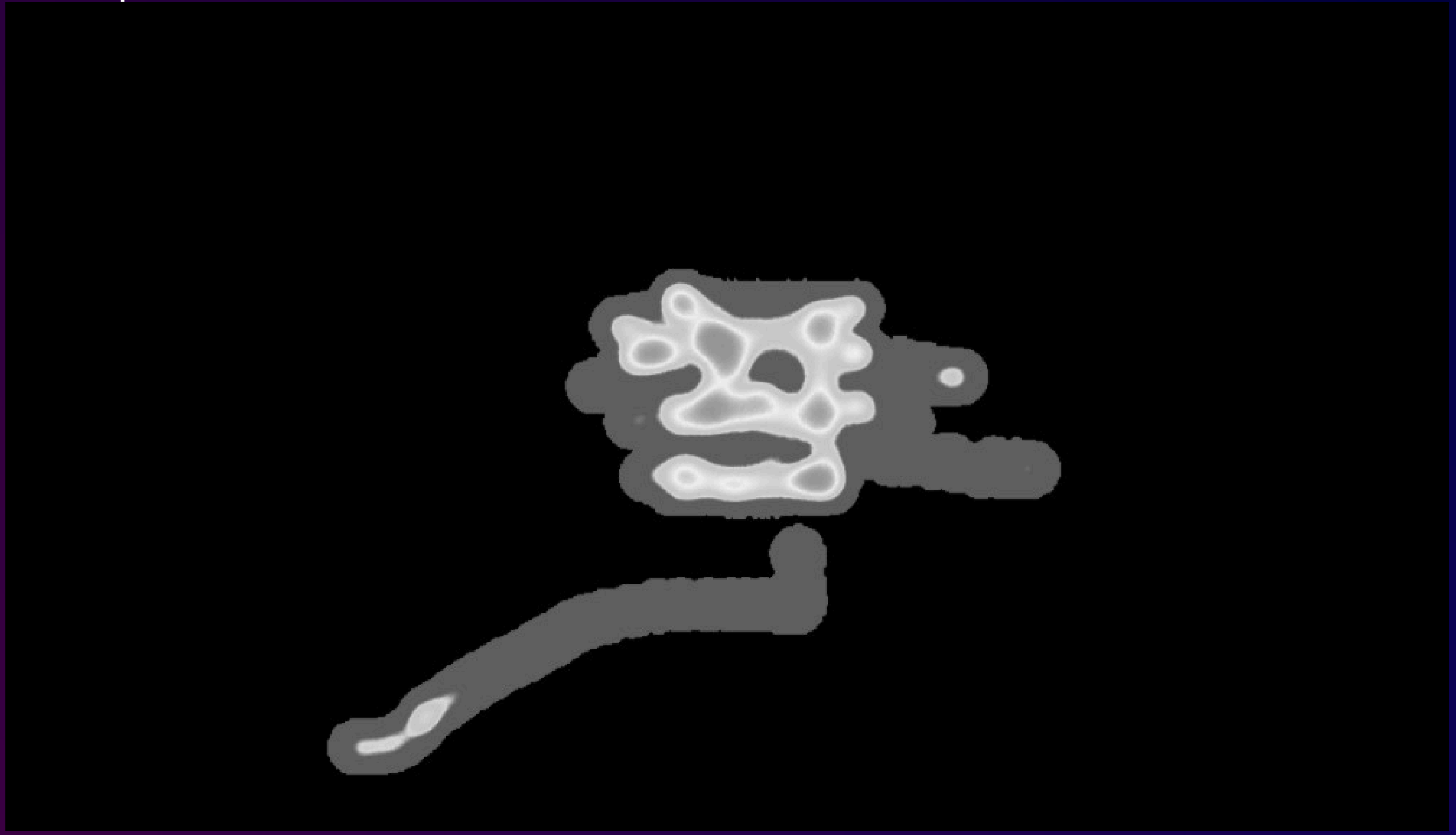
ML Model

```python
# Cell 3: Model Training and Evaluation

from sklearn.model_selection import train_test_split, cross_val_score, GridSearchCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Initialize the classifier
clf = RandomForestClassifier(random_state=42)

# Perform cross-validation
cv_scores = cross_val_score(clf, X, y, cv=5)
print(f'Cross-Validation Scores: {cv_scores}')
print(f'Mean CV Score: {cv_scores.mean()}')

# Hyperparameter tuning using GridSearchCV
param_grid = {
    'n_estimators': [50, 100, 150],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}

grid_search = GridSearchCV(estimator=clf, param_grid=param_grid, cv=5, scoring='accuracy', verbose=2, n_jobs=-1)
grid_search.fit(X, y)

# Best parameters from GridSearchCV
print(f'Best Parameters: {grid_search.best_params_}')

# Train the classifier with best parameters
best_clf = grid_search.best_estimator_

# Split the data into training and testing sets (80% training, 20% testing)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Login Page Example

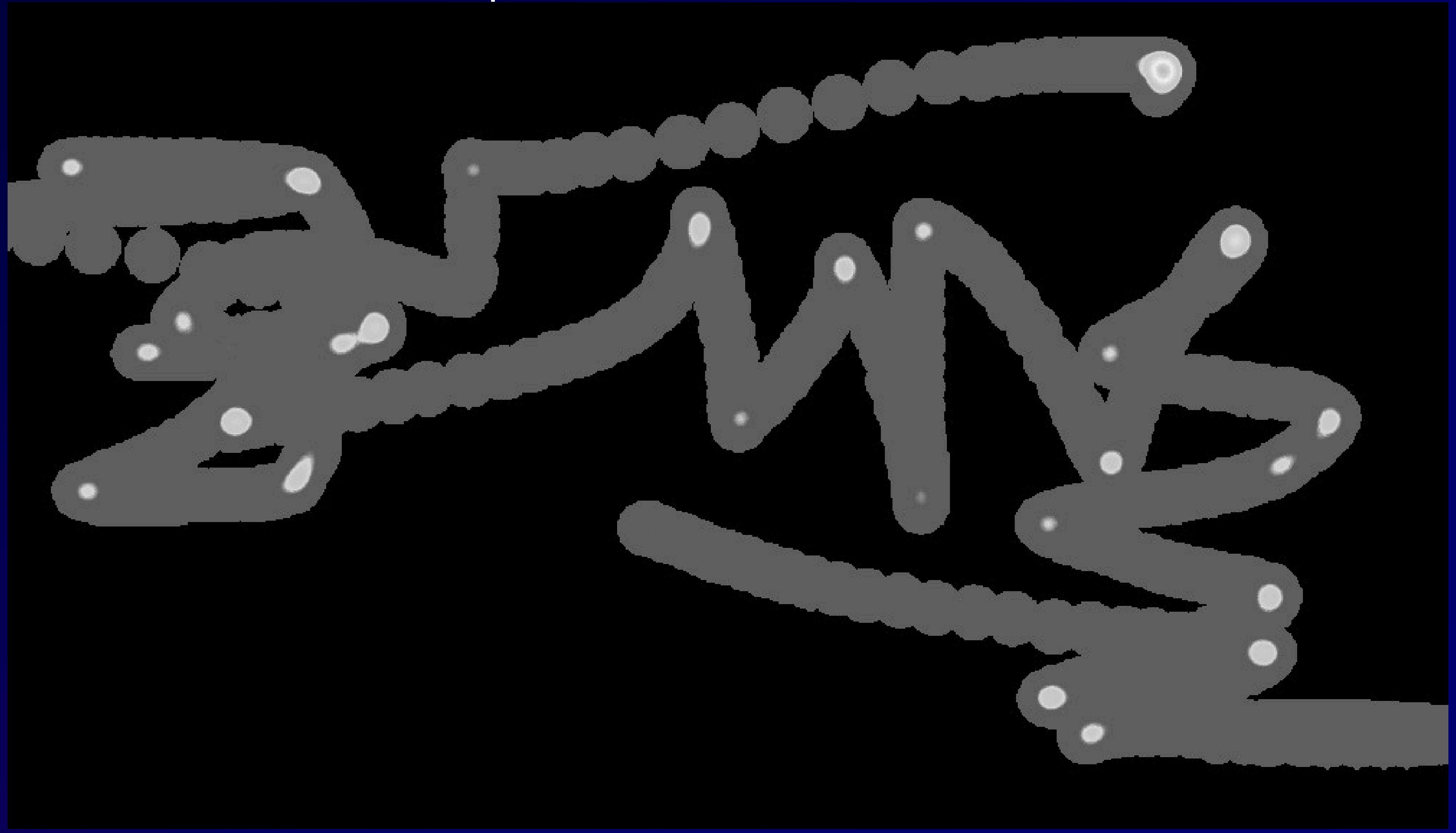
YES Image

The "YES" image represents a correct or well-designed UI layout. In these images, the heatmap likely shows concentrated user activity around important elements like buttons, forms, or key information areas. The design facilitates a smooth user experience, with minimal unnecessary mouse movement or confusion, indicating intuitive navigation and well-placed UI elements.



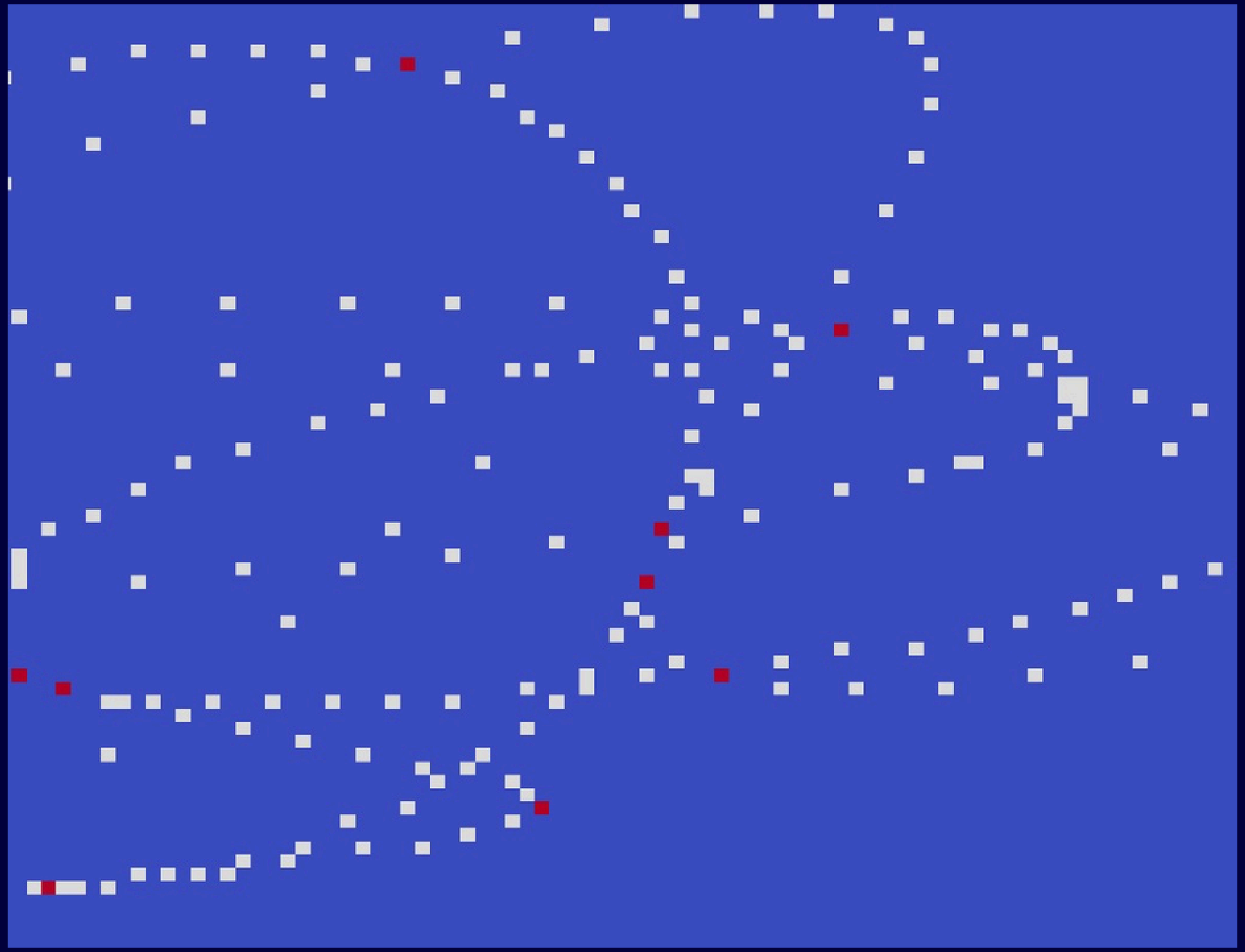
NO Image

The "NO" image depicts an incorrect or poorly designed UI layout. In these images, the heatmap shows scattered or excessive mouse movement, possibly indicating user confusion or difficulty in locating key elements. Users may spend more time hovering over non-interactive or irrelevant areas, signaling that the design causes friction, leading to a less effective user experience.

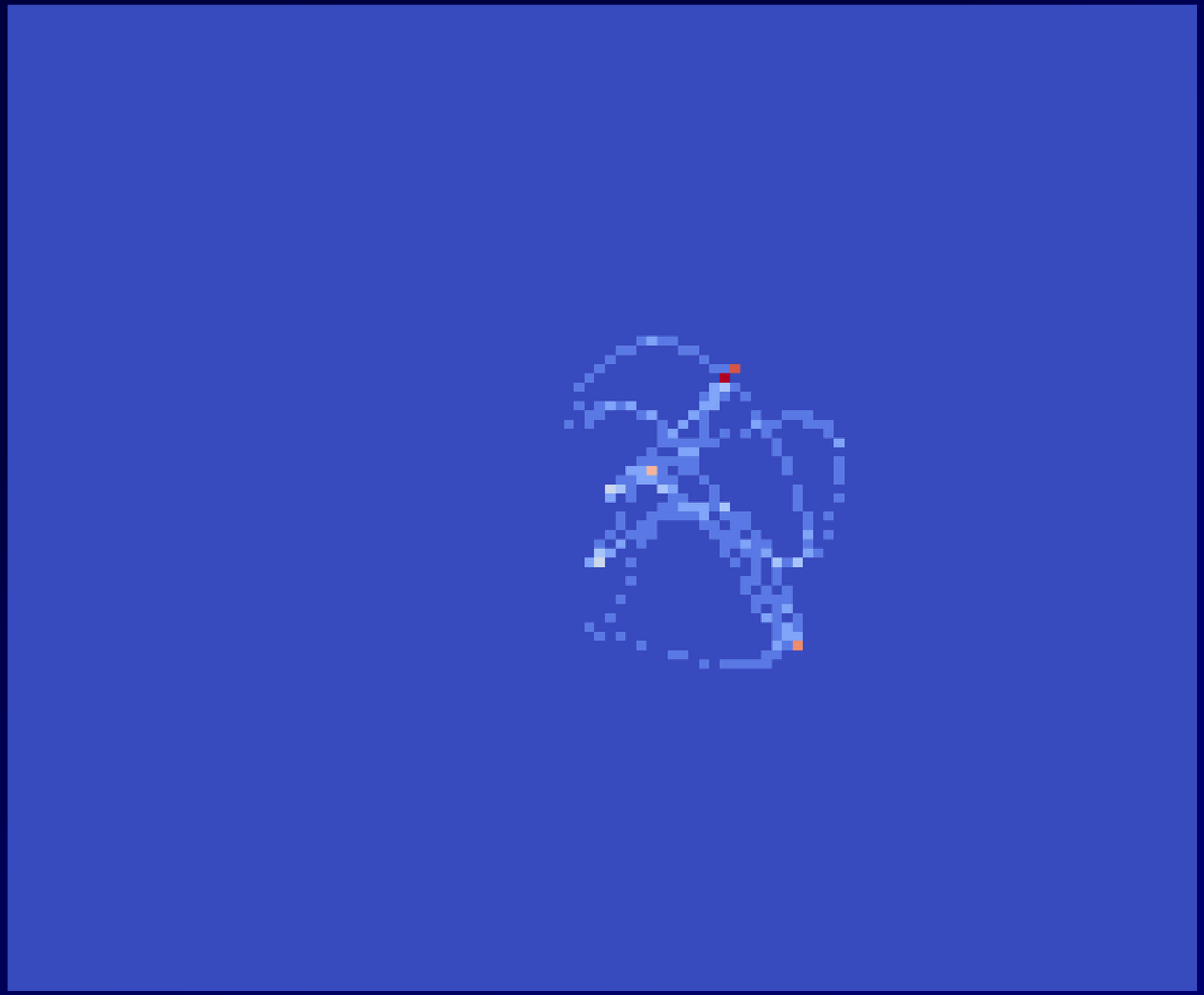


Python Script

```python
for img_index in range(num_images):
    mouse_data = []

    # Generate multiple smooth mouse movement paths
    for _ in range(num_paths):
        # Start and end points for the path with more randomness
        start_x, start_y = np.random.randint(0, width), np.random.randint(0, height)
        end_x, end_y = np.random.randint(0, width), np.random.randint(0, height)

        # Generate more control points for smoother paths
        control_x = np.random.randint(0, width, 6)  # 6 random intermediate control points
        control_y = np.random.randint(0, height, 6)

        # Add start and end points to control points
        x_points = np.concatenate([[start_x], control_x, [end_x]])
        y_points = np.concatenate([[start_y], control_y, [end_y]])

        # Create smooth curve using splines
        tck, u = splprep([x_points, y_points], s=0)
        u_new = np.linspace(u.min(), u.max(), points_per_path)
        smooth_path = splev(u_new, tck)

        # Extract the smooth path points and add to the mouse data
        x_smooth, y_smooth = smooth_path
        mouse_data.extend(list(zip(x_smooth, y_smooth)))

    # Create 2D histogram for heatmap with adjusted bins for scattering
    heatmap, xedges, yedges = np.histogram2d(
        [x for x, y in mouse_data],
        [y for x, y in mouse_data],
        bins=[np.arange(0, width, 15), np.arange(0, height, 15)]  # Larger bin sizes for more scatter
    )

    # Normalize the heatmap for better visibility
    heatmap = np.clip(heatmap.T, 0, 255)

    # Plotting the heatmap using seaborn with a custom colormap (blue -> green -> yellow -> red)
    plt.figure(figsize=(12, 6), facecolor='white')  # Set background to white
    sns.heatmap(heatmap, cmap='coolwarm', cbar=False)  # Using 'coolwarm' colormap
```



```python
for img_index in range(num_images):
    mouse_data = []

    # Generate multiple smooth mouse movement paths
    for _ in range(num_paths):
        # Choose a random hot zone
        zone = hot_zones[np.random.randint(len(hot_zones))]
        x_start = np.random.randint(zone[0], zone[0] + zone[2])
        y_start = np.random.randint(zone[1], zone[1] + zone[3])
        x_end = np.random.randint(zone[0], zone[0] + zone[2])
        y_end = np.random.randint(zone[1], zone[1] + zone[3])

        # Control points for the curve (Bezier-like curve)
        control_x = np.random.randint(zone[0], zone[0] + zone[2], 4)  # Random intermediate control points
        control_y = np.random.randint(zone[1], zone[1] + zone[3], 4)

        # Ensure control points are distinct
        if len(set(control_x)) < len(control_x) or len(set(control_y)) < len(control_y):
            continue

        # Add start and end points to control points
        x_points = np.concatenate([[x_start], control_x, [x_end]])
        y_points = np.concatenate([[y_start], control_y, [y_end]])

        # Create smooth curve using splines
        try:
            tck, u = splprep([x_points, y_points], s=0)
            u_new = np.linspace(u.min(), u.max(), points_per_path)
            smooth_path = splev(u_new, tck)
        except ValueError as e:
            print(f"Skipping path due to error: {e}")
            continue

        # Extract the smooth path points and add to the mouse data
        x_smooth, y_smooth = smooth_path
        mouse_data.extend(list(zip(x_smooth, y_smooth)))

    # Create 2D histogram for heatmap
    heatmap, xedges, yedges = np.histogram2d(
        [x for x, y in mouse_data],
        [y for x, y in mouse_data],
        bins=[np.arange(0, width, 10), np.arange(0, height, 10)]
    )
```



Back to Agenda

MODEL

- Reinforcement Learning with Human-in-the-Loop

OTHER POTENTIAL MODELS

- Transformer Models (e.g., GPT, BERT)
- Explainable AI (XAI) Frameworks

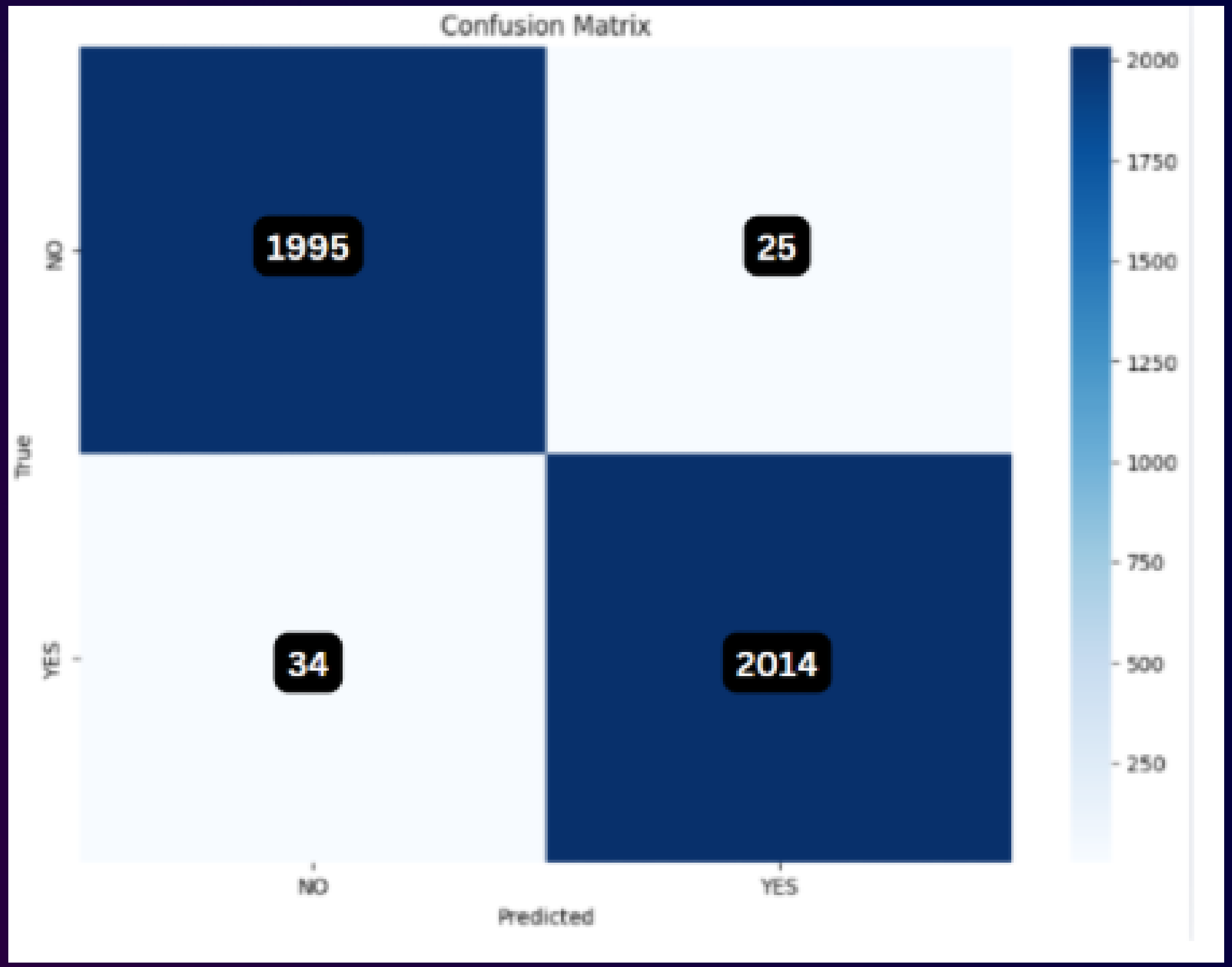
Why HITL is better?

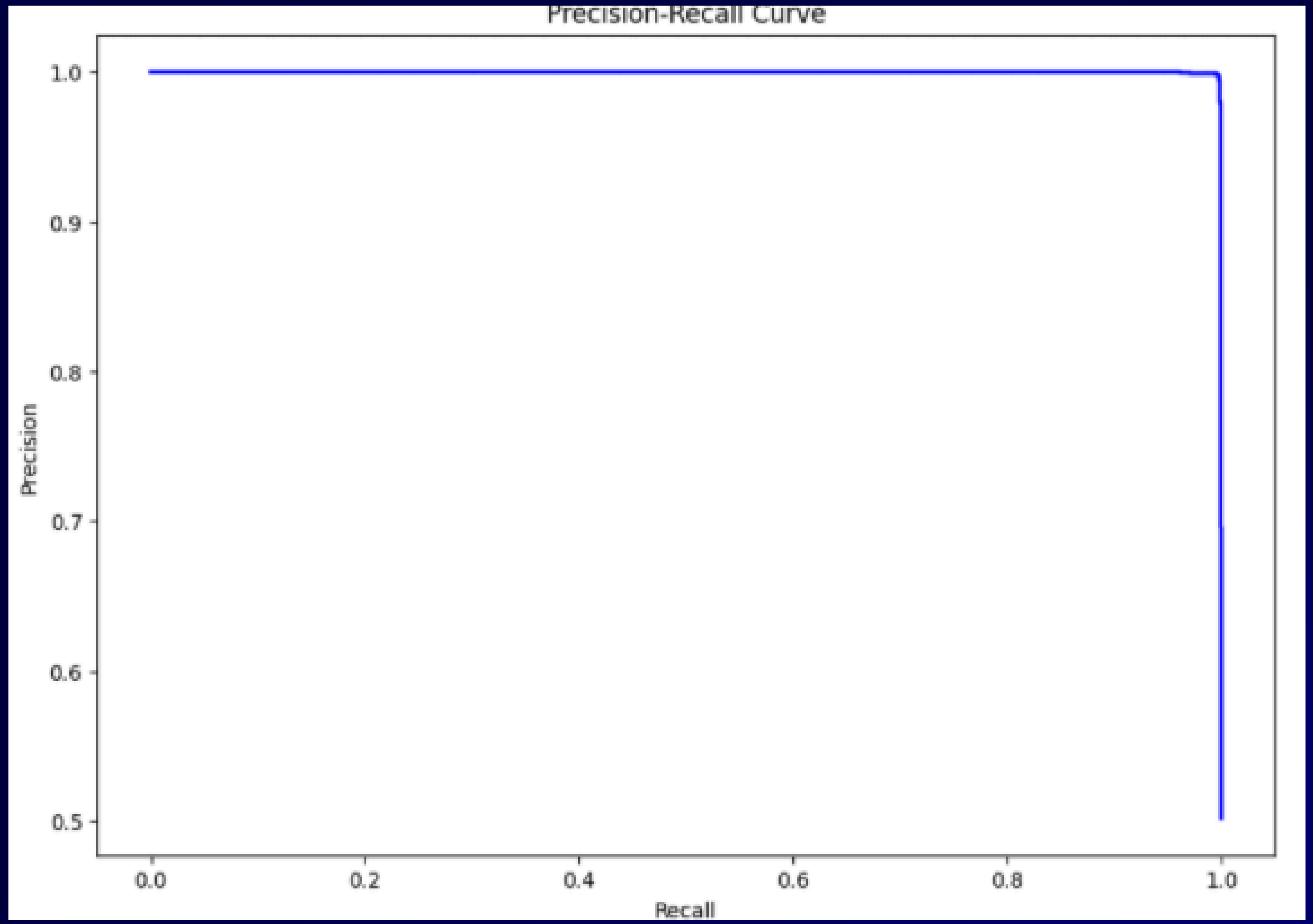
- Dynamic Learning and Adaptation
- Improved Performance in Real-World Scenarios
- User Alignment and Customization
- Better Handling of Complex and Ambiguous Tasks

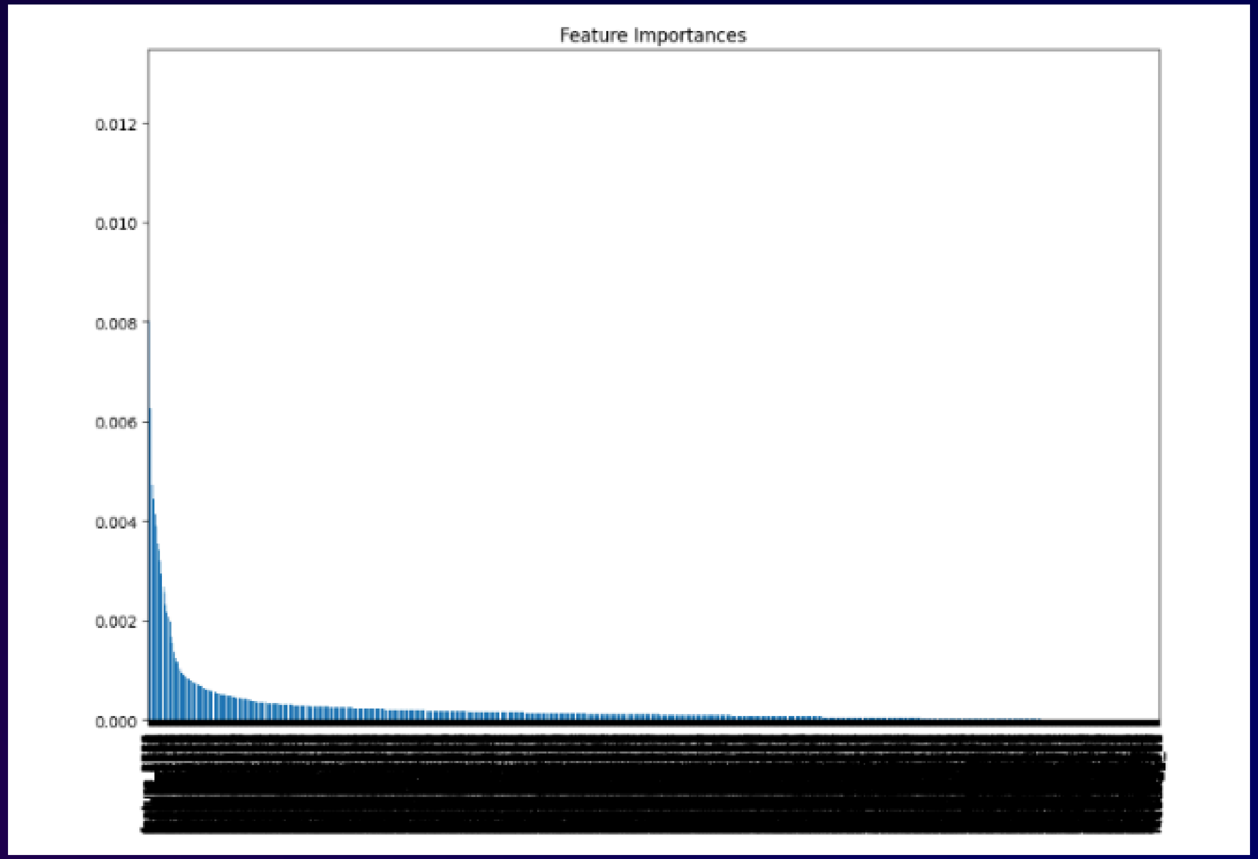
Back to Agenda

Result

```
Cross-Validation Scores: [0.73746313 0.99778761 0.99827925 0.99827883 0.99852471]
Mean CV Score: 0.9460667061721088
Fitting 5 folds for each of 108 candidates, totalling 540 fits
Best Parameters: {'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 100}
Model saved to best_random_forest_model.joblib
Model loaded successfully
Accuracy: 98.55%
Confusion Matrix:
[[1995   25]
 [  34 2014]]
Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.99      0.99      2020
           1       0.99      0.98      0.99      2048

    accuracy                           0.99      4068
   macro avg       0.99      0.99      0.99      4068
weighted avg       0.99      0.99      0.99      4068
```


Confusion Matrix
1995
25
34
2014
NO
YES
True
Predicted
2000
1750
1500
1250
1000
750
500
250


Precision-Recall Curve
Precision
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.2
0.4
0.6
0.8
1.0


Feature Importances
0.012
0.010
0.008
0.006
0.004
0.002
0.000

Conclusion

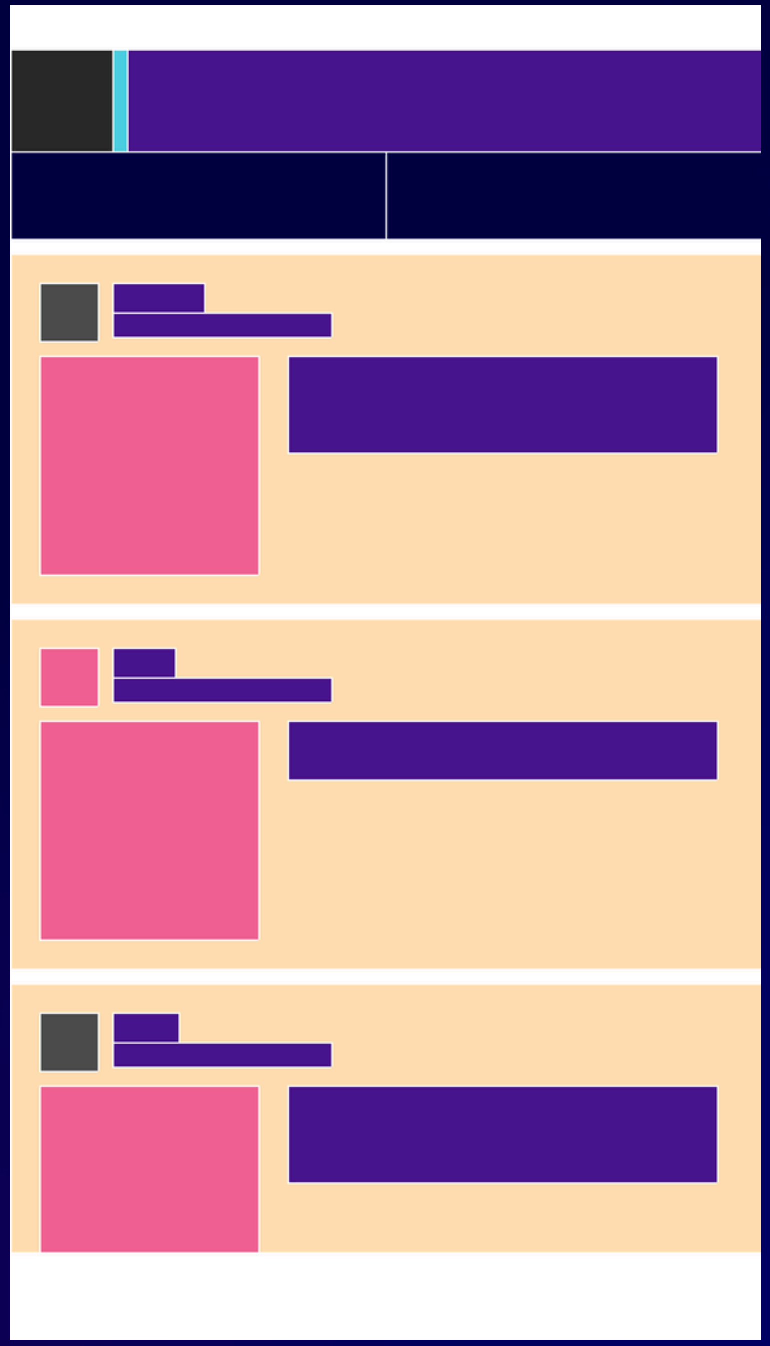
This research has demonstrated the development of an adaptive user interface (UI) system that uses machine learning and real-time interaction data to deliver personalized user experiences. Traditional static UIs often fail to address the diverse behaviors and preferences of users. By leveraging custom heatmapping tools and machine learning models, this system dynamically adjusts UI elements based on individual user interactions, enhancing engagement and satisfaction.

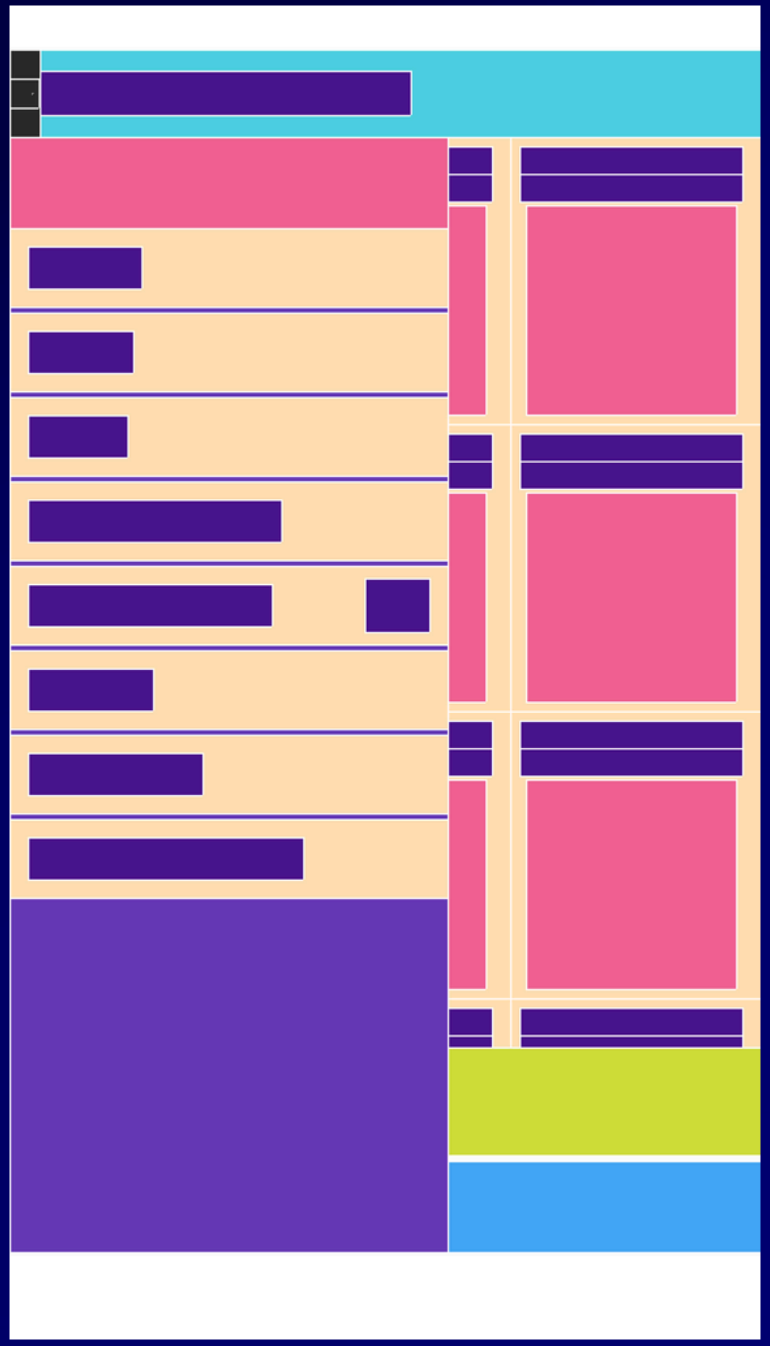
A key aspect of this project was ensuring that dynamic changes to the UI adhered to recognized usability heuristics, such as consistency, user control, and minimizing cognitive load. The integration of Convolutional Neural Networks (CNNs) allowed the system to analyze heatmap data in real time, predicting and implementing changes that improve the user experience without compromising usability.

Future Scope

1) Expansion to Diverse Applications
2) Open-Source Collaboration
3) Incorporation of New Interaction Modalities
4) Cross-Platform Compatibility
5) Integration of RICO Dataset for Benchmarking

Rico Dataset - Mapping





Thank You