Detecting working patterns from online workers and predicting task completion

Janani Chitra Rajendran, Pritham Vinay Jujjavarapu, Sahithi Tatineni
jrajend@clemson.edu, pjujjav@g.clemson.edu, sahitht@g.clemson.edu

Primary Analysis Method Selection

The primary baseline analysis method used is **Random Forest regression.**

a. Appropriateness for Project Objective:

- **Objective:** The project's primary goal was to accurately predict task completion times based on user behavior, platform usage, and task types.
- **Reasoning:** Random Forest regression effectively handles both numerical and categorical data, making it suitable for modeling complex relationships within diverse behavioral data, and thus aligns closely with the project's predictive objective.

b. Appropriateness for the Available Project Data:

- **Data Characteristics:** The dataset included approximately 3.5 million observations with both categorical variables (platform, task subtype, browser events) and numerical features (time-related attributes).
- **Reasoning:** Random Forest is robust against high-dimensionality, effectively handles mixed data types (numerical and categorical after preprocessing), and efficiently manages large-scale datasets. It also reduces variance by averaging multiple decision trees, making it particularly suitable for large, noisy datasets.

c. Appropriateness for Use as a Baseline Model:

- **Baseline Purpose:** A baseline model provides a starting point to benchmark the performance of more advanced models.
- **Reasoning:** Random Forest was chosen due to its simplicity of implementation, interpretability, robustness, and ability to provide reasonably accurate predictions without extensive tuning. Its initial performance helped identify the limitations and guided subsequent enhancements with more advanced methods, like XGBoost.

Methods Implementation

The following methods were used for data pre-processing and feature engineering in preparation for the baseline Random Forest analysis:

1. Data Cleaning:

Dropping Irrelevant Columns:

- Columns such as extra and skip were removed to reduce noise and enhance dataset quality.

Handling Missing Values:

- Categorical missing values were filled with a placeholder "Unknown" to preserve data integrity.
- Numerical missing values were imputed with the median to maintain central tendency without bias.

2. Time-Related Feature Engineering:

Datetime Conversion:

- Raw timestamps (milliseconds) were converted to a human-readable datetime format for interpretability and analysis.

Cyclic Encoding:

- The hour feature was derived from timestamps and cyclically encoded (using sine and cosine transformations) to effectively capture periodic patterns inherent to time-based data.

3. Categorical Variable Encoding:

One-hot Encoding:

- Categorical features (browser-event, platform, task-subtype, type, user) were one-hot encoded, transforming them into numerical format suitable for the Random Forest algorithm.

4. Numerical Feature Standardization:

- Numerical features were standardized using StandardScaler, ensuring consistent scales and preventing any single feature from disproportionately influencing the model's results.

5. Dataset Splitting:

- The cleaned and processed dataset was split into training (80%) and test sets (20%), ensuring reliable evaluation and generalizability of model performance.

Approach Used for Data Splitting:

Train-Test Split was used with 80% data allocated for training and 20% for testing.

Hyperparameter Selection for Baseline Model:

- The Random Forest regression baseline model involved hyperparameter tuning.
- Grid Search with 3-fold cross-validation was employed to systematically find optimal hyperparameters.
- The hyperparameters tuned were:
  Number of estimators (n_estimators): tested values [100, 200, 300]
  Maximum depth of trees (max_depth): [10, 20, 30, None]
  Minimum samples required to split a node (min_samples_split): [2, 5, 10]
  Minimum samples required per leaf node (min_samples_leaf): [1, 2, 4]
- The optimal selected parameters after tuning were:
  n_estimators : 200
  max_depth : 20
  min_samples_split : 5
  max_samples_split : 2

Evaluation Metrics Used:

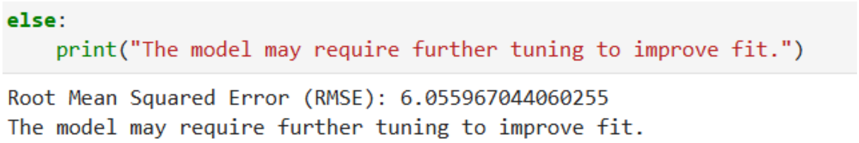
- Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) were used as metrics to evaluate model performance.

- Residual analysis (residual plots) was performed to visually assess the model's accuracy and potential biases.

Quantitative Analysis:

Random Forest Model:

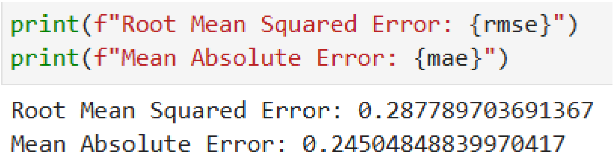
Metric	Value
Root Mean Squared Error(RMSE)	6.05596



```
else:
    print("The model may require further tuning to improve fit.")

Root Mean Squared Error (RMSE): 6.055967044060255
The model may require further tuning to improve fit.
```

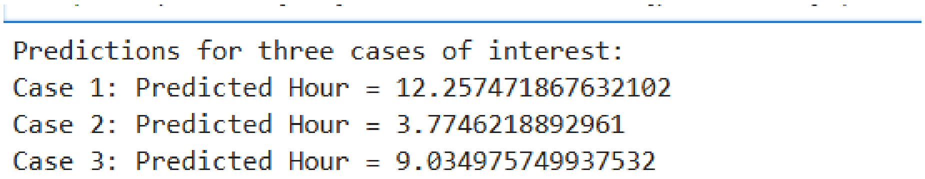
Optimized Random Forest Model:

Metric	Value
Root Mean Squared Error(RMSE)	0.2877
Mean Absolute Error(MAE)	0.2450



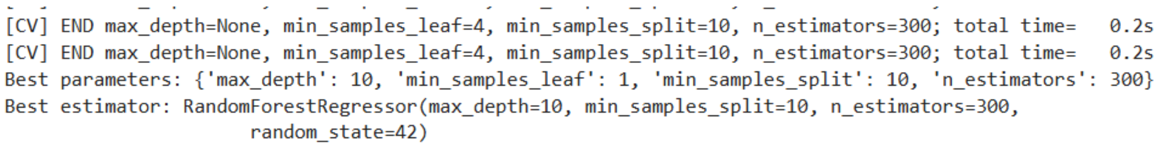
```
print(f"Root Mean Squared Error: {rmse}")
print(f"Mean Absolute Error: {mae}")

Root Mean Squared Error: 0.287789703691367
Mean Absolute Error: 0.24504848839970417
```

Prediction of Three cases of interest:

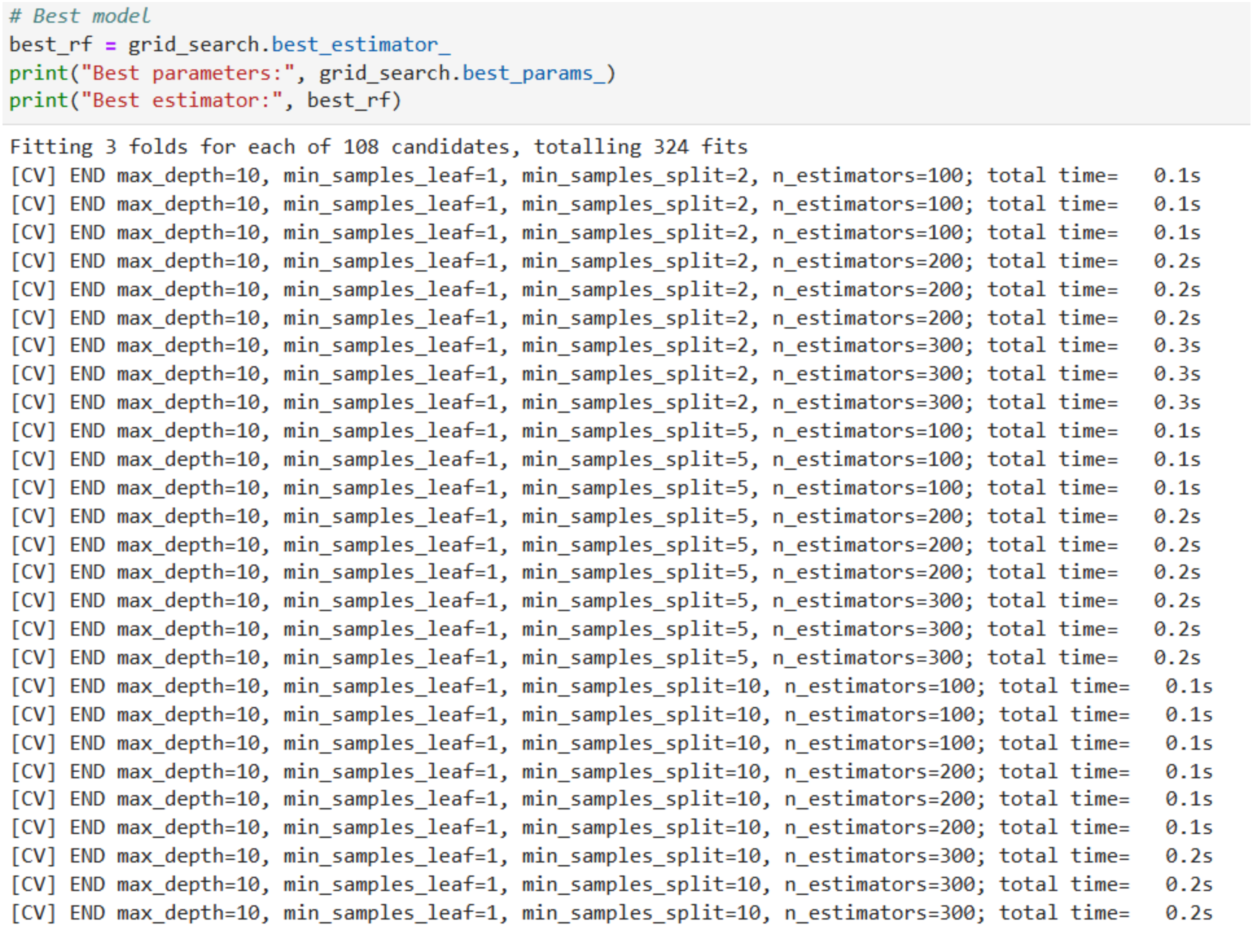


```
Predictions for three cases of interest:
Case 1: Predicted Hour = 12.257471867632102
Case 2: Predicted Hour = 3.7746218892961
Case 3: Predicted Hour = 9.034975749937532
```

Grid search best parameter and best estimator:



```
[CV] END max_depth=None, min_samples_leaf=4, min_samples_split=10, n_estimators=300; total time=   0.2s
[CV] END max_depth=None, min_samples_leaf=4, min_samples_split=10, n_estimators=300; total time=   0.2s
Best parameters: {'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 10, 'n_estimators': 300}
Best estimator: RandomForestRegressor(max_depth=10, min_samples_split=10, n_estimators=300,
                      random_state=42)
```

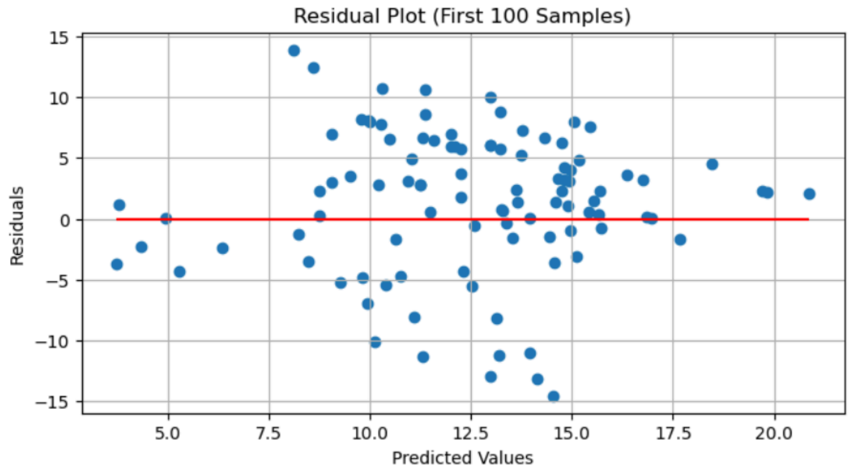


```
# Best model
best_rf = grid_search.best_estimator_
print("Best parameters:", grid_search.best_params_)
print("Best estimator:", best_rf)
```

```
Fitting 3 folds for each of 108 candidates, totalling 324 fits
[CV] END max_depth=10, min_samples_leaf=1, min_samples_split=2, n_estimators=100; total time=   0.1s
[CV] END max_depth=10, min_samples_leaf=1, min_samples_split=2, n_estimators=100; total time=   0.1s
[CV] END max_depth=10, min_samples_leaf=1, min_samples_split=2, n_estimators=100; total time=   0.1s
[CV] END max_depth=10, min_samples_leaf=1, min_samples_split=2, n_estimators=200; total time=   0.2s
[CV] END max_depth=10, min_samples_leaf=1, min_samples_split=2, n_estimators=200; total time=   0.2s
[CV] END max_depth=10, min_samples_leaf=1, min_samples_split=2, n_estimators=200; total time=   0.2s
[CV] END max_depth=10, min_samples_leaf=1, min_samples_split=2, n_estimators=300; total time=   0.3s
[CV] END max_depth=10, min_samples_leaf=1, min_samples_split=2, n_estimators=300; total time=   0.3s
[CV] END max_depth=10, min_samples_leaf=1, min_samples_split=2, n_estimators=300; total time=   0.3s
[CV] END max_depth=10, min_samples_leaf=1, min_samples_split=5, n_estimators=100; total time=   0.1s
[CV] END max_depth=10, min_samples_leaf=1, min_samples_split=5, n_estimators=100; total time=   0.1s
[CV] END max_depth=10, min_samples_leaf=1, min_samples_split=5, n_estimators=100; total time=   0.1s
[CV] END max_depth=10, min_samples_leaf=1, min_samples_split=5, n_estimators=200; total time=   0.2s
[CV] END max_depth=10, min_samples_leaf=1, min_samples_split=5, n_estimators=200; total time=   0.2s
[CV] END max_depth=10, min_samples_leaf=1, min_samples_split=5, n_estimators=200; total time=   0.2s
[CV] END max_depth=10, min_samples_leaf=1, min_samples_split=5, n_estimators=300; total time=   0.2s
[CV] END max_depth=10, min_samples_leaf=1, min_samples_split=5, n_estimators=300; total time=   0.2s
[CV] END max_depth=10, min_samples_leaf=1, min_samples_split=5, n_estimators=300; total time=   0.2s
[CV] END max_depth=10, min_samples_leaf=1, min_samples_split=10, n_estimators=100; total time=   0.1s
[CV] END max_depth=10, min_samples_leaf=1, min_samples_split=10, n_estimators=100; total time=   0.1s
[CV] END max_depth=10, min_samples_leaf=1, min_samples_split=10, n_estimators=100; total time=   0.1s
[CV] END max_depth=10, min_samples_leaf=1, min_samples_split=10, n_estimators=200; total time=   0.1s
[CV] END max_depth=10, min_samples_leaf=1, min_samples_split=10, n_estimators=200; total time=   0.1s
[CV] END max_depth=10, min_samples_leaf=1, min_samples_split=10, n_estimators=200; total time=   0.1s
[CV] END max_depth=10, min_samples_leaf=1, min_samples_split=10, n_estimators=300; total time=   0.2s
[CV] END max_depth=10, min_samples_leaf=1, min_samples_split=10, n_estimators=300; total time=   0.2s
[CV] END max_depth=10, min_samples_leaf=1, min_samples_split=10, n_estimators=300; total time=   0.2s
```

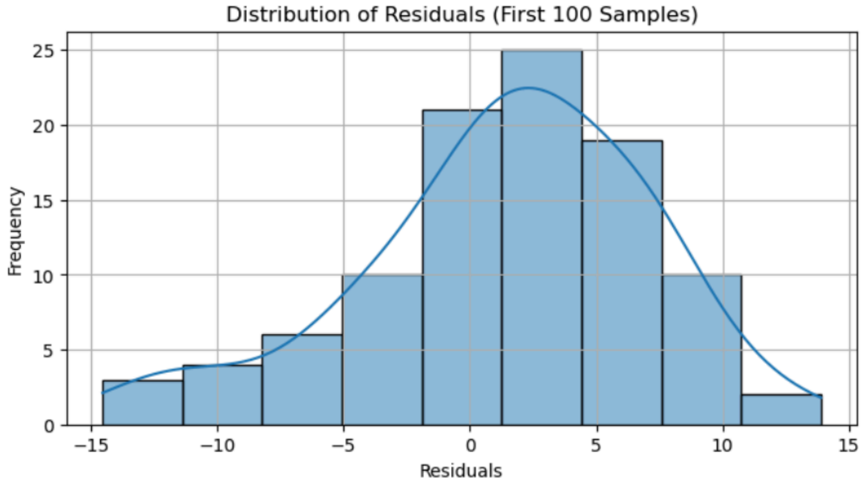
Visualizations of Results:

Results obtained from the Random Forest model:

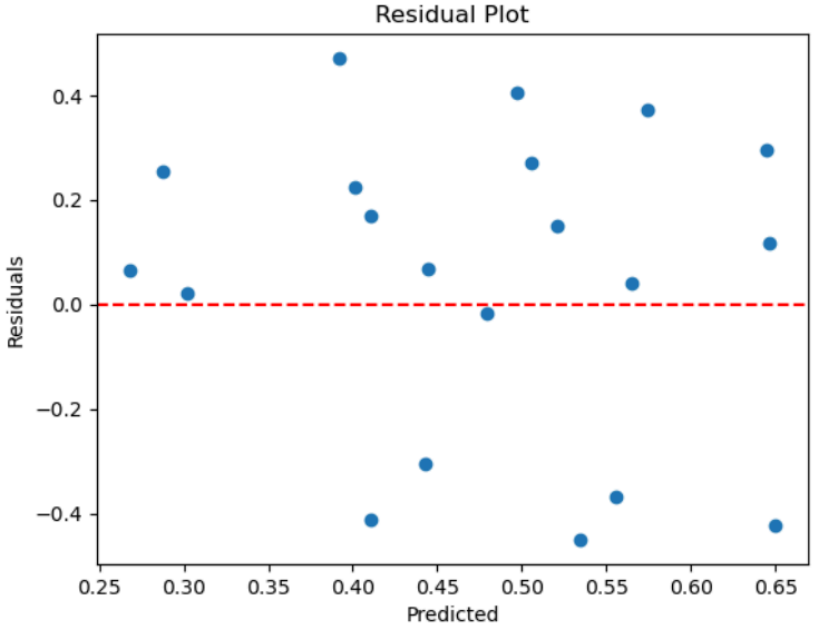


Residuals show model errors

Distribution of Residuals:



Evaluation of Optimized Random Forest Model:



Interpretation of Baseline Analysis Results:

Performance Quality Assessment:

- The initial Random Forest regression baseline model delivered moderate accuracy, indicated by an RMSE of **6.055 hours**, highlighting its capability to capture general patterns but also its limitations in precise predictions.

- After systematic hyperparameter tuning (Grid Search) and enhanced preprocessing, the optimized baseline significantly improved, achieving an RMSE of **0.2877 hours** and MAE of **0.2450 hours**, showing substantial reduction in prediction errors and higher model reliability.

Conclusions Drawn:

- The substantial decrease in RMSE and MAE after optimization confirms the effectiveness of systematic hyperparameter tuning and careful preprocessing, emphasizing their crucial role in enhancing predictive accuracy.
- Random Forest proved to be a robust baseline, effectively modeling complex, high-dimensional user behavior data, while also clearly identifying opportunities for further refinement through more advanced techniques such as XGBoost.

Limitations of Results:

- Despite improved accuracy, Random Forest models have inherent limitations in handling complex nonlinear patterns without extensive feature engineering.
- Residual plots from the initial baseline revealed clear patterns, indicating biases or unaccounted-for variations. While optimization significantly mitigated these, small prediction errors still remain, suggesting potential further improvements with more sophisticated models.