

Bundle Buddy: Personalized Outfit Recommendations for Weather Conditions

Olivia O'Donnell

12/16/25

Motivation

- Choosing the wrong outfit for the day causes discomfort
- There are many factors to how we experience the weather, which makes it hard to determine what to wear
- Everyone experiences the same weather differently








Background

The Problem:

- Predicting comfortable outfit choices based on weather is a **classification problem**
- Need to map weather conditions & activity level → appropriate outfit category

Dataset:

- 150 samples per tester with 5 input features:

- Temperature (°F) 
- Humidity (%) 
- Precipitation (mm) 
- Wind Speed (mph) 
- Activity Level (1=low, 2=moderate, 3=high) 

- 5 output classes: jacket, sweater, long_sleeve, t_shirt, t_shirt_and_shorts

Why Machine Learning?

- Complex, non-linear relationships between weather variables
- Individual comfort preferences vary
- Historical patterns can predict future comfort

Related Work

Tool	Similarities	Differences
DressMyRun	Recommends running outfits based on the weather	Not personalized and only focuses on exercise
Smart Thermostats	Learns user temperature preferences	Not focused on clothing choices
Cladwell	Recommends outfits based on the weather	Not personalized

No existing solution exists to learn your personal comfort level in different outfits and weather conditions, which is where Bundle Buddy differentiates itself

Claim / Target Task

I believe it is possible to predict a user's most comfortable outfit for the day's weather with high accuracy based on their past experiences

Proposed Solution

- Random Forest Model
- Each datapoint represents a weather experience
- Feedback mechanism
 - System adjusts outfit category if user reports discomfort
 - Automatically retrains model with new data
- Datapoint features
 - From OpenWeatherMap API
 - Temperature
 - Wind
 - UV Index
 - Humidity
 - Self-reported
 - Activity Level
 - Outfit Category

Implementation

Data Pipeline:

1. **Data Collection:** OpenWeatherMap API fetches current weather by zip code, users enter weather experiences through the UI
2. **Preprocessing:** StandardScaler normalizes features (temperature, humidity, precipitation, wind_speed, activity_level)
3. **Model Training:** Random Forest learns patterns from historical outfit choices
4. **Prediction:** Model recommends outfit for current conditions
5. **Feedback Loop:** Users report comfort level → model retrains automatically

Technical Stack:

- Python 3 with scikit-learn, pandas, numpy
- Interactive UI: Jupyter widgets (ipywidgets)
- Real-time weather: OpenWeatherMap API
- Dataset: CSV storage with automatic updates

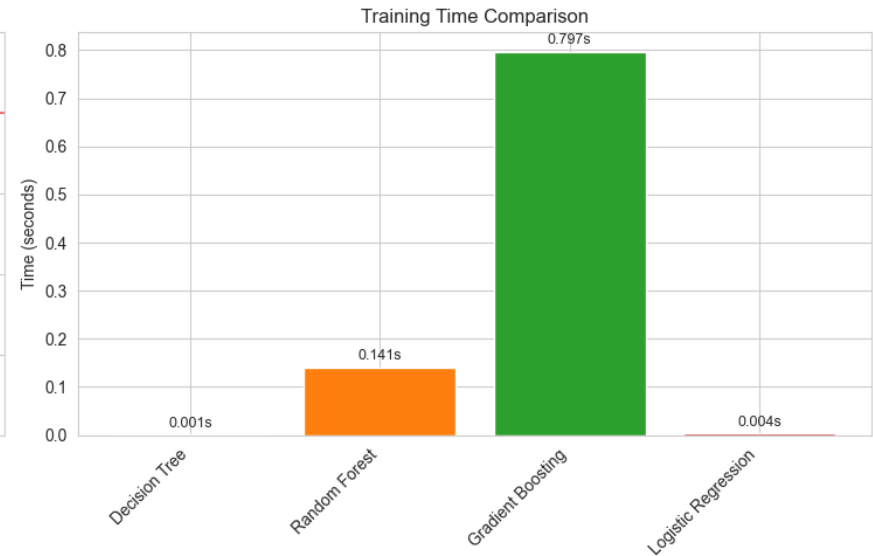
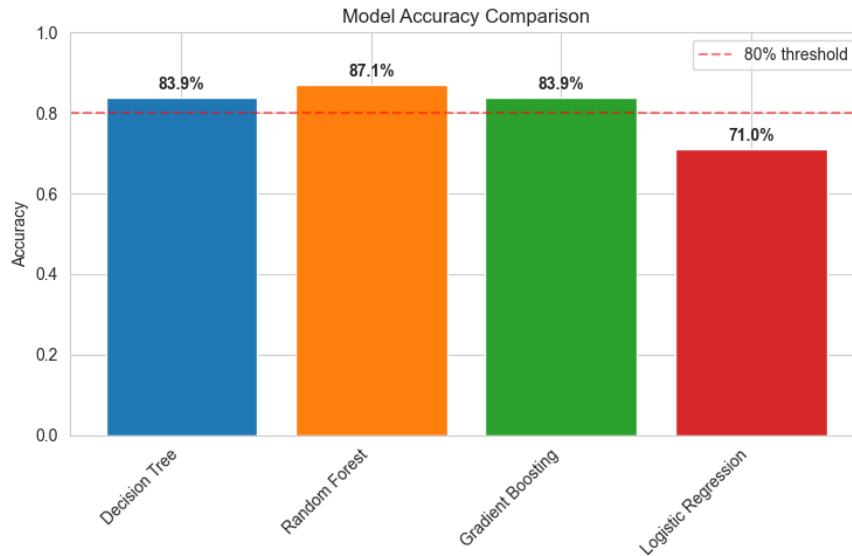
Key Features:

- Instant predictions based on zip code
- Automatic model retraining after feedback

Data Summary

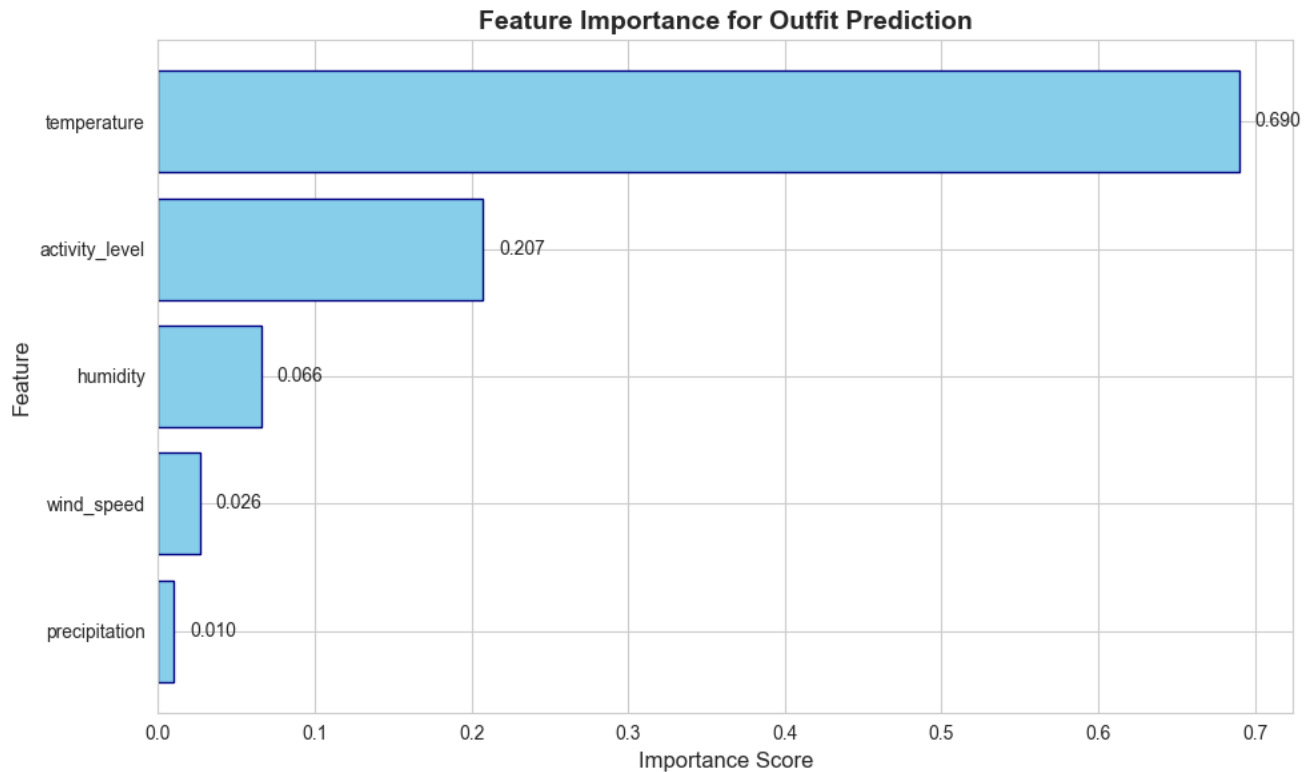
- My friend and I have been collecting data about our comfort levels each day of the semester and the accompanying weather conditions to use for the data
- Having two testers will allow us to see how the model becomes personalized
- Data has been collected over the course of multiple months to capture temperature variation

Experimental Results



Random Forest model had the highest accuracy
Overall Accuracy: 87.1%

Experimental Results



Temperature is the most important factor, but other factors still have notable impact

Experimental Analysis

- Why Random Forest Succeeds Here
 - Robust to noise/overfitting
 - Logistic Regression is too simple
 - Decision Tree may overfit
 - Gradient Boosting has a much longer training time so although it could be better with more tuning, Random Forest is more practical
- Temperature as the most important factor
 - Makes intuitive sense as temperature is the main thing most people consider
 - Activity level also matters a lot, as exercise generates body heat
 - Wind, precipitation, and humidity less important

Conclusion and Future Work

- Bundle Buddy uses a Random Forest model to learn each user's preferences and recommend outfits appropriate for the weather, reducing decision fatigue and improving daily comfort
- Potential Extensions
 - Experiment with pre-populating some data based on average comfort so recommendations start off more accurate
 - Morning notifications automatically recommending their outfit for the day
 - Add LLM explanation for the outfit recommendation

References

APIs & Tools:

- OpenWeatherMap API: <https://openweathermap.org/api>
- Scikit-learn Documentation: <https://scikit-learn.org/>
- Pandas Documentation: <https://pandas.pydata.org/>
- Jupyter Notebooks: <https://jupyter.org/>

Dataset:

- Custom-generated outfit preference data
- Weather data from OpenWeatherMap