



ToWear

Quick overview

User-Interface Vision

- ❖ User signs up for the app (email, username, password, zip code)
- ❖ User creates a closet with the clothes they have (just entering the types of clothes)
- ❖ User presses one button every day to get a suggestion of what they should wear
- ❖ User can improve the suggestions and make them more personal by giving the app more data; specifically, the user can enter what they wore each day and how they felt, OR they can just answer some basic questions about themselves (do you wear a lot of layers in the winter? Do you like wearing hats?)

Behind the Front-End

- ❖ User signs up for the app (email, username, password, zip code)
 - ❖ Zip code is used for the weather
 - ❖ We use an API called pyOWM where we can send a zip code and retrieve a Json file with the temperature, and other weather factors included

Behind the Front-End

- ❖ User creates a closet with the clothes they have (just entering the types of clothes)
 - ❖ Every type of clothing has a specific warmth score relating to each body part
 - ❖ A jacket can have a warmth of [0, 5, 0, 0]
 - ❖ This means that a jacket has a 0/10 effect on the warmth of its wearer's head, 5/10 effect on top, 0/10 on bottom, and 0/10 effect on feet
 - ❖ The user does not have to worry about these numbers; the user needs only to add clothes to their closets by specifying their types (and, optionally, names)
 - ❖ Notes: this is different than the current state of the app as of 1/22/2020 where users must enter the warmth scores themselves

Behind the Front-End

- ❖ User presses one button every day to get a suggestion of what they should wear
 - ❖ The suggestions are made using the user's closet, the user's predicted desired temperature, and the user's predicted warmth coefficients (more on that soon)
 - ❖ The suggestions are in the form of a list of lists, where the outer list includes four elements representing head, top, bottom, and feet. The inner lists include the suggested garment combination
 - ❖ Example: `[['pom pom hat'], ['winter coat'], ['jeans', 'shorts'], ['sneakers and light socks']]`

User's Desired Temperature

- ❖ This involves the concept that each user has a desired temperature they want to be at. Some people get warm faster than others. Their desired temperature could be as low as 90 Fahrenheit. Other people are always cold. Their desired temperature could be as high as 110 Fahrenheit.
- ❖ IMPORTANT: if the weather outside is the same as the user's desired temperature, then their perfect outfit for that weather is [0,0,0,0]
- ❖ We start all users at a desired temperature of 98.6, the average normal body temperature
- ❖ When users provide us with more data about how they felt in their outfits with the given outside temperature, we can start to predict their actual desired temperature

User's Warmth Coefficients

- ❖ Some people feel warmer on top than they do on bottom; so, we see some people wearing shorts and long sleeves in the winter, or many layers but nothing on their heads
- ❖ We represent this with another four-element list called the warmth coefficients list. Each element represents a part of the body: head, top, bottom, and feet. The four elements must add up to 1
- ❖ Take for example, [0.2, 0.3, 0.3, 0.2]. This means that 20 percent of the person's temperature comes from the warmth of their head, while 30 percent comes from the warmth of their top, and so on
- ❖ This helps us decide which outfit to suggest to the user, alongside their desired temperature, both of which can be predicted as we previously noted on slide 6

Math

- ❖ Perfect Outfit Temperature = Desired Temperature - Outside Temperature
- ❖ If the weather outside is 60, and the user's desired temperature is 100, we need an outfit with warmth of 40
- ❖ If the person's coefficients are [0.1, 0.4, 0.4, 0.1], then we multiply 40 by each one of these coefficients and get [4, 16, 16, 4] as the perfect outfit, which accurately adds up to 40
- ❖ Note: all temperature is scaled to a number between 0 and 40

Math

- ❖ Perfect Outfit Temperature = Desired Temperature - Outside Temperature
- ❖ If the weather outside is 60, and the user's desired temperature is 100, we need an outfit with warmth of 40
 - ❖ Scaled by 40/134, the weather outside is 17.91, the desired temperature is 29.86, and the outfit warmth is 11.94
- ❖ If the person's coefficients are [0.1, 0.4, 0.4, 0.1], then we multiply 11.94 by each one of these coefficients and get [1.19, 4.78, 4.78, 1.194] as the perfect outfit

Real Example

- ❖ Temperature: 22
- ❖ Secret Coefficients are .05 .35 .35 .25
- ❖ Desired temperature is 100
- ❖ Suggested outfit: `[['beanie'] ['winter coat', 'long sleeve shirt'] ['jeans', 'shorts'] ['winter boots and thick socks']]`
- ❖ [2, 7, 7, 6]

Training

- ❖ Training amount is how much data in the training set to include. The training set has the weather – appropriate outfit pairs
- ❖ Currently, 1/22/2020, the user gives their secret coefficients, desired temperature, training amount, and the algorithm tries to predict these secret coefficients and desired temperatures. Simply put, the learning process is simulated
- ❖ Starting at a default desired temperature and coefficients, and with real user input every day for 4 weeks*, the AI can start making accurate, personalized suggestions

*an alternative is to ask users basic questions one time that help us approximate their coefficients and desired body temperature

Money?

- ❖ Sell data of what people are wearing to department store chains and clothing companies
- ❖ Allow ads when perfect outfits cannot be created using the garments in the user's closet (you need to purchase this to get a perfect outfit!)
- ❖ Charge for allowing users to interact with other users with the same desired temperature and warmth coefficients so that they can share outfit ideas!

FIXME

- ❖ Make garments that affect multiple parts of the body, such as coats
- ❖ Create database for users
- ❖ Output pictures for the users to see their outfits
- ❖ Allow style and chemistry points for garments (i.e. a matching score for outfits to be more fashionable)
- ❖ Organize users closets into seasons
- ❖ Suggest outfits based on events?