

# Make Me Up

## Introduction

### Problem and Motivation

Make-up artists use many techniques to select their colour palette to create a polished and professional look, whether it is strikingly bold or composed and subtle. Professional make-up artists need to learn colour theory and lighting. The make-up hobbyist may still enjoy working with make-up while only having a limited understanding of those techniques. Every make-up artist is also subject to boredom and want to create a variety of looks. Some achieve this through rolling a die to choose random colours. Others through following others on Instagram or YouTube for inspiration.

*Make Me Up* is an artificial intelligence recommender system which offers the user combinations of make-up looks, depending on a profile that the user inputs. *Make Me Up* is accessible to all users, from beginners who are starting to understand colour theory, or the advanced user seeking inspiration. *Make Me Up* will provide recommendations based on known make-up heuristics and classify colours as a warm-temperature or cool-temperature

### Application Domain

Being skilled with make-up depends on understanding colour theory at a deep level. Complementary colours (those that are opposite each other on the colour wheel) are used to cancel each other out. Certain colours will draw attention to the eyes and make those features more noticeable, giving them a 'pop'. Colours tend to be either 'warm' (reds, oranges, yellows) or 'cool' (purple, blue, green). Skin tone becomes more complicated as skin can be classified into a 'warm temperature' with yellow undertones, or a 'cool temperature' with blue undertones. A common heuristic is that warm make up suits warm-toned people, and cool make-up suits cool-toned people.

### Style Me!

*Make Me Up* was proposed as a re-implementation of *Style Me!* by H. Wang<sup>1</sup>. *Style Me!* is an artificial intelligence which pairs dresses and shoes and scores them based on events and user profile. *Make Me Up* took another path since information was missing, it wasn't clear what classifiers were used. They also differ in terms of technology, *Make Me Up* is a web application using Python 2.7 and followed tutorials to implement the neural network. *Style Me!* is a Java application which leverages the WEKA<sup>2</sup> library.

---

<sup>1</sup> Wang, H. "Machine Fashion: An Artificial Intelligence Based Clothing Fashion Stylist." N.p., 2014. Web. 5 Mar. 2017. <[https://www.ai.uga.edu/sites/default/files/theses/wang\\_haosha.pdf](https://www.ai.uga.edu/sites/default/files/theses/wang_haosha.pdf)>

<sup>2</sup> <http://www.cs.waikato.ac.nz/ml/weka/>

## Artificial Intelligence

### Recommender

The main application of *Make Me Up* is the style engine (**style\_engine.py**), a recommender system which pairs eyeshadows and lipsticks together and assigns the pairs a score based on how suited they are to the current user profile. The style engine reads in all eyeshadows and lipsticks from the database and pairs them into all possible combinations. Next, the style engine determines a score and popularity measure for every pair.

$$Score_{pair} = \frac{C_{Likes} - C_{Dislikes}}{C_{Likes} + C_{Dislikes} + 1}$$

The score is the average of the ‘likes’ and ‘dislikes’ assigned to the pair. The number of ‘likes’ and ‘dislikes’ for each component of the pair are read in from the database and assigned as the initial value. These ‘likes’ and ‘dislikes’ are values which the user has assigned through the user interface. When the system is first initialized or reset, these values are 0. Next, the style engine runs the pair through a set of heuristics (included as Table 1). Each heuristic is either positive (warm eyeshadow with warm lipstick) or negative (bold eyeshadow and bold lipstick paired together). If the pair matches the heuristic, then either a positive counter or a negative counter is incremented. When the style engine is finished with heuristics, the number of positive heuristics is added to the ‘likes’ and the number of negative heuristics is added to the ‘dislikes.’ Thus the score is assigned.

For one of the heuristics, a Nearest Neighbour classifier is used to determine the closest colour name for any hex value (**colour\_classifier.py**). This classifier is used to find a colour name for the colour closest to the hex value of the eyeshadow. The classifier works by representing each colour as a point in a 3D space, where the red value is x, the green value is y, and the blue value is z. The distance between the unknown colour point and the known points are used to determine the nearest colour. The colour name is then used to determine if the colour compliments the user’s eye colour. There are three lists of colour names suiting each eye colour, one for blue, one for green, and one for brown. The colour names and values used by *Make Me Up* come from the Wikipedia list of Crayola colours<sup>3</sup> and are saved as a JSON list in **data/colours.json**.

$$Popularity = Score - 0.03 * C_{Wears}$$

Popularity is a measure of boredom. The more a user wears a particular eye shadow or lipstick, the less likely they will want to wear it again. The popularity of a pair is the score subtracted by the weighted average number of wears of a pair. The weight 0.03 comes from *Style Me!* but could be adjusted.

---

<sup>3</sup> [https://en.wikipedia.org/wiki/List\\_of\\_Crayola\\_colored\\_pencil\\_colors](https://en.wikipedia.org/wiki/List_of_Crayola_colored_pencil_colors)

Table 1: Heuristics used by *Make Me Up*

Heuristic	Positivity
A bold eyeshadow paired with a bold lip	Negative
A bold eyeshadow paired with a nude lip	Positive
A subtle eyeshadow paired with a bold lip	Positive
The lipstick temperature does not match the eyeshadow temperature	Negative
The lipstick temperature matches the eyeshadow temperature	Positive
The lipstick temperature matches the user's skin temperature	Positive
The primary colour of the eyeshadow is in the subset of colours matching the user's eye colour	Positive
The secondary colour of the eyeshadow is in the subset of colours matching the user's eye colour	Positive

## Neural Network

*Make Me Up* also includes a neural-network classifier (**neural\_network.py**). Wang mentions using a multi-layer perceptron model in Style Me! to classify the pairs of data; it is not clear what classes they were using. *Make Me Up* classifies colours into 'warm' or 'cool' temperature. The neural network is trained using the list of all colours from Wikipedia<sup>4</sup> as a dataset and testing using 5-fold testing. The dataset is available in data/temps.csv. The training dataset was manually classified using the heuristic that a warm colour will contain more red content than a cool colour, although a better heuristic should be used.

The neural-network was implemented in **neural\_network.py** by following the tutorial provided by WildML<sup>5</sup>. The number of input features from our dataset was 3, the number of output features was 2, and the number of hidden layers used was 3. The activation function for the hidden layer is tanh. The tutorial and algorithm use batch gradient descent with a fixed learning rate. The data is iterated through 20,000 times, the model values are adjusted slightly and the loss is calculated

## Design Choices

### Server Modules

*Make Me Up* is separated into the following modules: **server.py**, **style\_engine.py**, **colour\_classifier.py**, **neural\_network.py**, and **mm\_database.py**. The modules are separated based on their functionality. The server (**server.py**) module provides the interaction between the HTML client and the server. It passes interactions from the webpage to the style engine. The style engine (**style\_engine.py**) module assigns a score to every pair of eyeshadow and lipstick. The scores are determined by heuristics within the style engine, as discussed in the previous section. The style engine is the only module to communicate with the database module (**mm\_database.py**). The database module provides an interface between *Make Me Up* and the sqlite3 database containing the eyeshadows and lipsticks.

<sup>4</sup> List of all colours

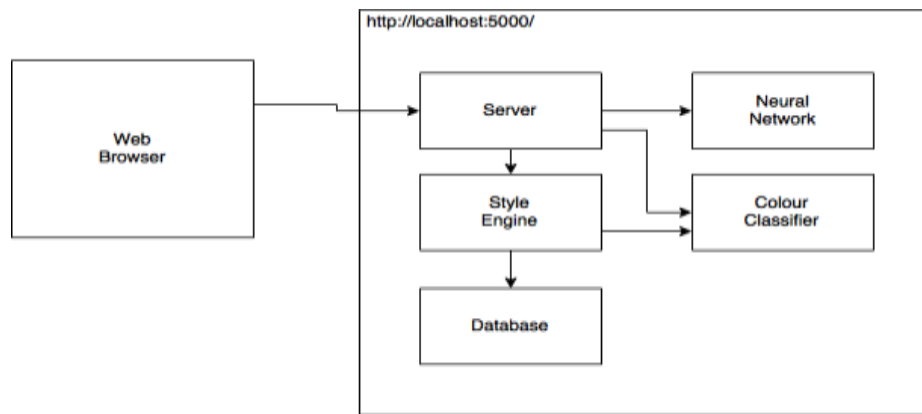
<sup>5</sup> <http://www.wildml.com/2015/09/implementing-a-neural-network-from-scratch/>

The colour name classifier (**colour\_classifier.py**) contains the functionality to determine the closest colour name to a given hex value. Both the server and the style engine use the colour classifier. The server allows a user to submit a hex value through a form to receive the nearest colour name. The style engine uses the colour classifier to find the colour name of an eyeshadow, the colour name is used to determine if the eyeshadow colour will compliment the user's eye colour. The colour temperature classifier is the neural network and contained in the neural network module (**neural\_network.py**). The colour temperature is used by the server to allow a user to train the dataset and find the temperature of a colour.

## User Interface

The user interface of *Make Me Up* is served as HTML pages. The HTML pages are generated from Jinja templates using Flask. The user interface simply displays the information stored in the system and the results of the algorithms, it does not possess any intelligence of its own. No CSS was used to style the HTML pages.

**Figure 1:** Components of Make Me Up



## Database Schema

There are three tables in the makemeup database. One table contains lipsticks, another contains eyeshadows, and the third contains pairs of lipsticks and eyeshadows.

**Table 2:** Lipstick Table Scheme

Column(s)	Value
<u>Id</u>	Numeric
Texture	"MATTE", "GLOSSY"
Company	String
Name	String
Intensity	"BOLD", "NUDE"
Colour Family	"CORAL", "PINK", "RED", "PURPLE", "DARK", "NUDE"
Red, Green, Blue	Integer between 0 and 255
Likes, Dislikes, Wear	Positive Integer
Score	Real between 0 and 1

**Table 3:** Eyeshadow Table Schema

Column(s)	Value
<u>Id</u>	Numeric
Texture	"MATTE", "SHINE"
Company	String
Name	String
Style	"SUBTLE", "BOLD", "METALLIC"
Temperature	"WARM", "COOL"
Primary Red, Primary Green, Primary Blue	Integer between 0 and 255
Secondary Red, Secondary Green, Secondary Blue	Integer between 0 and 255
Likes, Dislikes, Wears	Positive Integer
Score	Real between 0 and 1

**Table 4:** Pairs Table Schema

Column	Value
<u>Eyeshadow Id</u>	Numeric
<u>Lipstick Id</u>	Numeric
Score	Real between 0 and 1
Popularity	Real between 0 and 1

## Results

### Recommender

The recommender will show different results depending on the user profile. The following screenshots show the user interface of the recommender as it is interacted with and updated.










**Figure 2:** The interface for the recommender for a user with blonde hair, blue eyes, a cool skin tone

Username: Lindsay Hair Colour: BLONDE Eye Colour: BLUE Skin Temperature: COOL Skin Tone: 15

Recommendations									
Update Scores		Reset Likes/Dislikes		Reset Wears					
Images	Score	Popularity	Eyeshadow	Lipstick	Like	Dislike	Wear it!		
	0.4	0.4	AVON Tenderness	COLOURPOP The Twirl	Like (0)	Dislike (0)	Worn 0 times	Wear!	
	0.4	0.4	AVON Tenderness	COLOURPOP Fresh Cut	Like (0)	Dislike (0)	Worn 0 times	Wear!	
	0.4	0.4	AVON Tenderness	COLOURPOP Bumble	Like (0)	Dislike (0)	Worn 0 times	Wear!	
	0.4	0.4	AVON Tenderness	COLOURPOP Creeper	Like (0)	Dislike (0)	Worn 0 times	Wear!	
	0.4	0.4	AVON Tenderness	COLOURPOP Saigon	Like (0)	Dislike (0)	Worn 0 times	Wear!	
	0.4	0.4	AVON Tenderness	COLOURPOP Viper	Like (0)	Dislike (0)	Worn 0 times	Wear!	
	0.4	0.4	AVON Tenderness	COLOURPOP More Better	Like (0)	Dislike (0)	Worn 0 times	Wear!	
	0.4	0.4	AVON Tenderness	COLOURPOP Zipper	Like (0)	Dislike (0)	Worn 0 times	Wear!	
	0.4	0.4	AVON Tenderness	COLOURPOP LAX	Like (0)	Dislike (0)	Worn 0 times	Wear!	

**Figure 3:** The same recommendations once the users has 'liked' some combinations, 'disliked' others, and worn some combinations.

Username: Lindsay Hair Colour: BLONDE Eye Colour: BLUE Skin Temperature: COOL Skin Tone: 15

Recommendations									
Update Scores		Reset Likes/Dislikes		Reset Wears					
Images	Score	Popularity	Eyeshadow	Lipstick	Like	Dislike	Wear it!		
	0.727272727273	0.712272727273	MAYBELLINE Autumn Coppers	COLOURPOP Finder's Keepers	Like (6)	Dislike (0)	Worn 1 times	Wear!	
	0.666666666667	0.651666666667	MAYBELLINE Autumn Coppers	COLOURPOP Limbo	Like (4)	Dislike (0)	Worn 1 times	Wear!	
	0.625	0.61	MAYBELLINE Autumn Coppers	COLOURPOP Airplane Mode	Like (3)	Dislike (0)	Worn 1 times	Wear!	
	0.625	0.61	MAYBELLINE Autumn Coppers	COLOURPOP Speed Dial	Like (3)	Dislike (0)	Worn 1 times	Wear!	
	0.625	0.61	MAYBELLINE Autumn Coppers	COLOURPOP Trap	Like (3)	Dislike (0)	Worn 1 times	Wear!	
	0.625	0.595	MAYBELLINE Autumn Coppers	COLOURPOP Knotty	Like (3)	Dislike (0)	Worn 2 times	Wear!	
	0.625	0.61	MAYBELLINE Autumn Coppers	COLOURPOP Fairy Floss	Like (3)	Dislike (0)	Worn 1 times	Wear!	
	0.625	0.61	MAYBELLINE Autumn Coppers	COLOURPOP Aquarius	Like (3)	Dislike (0)	Worn 1 times	Wear!	
	0.625	0.61	MAYBELLINE Autumn Coppers	COLOURPOP Bestie	Like (3)	Dislike (0)	Worn 1 times	Wear!	

**Figure 4:** The user interface for another user with black hair, green eyes, and a warm skin tone.










## Make Me Up

[Update the user profile](#) [Classify some colours](#)

### Current user

Username: User 2 Hair Colour: BLACK Eye Colour: GREEN Skin Temperature: WARM Skin Tone: 25

### Recommendations

Update Scores			Reset Likes/Dislikes		Reset Wears					
Images	Score	Popularity	Eyeshadow	Lipstick	Like	Dislike	Wear it!			
	0.75	0.75	AVON Tenderness	COLOURPOP The Twirl	<a href="#">Like (0)</a>	<a href="#">Dislike (0)</a>	<a href="#">Worn 0 times</a>	<a href="#">Wear!</a>		
	0.75	0.75	AVON Tenderness	COLOURPOP Fresh Cut	<a href="#">Like (0)</a>	<a href="#">Dislike (0)</a>	<a href="#">Worn 0 times</a>	<a href="#">Wear!</a>		
	0.75	0.75	AVON Tenderness	COLOURPOP Bumble	<a href="#">Like (0)</a>	<a href="#">Dislike (0)</a>	<a href="#">Worn 0 times</a>	<a href="#">Wear!</a>		
	0.75	0.75	AVON Tenderness	COLOURPOP Creeper	<a href="#">Like (0)</a>	<a href="#">Dislike (0)</a>	<a href="#">Worn 0 times</a>	<a href="#">Wear!</a>		
	0.75	0.75	AVON Tenderness	COLOURPOP Saigon	<a href="#">Like (0)</a>	<a href="#">Dislike (0)</a>	<a href="#">Worn 0 times</a>	<a href="#">Wear!</a>		
	0.75	0.75	AVON Tenderness	COLOURPOP Viper	<a href="#">Like (0)</a>	<a href="#">Dislike (0)</a>	<a href="#">Worn 0 times</a>	<a href="#">Wear!</a>		
	0.75	0.75	AVON Tenderness	COLOURPOP More Better	<a href="#">Like (0)</a>	<a href="#">Dislike (0)</a>	<a href="#">Worn 0 times</a>	<a href="#">Wear!</a>		
	0.75	0.75	AVON Tenderness	COLOURPOP Zipper	<a href="#">Like (0)</a>	<a href="#">Dislike (0)</a>	<a href="#">Worn 0 times</a>	<a href="#">Wear!</a>		
	0.75	0.75	AVON Tenderness	COLOURPOP LAX	<a href="#">Like (0)</a>	<a href="#">Dislike (0)</a>	<a href="#">Worn 0 times</a>	<a href="#">Wear!</a>		

## Neural Network

The accuracy of the neural network is determined through k-fold testing. Table X shows the accuracy of testing data/temps.csv while varying the number of hidden layers. As the number of layers increases, the running time of the algorithm also increases. 4 hidden layers has the highest accuracy in the range of 3 layers to 10 layers. A higher accuracy can be achieved with 25 layers, but the run time is noticeably slower. The accuracy decreases again by the time 50 layers are used.

**Table 5:** The results of the neural network

Number of Hidden Layers	Accuracy	Prediction for #0000FF
3	63.1168831169	COOL
4	79.6103896104	COOL
5	62.5974025974	COOL
6	62.987012987	WARM
7	62.5974025974	COOL
8	61.8181818182	WARM
9	62.8571428571	WARM
10	64.025974026	COOL
25	89.8701298701	COOL
50	62.8571428571	WARM

## Future Work

Although *Make Me Up* is a complete system, it is still in the proof-of-concept stages. The system would benefit the most from an increase in available data and heuristics. By increasing the number of eyeshadows and lipsticks in the database, there will be more options in terms of pairs. The more pairs there are, the more likely there is a combination that suits the current user. The top 25 pairs returned to the user will have higher scores. By increasing the number of heuristics used, the scores will be more accurate to the real world. For this project, I made up the heuristics used by the style engine. Heuristics obtained by talking with make-up artists would be far superior.

*Make Me Up* could also be improved in terms of functionality: by including more options instead of just pairs, by allowing the user to filter the results based on products they already own, and by allowing the user to specify an event. Including more options besides eyeshadow and lipstick means including other type of makeup, such as blush, eyeliner, highlight. Instead of scoring pairs, the entire combination would be scored. The more products added leads to an exponential number of combinations but would provide a more complete look. Further, allowing the user to specify that they do not want to wear a product (such as blush) would improve the experience for the user. By allowing the user to filter the recommendations based on products they already own, the user would be able to use *Make Me Up* at home in a practical way. This feature will include storing users in the database and allowing them to enter what products they have at home. Allowing the user to specify an event to wear makeup for



would be an enhancement to their profile. The scores would be updated to match the parameters of the event, but the likes, dislikes and wears would not be affected.

The classification of colour temperatures would be improved through the training data. Currently, the colour temperature data used to train the network is classified manually and determined through a very simple measurement; whether the content of red is higher than the content of blue. Colour temperature is more complex; multiple variables can be leveraged to determine the actual temperature. Colour can also be very subjective, and will even objectively vary depending on the monitor it is displayed on.

The neural network would be improved through more advanced techniques. More advanced techniques for gradient descent are minibatch gradient descent and stochastic gradient descent. The learning rate should decay over time to provide a better classification. Using other activation functions will provide different results.

## References

### Technology

[Flask] <http://flask.pocoo.org>

[Jinja] <http://flask.pocoo.org/docs/0.12/templating/>

### Style Me!

Wang, H. "Machine Fashion: An Artificial Intelligence Based Clothing Fashion Stylist." N.p., 2014. Web. 5 Mar. 2017. [https://www.ai.uga.edu/sites/default/files/theses/wang\\_haosha.pdf](https://www.ai.uga.edu/sites/default/files/theses/wang_haosha.pdf)

### Colour Names

[https://en.wikipedia.org/wiki/Lists\\_of\\_colors](https://en.wikipedia.org/wiki/Lists_of_colors)

[https://en.wikipedia.org/wiki/List\\_of\\_Crayola\\_colored\\_pencil\\_colors](https://en.wikipedia.org/wiki/List_of_Crayola_colored_pencil_colors)

### Colour Classifier

<http://gauth.fr/2011/09/get-a-color-name-from-any-rgb-combination/>

### Neural Networks

<http://www.wildml.com/2015/09/implementing-a-neural-network-from-scratch/>

<http://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/>

<http://cs231n.github.io/optimization-1/>

<http://cs231n.github.io/neural-networks-3/>

<http://colah.github.io/posts/2015-08-Backprop/>

## Running the Code

### List of Files

- colour\_classifier.py
- mm\_database.py
- neural\_network.py
- server.py
- style\_engine.py
- data/
  - colours.json
  - eyeshadow.csv
  - lipsticks.csv
  - makemeup.db
  - temps.csv
- static/
  - images/
- templates/
  - classification.html
  - index.html
  - user\_profile.html

### Requirements

The code for *Make Me Up* was written with Python 2.7 and leverages a sqlite3 database. Both are included with MacOS.

Note: for the server to successfully run, the database must already exist with the correct tables.

The dependencies for Python are as follows

- sqlite3
- Flask
- Numpy

### Command (Unix)

```
export FLASK_APP=server.py  
python -m flask run
```

The URL of the application is

<http://localhost:5000/>

### Github Link

<https://github.com/lindsaybangs/makemeup>