The probability of meeting your almost-doppelgänger

Jack William Mann

13th April 2026

---

1 Motivation

Whenever I get on my bus to sixth form, there's a man that looks just like me: same hair, same face, and (unfortunately for him) the same height. Naturally, I started doing whatever any rational person would do; I scribbled some calculations and graphs, and asked myself:

"What are the odds of stumbling across a lookalike?"

2 Introduction

It is tempting to think that encountering a near-identical stranger is rather unlikely; after all, appearance is shaped by so many variables. Yet in the UK, even rare combinations emerge with surprising frequency. The problem is not whether lookalikes exist, but how to quantify their likelihood!

This essay models similarity by isolating key traits and assigning them probabilistic structure, applying different distributions and analysis with links to fascinating concepts like Shannon's information theory.

We first define our key variables…

A_0: age in years
G_0: biological gender
W_0: BMI
E_0: ethnicity
H_0: hair colour

These variables are limited but form a structured system throughout the paper. We will estimate the number of encounters per year with a *potential* lookalike (same key variables but may not be an exact lookalike!) based on our inputs into the variables above, assuming we are in the UK (*and assuming we pay no attention to anybody around us if we travel abroad…)*

Also, in chapter 5, I show my working methods but sadly could not for everything due to the word limit.

This model should be interpreted as a first-order approximation rather than a precise predictive system.

3 Age variable

Age is a primary measure contributing to the probability of meeting an almost-doppelgänger. Age constrains the pool of potential matches.

First, we define the variables:

A_0 = age input
A_1 = another person's age (random variable from UK population)

Assume UK age follows a normal distribution:

$$A_1 \sim N(\mu, \sigma^2)$$

Where:
μ = mean age of population
σ = standard deviation of ages

Using the ONS data for population by age[1]:

$$\mu = \frac{\sum xf}{\sum f}$$

Where:

$x = 0, 1, 2, \ldots, 90$
f = population at each age

$$\therefore \ \mu = 39.71$$

And

$$\sigma = \sqrt{\frac{\sum f(x-\mu)^2}{\sum f}}$$

$$\sigma = 23.74$$

$$\sigma^2 = 563.59$$

$$\therefore A_1 \sim N(39.71, 563.59)$$

Now that we have a distribution approximation, we must consider a tolerance function. It is clear that as age increases, it is harder to distinguish between persons of *similar* ages. E.g. a 1-year-old will almost always look younger than a 3-year-old, whereas a 60-year-old may look younger than a 50-year-old in some cases! Hence, our tolerance must increase with age to represent the ambiguity associated with older ages.

[1] https://www.ons.gov.uk/visualisations/dvc3349/fig05/datadownload.xlsx?

Let $t(A_0)$ be our tolerance function.

$$\therefore |A_1 - A_0| \leq t(A_0)$$

This defines the acceptable age interval for similarity,

Assume:

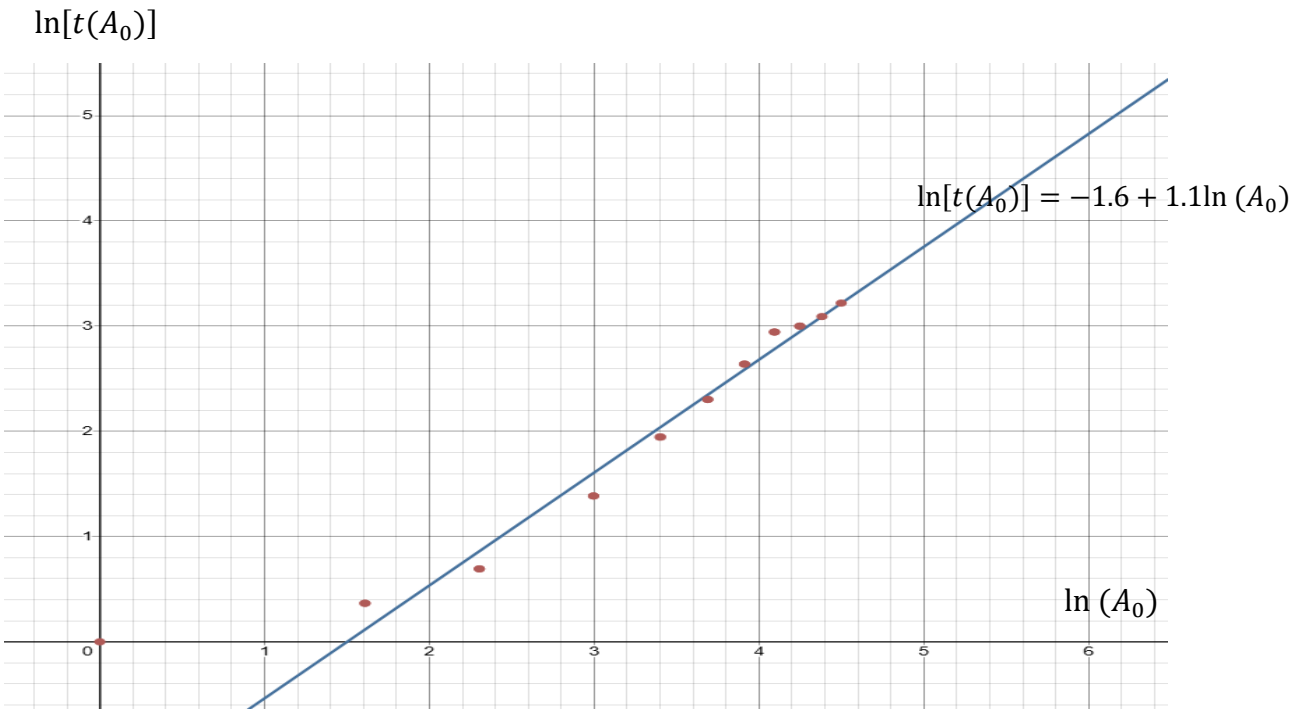
$$t(A_0) = aA_0^b$$

We will now generate approximate data for tolerance per age and find a suitable curve for the data. However, the data is subjective, introducing slight error to the function.

A_0	$t(A_0)$	$\ln(A_0)$	$ln[t(A_0)]$
1	1.0	0	0
5	1.4	1.609	0.336
10	2	2.303	0.693
20	4	2.996	1.386
30	7	3.401	1.946
40	10	3.689	2.303
50	14	3.912	2.639
60	19	4.094	2.944
70	20	4.248	3.000
80	22	4.382	3.091
90	25	4.499	3.219

Transform to log-log space for linear regression.

$$\ln[t(A_0)] = \ln(a) + b\ln(A_0$$



We must now convert the regression line to our original power model by exponentiating both sides.

$$\ln[t(A_0)] = -1.6 + 1.1\ln(A_0)$$

$$t(A_0) = e^{-1.6} \cdot A_0^{1.1}$$

$$e^{-1.6} = 0.202$$

$$\therefore t(A_0) = 0.202A_0^{1.1}$$

We now change the tolerance interval into a probability using the normal distribution.

As $A_1 \sim N(\mu, \sigma^2)$, we can standardise it by

$$Z = \frac{A_1 - \mu}{\sigma}$$

Where $Z \sim N(0,1)$.

Hence, the probability that a randomly selected individual lies within the age tolerance of A_0 is:

$$P_A(A_0) = \Phi\left(\frac{A_0 + t(A_0) - \mu}{\sigma}\right) - \Phi\left(\frac{A_0 - t(A_0) - \mu}{\sigma}\right)$$

Since $t(A_0) = 0.202A_0^{1.1}$:

$$\boldsymbol{P_A(A_0) = \Phi\left(\frac{A_0 + 0.202A_0^{1.1} - 39.71}{23.74}\right) - \Phi\left(\frac{A_0 - 0.202A_0^{1.1} - 39.71}{23.74}\right)}$$

Test formula with $A_0 = 16$:

$$t(16) = 0.202 \cdot 16^{1.1} = 4.265$$

$$\therefore 16 - 4.265 \leq A_1 \leq 16 + 4.265$$
$$11.735 \leq A_1 \leq 20.265$$

$$Z_1 = \frac{11.735 - 39.71}{23.74} = -1.178$$
$$Z_2 = \frac{20.265 - 39.71}{23.74} = -0.819$$

$$P_A(16) = \Phi(-0.819) - \Phi(-1.178)$$
$$= 0.0873 \text{ or } 8.73\%$$

Hence, 8.73% of the population lies within the age tolerance of a 16-year-old.

4 Gender variable

Biological gender is treated as a male/female variable. This makes it a categorical variable, falling into fixed groups rather than a scale.

Let G be gender. Based on UK data[2]:

$$P(G_1 = \text{male}) = 0.49 \quad \text{and} \quad P(G_1 = \text{female}) = 0.51$$

Hence, the probability that two people match in gender is:

$$\boldsymbol{P_G(G_0)} = \begin{cases} \mathbf{0.49} & \textbf{if } \mathbf{G_0 = male} \\ \mathbf{0.51} & \textbf{if } \mathbf{G_0 = female} \end{cases}$$

Where:

G_0 = your gender
G_1 = gender of a randomly selected individual

E.g. I am male, therefore there is a 49% chance that a randomly selected individual in the UK is also male.

5 BMI variable

The relative weight of an individual can be more meaningfully understood through Body Mass Index (BMI). For example, two people may both weigh 70kg; however, one is 1.4m tall and the other 1.9m tall, creating starkly different appearances. Hence, we group 'weight'[3] into regions! We will cover the basic mathematics behind how BMI actually works, and how it can be applied to a probability estimate. How exciting…

$$W = \text{'BMI'} = \frac{m}{H^2}$$

Where:

m is mass in kg

H is height in m

The NHS BMI classification can be shown as a piecewise function where B represents BMI *class*:

$$B = \begin{cases} 1 & \text{BMI} < 18.5 \\ 2 & 18.5 \leq \text{BMI} < 25 \\ 3 & 25 \leq \text{BMI} < 30 \\ 4 & \text{BMI} \geq 30 \end{cases}$$

Where 1 is 'underweight', 2 is 'healthy', 3 is 'overweight', 4 is 'obese'.

---

[2] https://www.ethnicity-facts-figures.service.gov.uk/uk-population-by-ethnicity/demographics/male-and-female-populations/latest/
[3] WEIGHT AND MASS AREN'T THE SAME

I averaged the total percentages for each BMI class from NHS data as shown:

Base: Adults aged 16 to 64	Underweight	Healthy weight	Overweight	Obese
16-24	11	51	20	18
25-34	1	37	35	26
35-44	1	31	37	30
45-54	1	29	37	34
55-64	1	27	37	36
65-74	1	25	40	35
75+	2	26	44	28
Overall %	2.571428571	32.28571429	35.714286	29.57142857

Hence,

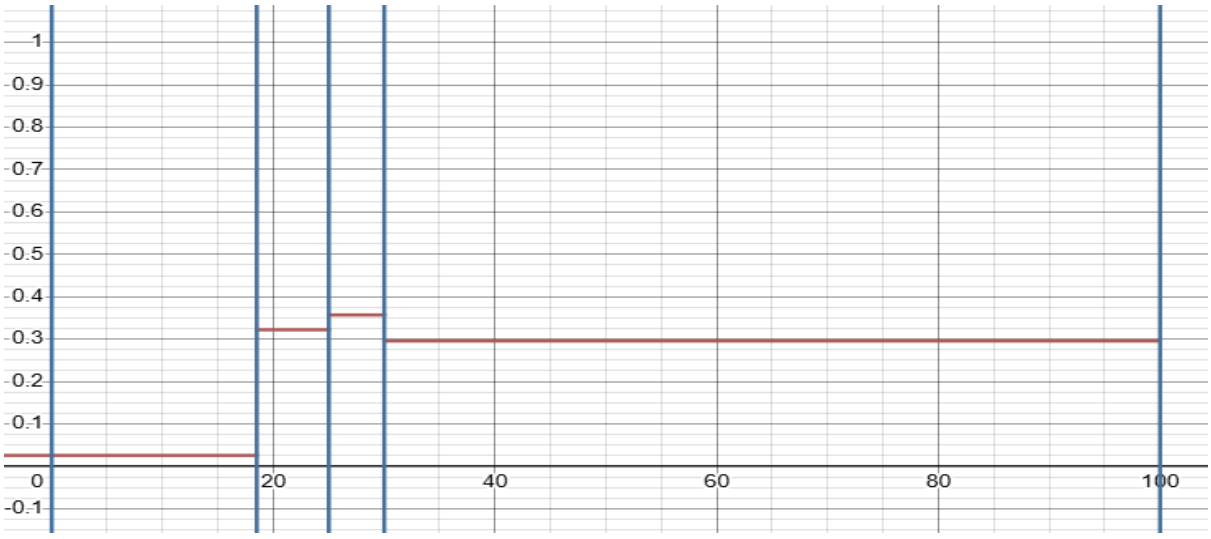
$$P(B = 1) = 0.0257$$
$$P(B = 2) = 0.3220$$
$$P(B = 3) = 0.3571$$
$$P(B = 4) = 0.2957$$

We define B_0 as the BMI class of the reference individual and B_1 as a randomly selected individual. The similarity contribution from weight is then:

$$P_B = P(B_1 = B_0)$$

The probabilities can be displayed in Desmos as the step function:

$$P(x) = \{x < 18.5: 0.0257, x < 25: 0.3220, x < 30: 0.3571, 30 < x: 0.2957\}$$



However, this probability model is weak as it implies that a BMI of 30 and a BMI of 40 have similar appearance. If we only used the 4 BMI categories, the probability function would basically be asking: 'what is the probability that another person is in the same BMI *class* as me?' That's way too ambiguous! We should first define the class intervals and use linear interpolation to estimate a normal distribution from the mean and variance.

	Class	Range	Probability p_i	Midpoint x_i
1	Underweight	$12.5 \leq x < 18.5^*$	0.0257	15
2	Healthy	$18.5 \leq x < 25$	0.3220	21.75
3	Overweight	$25 \leq x < 30$	0.3571	27.5
4	Obese	$30 \leq x < 45^*$	0.2957	37.5

*interval of 1 and 4 estimated to be halved to account for a distribution curve (not many people will a BMI of 90… so we assume its probability is negligible)

Assume BMI is distributed normally, such that:

$$X \sim N(\mu, \sigma^2)$$

$$\mu = \int x \cdot f(x)dx = \sum p_i \, x_i$$

$$\mu = (0.0257 \cdot 15) + (0.3220 \cdot 21.75) + (0.3571 \cdot 27.5) + (0.2957 \cdot 37.5)$$
$$\mu = 0.4125 + 7.0035 + 9.8203 + 11.0888$$
$$\mu = 28.33$$

$$\sigma^2 = \sum p_i x_i^2 - \mu^2$$
$$\sigma^2 = 843.22 - (28.33)^2$$

$$\sigma^2 = 40.63$$

$$\therefore X \sim N(28.33, 40.63)$$

First, I will show an unconditional model. Let X and Y be the BMIs of two independent people and $D = X - Y$

$$\therefore \sigma_D^2 = 2\sigma^2 \text{ and } \mu_D = 0$$

$$D \sim N(0, 2\sigma^2)$$

$$D \sim N(0, 81.26)$$

Let $k = 2$ be our BMI tolerance, such that:

$$P(|X - Y| \leq k) = P(|D| \leq k)$$

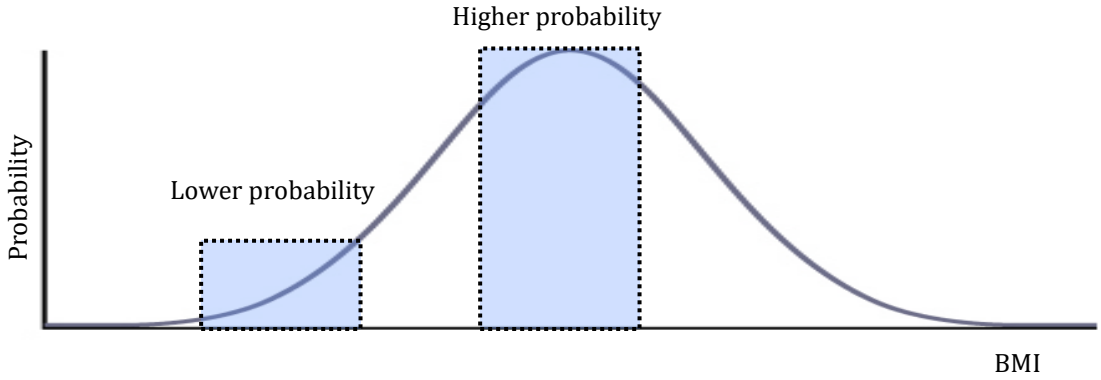
$$\therefore P(|D| \leq 2)$$

Standardise:

$$Z = \frac{D}{\sigma_D} \ , \qquad \sigma_D = \sqrt{81.26} = 9.01$$

$$P(|D| \leq 2) = P\left(-\frac{2}{9.01} \leq Z \leq \frac{2}{9.01}\right)$$

This gives $P = 0.176$, suggesting that for any BMI, there is a 17.6% chance that another randomly selected person is within ± 2 BMI. However, looking at a normal distribution curve, the probabilities *should* decrease toward the tails:



Actual model:

To refine this, we condition on a fixed individual $W_0 = w_0$, where W is the actual BMI. The probability that a randomly selected individual W_1 is a lookalike within ± 2 BMI of w_0 is:

$$P_{w_0} = P(w_0 - 2 \leq W_1 \leq w_0 + 2)$$

Since $W_1 \sim N(\mu, \sigma^2)$, it becomes:

$$P_{w_0} = \int_{w_0-2}^{w_0+2} f(t)\, dt$$

Where $f(t)$is the normal density function.

Standardising:

$$P_{w_0} = \Phi\left(\frac{w_0 + 2 - \mu}{\sigma}\right) - \Phi\left(\frac{w_0 - 2 - \mu}{\sigma}\right)$$

Since $\mu = 28.33$ and $\sigma = \sqrt{40.63} = 6.37$

$$P_{w_0} = \Phi\left(\frac{w_0 + 2 - 28.33}{6.37}\right) - \Phi\left(\frac{w_0 - 2 - 28.33}{6.37}\right)$$

Hence, our final 'weight' (BMI) variable probability is modelled by:

$$\boldsymbol{P_W(W_0) = \Phi\left(\frac{w_0 + 2 - 28.33}{6.37}\right) - \Phi\left(\frac{w_0 - 2 - 28.33}{6.37}\right)}$$

6 Ethnicity variable

This variable is really simple; all we need is fixed probabilities for different ethnic groups with data, kindly taken from ONS[4], all put into a beautiful piecewise probability function!

Let:

E_0= your ethnicity category
E_1= random UK individual

$$P_E = P(E_1 = E_0)$$

ONS data:

Ethnic category		Probability
White	e_1	0.817
Asian, Asian British, or Asian Welsh	e_2	0.093
Black, Black British, Black Welsh, Caribbean, or African	e_3	0.040
Mixed or Multiple ethnic groups	e_4	0.029
Other ethnic group	e_5	0.021

$$\boldsymbol{P_E(E_0) = \begin{cases} 0.817, & e_1 \\ 0.093, & e_2 \\ 0.040, & e_3 \\ 0.029, & e_4 \\ 0.021, & e_5 \end{cases}}$$

7 Hair colour variable

This variable is exciting. Hair colour can be modelled as a categorical variable but with an additional component of artificial colours. This introduces the idea that some variables carry more information than others (in this case, dyed hair!) For example, if you knew a friend with bright pink hair and saw a random person with it, they'd instantly become a viable lookalike!

This *concept* can be formalised using Shannon information theory[5], in which information content of an event is defined by:

$$I(x) = \log\left(\frac{1}{P(x)}\right)$$

Where x is an event.

[4]https://www.ons.gov.uk/peoplepopulationandcommunity/culturalidentity/ethnicity/bulletins/ethnicgroupenglandandwales/census2021
[5]From this *fascinating* paper: https://arxiv.org/pdf/1612.09316

For example, a natural hair colour like brown ($P \approx 0.48$) contributes relatively small information, whereas a rarer colour such as blue ($P \ll 0.05$) yields a much larger value of $I(x)$. Observing rare features reduces uncertainty.

We can model hair using a piecewise probability function.

Let:

H_0 be the observed hair colour
$q = P(D = 1)$, the probability hair is dyed

Define the observed category space:

$$H_0 \in \{\text{blonde, brown, black, red, } N_D\text{,blue, pink, green, purple, other}\}$$

Where N_D is a natural dye.

Then the probability function $P_H(H_0 = h)$is:

$$P_H(H_0) = \begin{cases} (1 - 0.5q)\,P(N = h), & h \in \{\text{blonde, brown, black, red}\} \\ 0.5q, & h = N_D \\ 0.1q, & h \in \{\text{blue, pink, green, purple, other}\} \end{cases}$$

Where:

$P(N = \text{blonde}) = 0.29, P(N = \text{brown}) = 0.48, P(N = \text{black}) = 0.18, P(N = \text{red}) = 0.05$
This formulation assumes conditional independence between D and the distribution N, such that the observed variable H_0 is determined by genetics and the decision to dye one's hair!

We *could* compute similarity using a combined function:

$$P(H_0 = H_1) = \sum_h P_H\,(h)^2$$

$$\sum_i [(1 - 0.5q)P(N = n_i)]^2$$

$$= (1 - 0.5q)^2 \sum_i P(N = n_i)^2$$

$$\sum_i P(N = n_i)^2 = 0.3494$$

$$P(H_0 = H_1) = 0.3494(1 - 0.5q)^2 + 0.30q^2$$

But that would give us an average probability only, not specific to a chosen colour. If we did use it, the term $0.3494(1 - 0.5q)^2$ represents a biological baseline where similarity is driven by genetics but is reduced as dye percentage increases. The term $0.30q^2$ represents the raw percentage of dyed hair which instead is a behavioural factor.

Although the expression $P(H_0 = H_1)$ provides a useful summary statistic, it compresses the full categorical structure of the distribution into a single scalar value. As a result, the piecewise formulation is more informative as it preserves the full state of the system of hair colour probabilities. Hence, hair colour is best modelled by the function in bold.

We will assume $q = 0.36$ based on UK estimates for dyed hair probability estimates. Hence, we get:

$$\mathbf{P_H(H_0)} = \begin{cases} \mathbf{(1-0.18)P(N = h_{natural})}, & \mathbf{h_{natural}} \in \begin{cases} \mathbf{0.29} & \mathbf{blonde} \\ \mathbf{0.48} & \mathbf{brown} \\ \mathbf{0.18} & \mathbf{black} \\ \mathbf{0.05} & \mathbf{red} \end{cases} \\ \mathbf{0.18}, & \mathbf{h = N_D} \\ \mathbf{0.036}, & \mathbf{h} \in \{\text{blue,pink,green,purple,other}\} \end{cases}$$

8 Rate of encounter

Up to here, the model has focused on the probability of two individuals sharing similar physical characteristics. However, this does not account for the number of opportunities for such comparisons to occur.

Potential similarity is only meaningful when individuals are exposed to each other, or if someone who knows another person sees someone that looks like them. Therefore, we introduce a layer to the model which defines the rates at which new people are encountered.

We can model this using a Poisson distribution. This is appropriate as encounters are independent events, occurring approximately constant over long periods of time.

Let $N =$ usable daily encounters

Such that:

$$N \sim Po(\lambda)$$

The parameter λ, number of people encountered per day, is affected by location and age. For example, a person living in central London will see more people than someone in the Outer Hebrides, and a 23-year-old student will see more people than a retired 93-year-old man.

Pedestrian flow in London is around 800/hour; with attention filtering of about 5%, $\lambda_0 \approx 40$.

Let $\lambda = \lambda_0 \cdot \lambda_L \cdot \lambda_A$, such that L and A represent location and age scaling respectively. If we define λ_L as:

$$\lambda_L = \begin{cases} 1.00 & \text{City} \\ 0.2 & \text{Urban} \\ 0.1 & \text{Suburban} \\ 0.01 & \text{Rural} \end{cases}$$

And λ_A as:

$$\lambda_A = \begin{cases} 1.0 & 16 \le A < 25 \\ 0.6 & 25 \le A < 65 \\ 0.2 & A < 16\,, A \ge 65 \end{cases}$$

Hence, expected matches $E(L)$ per year is:

$$E(L) = 365 \cdot \lambda \cdot p$$

Such that p is the product of all variables:

$$p = \prod_{X \in \{A,G,W,E,H\}} P(X_0 = X_1)$$

Where X represents the variables A, G, W, E, H.

$$\therefore \boldsymbol{E(L) = 365\lambda \prod_{X \in \{A,G,W,E,H\}} P(X_0 = X_1)}$$

9 Overall probability function

$$\mathrm{E(L)} = 365\lambda_0\lambda_\mathrm{L}\lambda_\mathrm{A} \cdot \mathrm{P_A(A_0)} \cdot \mathrm{P_G(G_0)} \cdot \mathrm{P_W(W_0)} \cdot \mathrm{P_E(E_0)} \cdot \mathrm{P_H(H_0)}$$

Assuming conditional independence, where:

$$P_A(A_0) = \Phi\left(\frac{A_0 + 0.202A_0^{1.1} - 39.71}{23.74}\right) - \Phi\left(\frac{A_0 - 0.202A_0^{1.1} - 39.71}{23.74}\right)$$

$$\mathrm{P_G(G_0)} = \begin{cases} 0.49 & \mathrm{G_0 = male} \\ 0.51 & \mathrm{G_0 = female} \end{cases}$$

$$P_W(W_0) = \Phi\left(\frac{w_0 + 2 - 28.33}{6.37}\right) - \Phi\left(\frac{w_0 - 2 - 28.33}{6.37}\right)$$

$$\mathrm{P_E(E_0)} = \begin{cases} 0.817 & \mathrm{e_1} \\ 0.093 & \mathrm{e_2} \\ 0.040 & \mathrm{e_3} \\ 0.029 & \mathrm{e_4} \\ 0.021 & \mathrm{e_5} \end{cases}$$

$$\mathrm{P_H(H_0)} = \begin{cases} (1 - 0.18)\mathrm{P(N = h_{natural})}, & \mathrm{h_{natural}} \in \begin{cases} 0.29 & \text{blonde} \\ 0.48 & \text{brown} \\ 0.18 & \text{black} \\ 0.05 & \text{red} \end{cases} \\ 0.18, & \mathrm{h = N_D} \\ 0.036, & \mathrm{h} \in \{\text{blue,pink,green,purple,other}\} \end{cases}$$

$$\lambda_0 = 40, \lambda_L = \begin{cases} 1.00 & \text{City} \\ 0.50 & \text{Urban} \\ 0.10 & \text{Suburban} \\ 0.05 & \text{Rural} \end{cases}, \lambda_A = \begin{cases} 1.0 & 16 \le A < 25 \\ 0.6 & 25 \le A < 65 \\ 0.2 & A < 16\,, A \ge 65 \end{cases}$$

```python
import math


def standard_normal_cdf(z):  4 usages
    """Approximate the standard normal CDF using the error function."""
    return 0.5 * (1 + math.erf(z / math.sqrt(2)))


def get_age_probability(age):  1 usage
    """Calculate P_A(A_0) - probability someone is within age tolerance."""
    mu = 39.71
    sigma = 23.74
    tolerance = 0.202 * (age ** 1.1)

    z_upper = (age + tolerance - mu) / sigma
    z_lower = (age - tolerance - mu) / sigma

    return standard_normal_cdf(z_upper) - standard_normal_cdf(z_lower)


def get_gender_probability(gender):  1 usage
    """Calculate P_G(G_0) - probability of matching gender."""
    return 0.49 if gender.lower() == 'male' else 0.51


def get_bmi_probability(bmi):  1 usage
    """Calculate P_W(W_0) - probability someone is within BMI tolerance."""
    mu = 28.33
    sigma = 6.37
    tolerance = 2

    z_upper = (bmi + tolerance - mu) / sigma
    z_lower = (bmi - tolerance - mu) / sigma

    return standard_normal_cdf(z_upper) - standard_normal_cdf(z_lower)


def get_ethnicity_probability(ethnicity):  1 usage
    """Calculate P_E(E_0) - probability of matching ethnicity."""
    probabilities = {
        'white': 0.817,
        'asian': 0.093,
        'black': 0.040,
        'mixed': 0.029,
        'other': 0.021
    }
    return probabilities.get(ethnicity.lower(), default: 0.021)


def get_hair_probability(hair_colour):  1 usage
    """Calculate P_H(H_0) - probability of matching hair colour."""
    natural_probs = {
        'blonde': 0.29,
        'brown': 0.48,
        'black': 0.18,
        'red': 0.05
    }

    if hair_colour.lower() in natural_probs:
        return 0.82 * natural_probs[hair_colour.lower()]
    elif hair_colour.lower() == 'natural dye':
        return 0.18
    else:  # Unnatural colours (blue, pink, green, purple, etc.)
        return 0.036


def get_location_multiplier(location):  1 usage
    """Get λ_L based on location type."""
    multipliers = {
        'city': 1,
        'urban': 0.2,
        'suburban': 0.1,
        'rural': 0.01
    }
    return multipliers.get(location.lower(), default: 0.05)


def get_age_activity_multiplier(age):  1 usage
    """Get λ_A based on age group."""
```

```
    if 16 <= age < 25:
        return 1.0
    elif 25 <= age < 65:
        return 0.6
    else:
        return 0.2


def calculate_bmi(weight_kg, height_m):  1 usage
    """Calculate BMI from weight and height."""
    return weight_kg / (height_m ** 2)


def calculate_lookalikes():  1 usage
    """Main function to gather inputs and calculate expected lookalikes per year."""
    print("=" * 60)
    print("  LOOKALIKE PROBABILITY CALCULATOR")
    print("  Based on UK population statistics")
    print("=" * 60)
    print()

    # Gather inputs
    age = int(input("Enter your age: "))
    gender = input("Enter your gender (male/female): ")

    print("\nFor BMI calculation:")
    weight = float(input("Enter your weight in kg: "))
    height = float(input("Enter your height in metres (e.g., 1.75): "))
    bmi = calculate_bmi(weight, height)
    print(f"Your BMI: {bmi:.1f}")

    print("\nEthnicity options: white, asian, black, mixed, other")
    ethnicity = input("Enter your ethnicity: ")

    print("\nHair colour options:")
    print("  Natural: blonde, brown, black, red")
    print("  Dyed natural: 'natural dye'")
    print("  Unnatural: blue, pink, green, purple, other")
    hair = input("Enter your hair colour: ")

    print("\nLocation options: city, urban, suburban, rural")
    location = input("Where do you primarily spend time? ")

    # Calculate individual probabilities
    p_age = get_age_probability(age)
    p_gender = get_gender_probability(gender)
    p_bmi = get_bmi_probability(bmi)
    p_ethnicity = get_ethnicity_probability(ethnicity)
    p_hair = get_hair_probability(hair)

    # Calculate encounter rate
    lambda_0 = 40  # Base encounters per day
    lambda_L = get_location_multiplier(location)
    lambda_A = get_age_activity_multiplier(age)
    daily_encounters = lambda_0 * lambda_L * lambda_A

    # Combined probability
    p_total = p_age * p_gender * p_bmi * p_ethnicity * p_hair

    # Expected lookalikes per year
    expected_per_year = 365 * daily_encounters * p_total

    # Display results
    print("\n" + "=" * 60)
    print("  RESULTS")
    print("=" * 60)
    print("\nIndividual Probabilities:")
    print(f"  Age match:       {p_age:.4f} ({p_age * 100:.2f}%)")
    print(f"  Gender match:    {p_gender:.4f} ({p_gender * 100:.2f}%)")
    print(f"  BMI match:       {p_bmi:.4f} ({p_bmi * 100:.2f}%)")
    print(f"  Ethnicity match: {p_ethnicity:.4f} ({p_ethnicity * 100:.2f}%)")
    print(f"  Hair match:      {p_hair:.4f} ({p_hair * 100:.2f}%)")
    print(f"\nCombined probability: {p_total:.6f} ({p_total * 100:.4f}%)")
    print(f"  (1 in {1 / p_total:,.0f} people)")

    print("\nEncounter Rate:")
    print(f"  Base daily encounters (λ₀): {lambda_0}")
    print(f"  Location multiplier (λ_L):  {lambda_L}")
```

```
    print(f"  Age activity multiplier (λ_A): {lambda_A}")
    print(f"  Effective daily encounters: {daily_encounters:.0f}")
    print(f"  Annual encounters: {365 * daily_encounters:,.0f}")

    print("\n" + "-" * 60)
    print(f"  EXPECTED LOOKALIKES PER YEAR: {expected_per_year:.2f}")
    print("-" * 60)

    if expected_per_year < 1:
        days_per_lookalike = 365 / expected_per_year
        print(f"  On average, you'd encounter a lookalike every {days_per_lookalike:.0f} days")
    else:
        print(f"  On average, you'd encounter ~{expected_per_year:.0f} lookalikes per year")

    return expected_per_year


if __name__ == "__main__":
    calculate_lookalikes()
```

The above is a Python script which executes all of the functions made in this paper. It can be found online here too: https://github.com/JackWM-Code/UK-Lookalike-Probability-Function

10 Conclusion and criticisms

This essay provides a structured estimate of lookalike encounters but relies on simplified assumptions that limit its precision. The assumption of conditional independence between variables makes some probabilities inflate, as traits like ethnicity are hair colour are, in reality, correlated. Additionally, tolerance functions are subjective, introducing bias into 'similarity' thresholds. The Poisson model also assumes constant rates and uniform attention span, oversimplifying human perception and social behaviour.

The function gives the number of *potential* lookalikes. It may be useful to further refine this by estimating the proportion of these potential matches that are genuinely perceived as lookalikes and incorporating this as a fixed probability.

Despite these limitations, the model offers insight into how complex appearance can be approximated probabilistically, demonstrating how mathematical modelling can (attempt to) answer any of our bus-ride questions!