Statistical Methods in AI

<u>PizzaPizzaPizza Report</u>

Model checking

This section basically deals with 2 problems:

- Are the model and parameters estimated a good fit for the underlying data?
- Given two separate models, which is a better fit for the underlying data?

1) Method I: Posterior Predictive Check

Once a posterior distribution for a particular model is obtained, it can be used to simulate new data conditional on this distribution that might be helpful to assess whether the model provides valid predictions so that these can be used for extrapolating to future events.

The posterior predictive check requires one to generate new data from the predicted model. It means that we have estimated 200,000 credible values of µ for the Poisson distribution. That means we can construct 200,000 Poisson distributions with these values and then randomly sample from these distributions. This is represented as:

$$p(y' \mid y) = \int p(y' \mid \theta) \cdot f(\theta \mid y) d\theta$$

Conceptually, if the model is a good fit for the underlying data, then the generated data should resemble the original observed data.

With the help of PyMC we fitted our model-:

```python
import pymc3 as pm
y_pred = pm.Poisson('y_pred', mu=mu)
```

`y_pred` is identical to `y_est` but instead we do not specify the observed data.

Plotting `y_pred` and compare it to the observed data `y_est:`
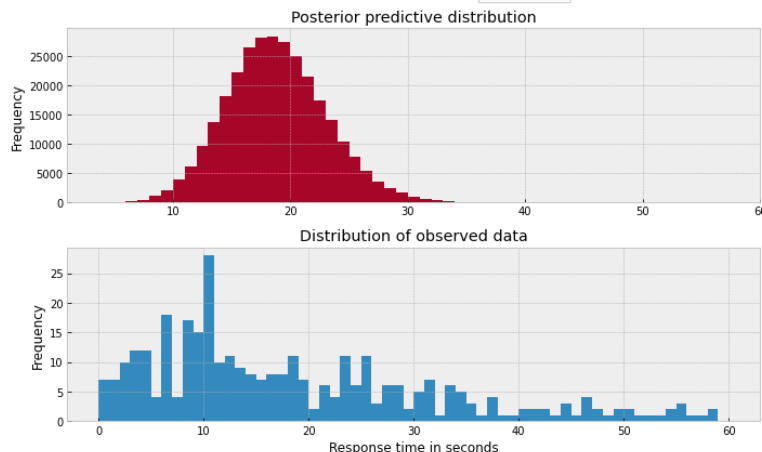
1.1) Choosing the most suitable distribution

In the above figure, the 2 plots are not similar. In an ideal case, we want the posterior predictive distribution to somewhat resemble the distribution of the observed data. We can say that if we have correctly estimated the parameters of the model, then we should be able

to sample similar data from that model. This is not the case. Poisson distribution is not suitable for this data.

After testing on different distributions we got to Negative Binomial distribution. It has 2 parameters - μ & α.
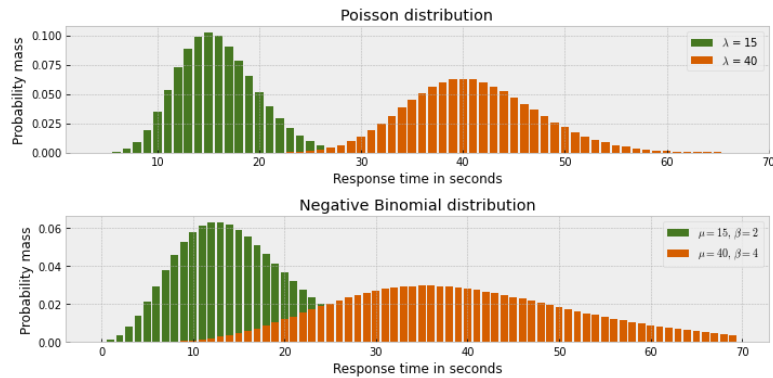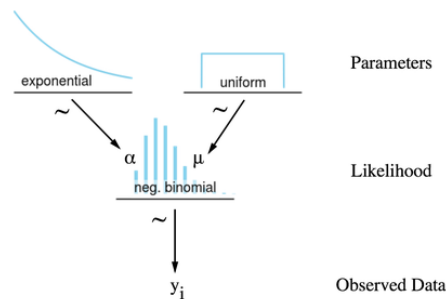

Now estimating the parameters for a Negative Binomial distribution given the same dataset used before. We will use a Uniform distribution to estimate both μ and α.
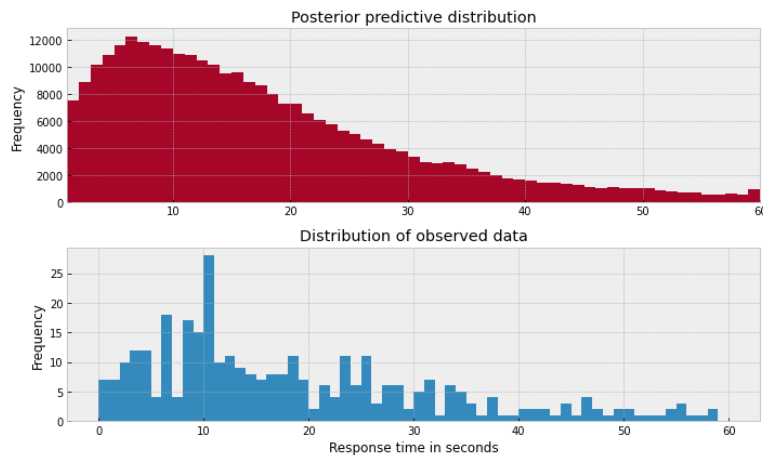
The model can be showed as:

$$y_j \sim \text{NegBinomial}(\mu, \alpha)$$
$$\alpha = \text{Exponential}(0.2)$$
$$\mu = \text{Uniform}(0, 100)$$

Look at the posterior preditive distribution and see if it more closely resembles the distribution from the observed data:

These 2 distributions look a lot similar to each other. Therefore we can say that Negative binomial model is a more appropriate fit for the this data.

2) Method II: Bayes Factor

It is used for calculating the evidence for two competing hypothesis. It is the ratio of posterior odds to the prior odds for two hypothesis and it is also equal to the ratio of likelihoods for those hypothesis. The bayes factor in favour of H_{10} is given by:

$$BF_{10} = \frac{p(H_1|y)/p(H_0|y)}{p(H1)/p(H_0)}$$

where the prior probabilities are p(H_0) and p(H_1) = 1-p(H_0).Thus, the Bayes factor can equivalently be written as the ratio of the marginal likelihoods of the observed data under the two hypotheses:

$$BF_{10} = \frac{p(y|H_1)}{p(y|H_0)}$$

2.0.1) In this case the two competing hypothesis are :

- The probability of the distribution being a negative binomial distribution
- The probability of the distribution being a poisson distribution

```
# Compute the Bayes factor
prob_pois = trace[burnin:]['tau'].mean()
prob_nb = 1 - prob_pois
BF = (prob_nb/prob_pois)*(prior_model_prob/(1-prior_model_prob))
print("Bayes Factor: %s" % BF)
```
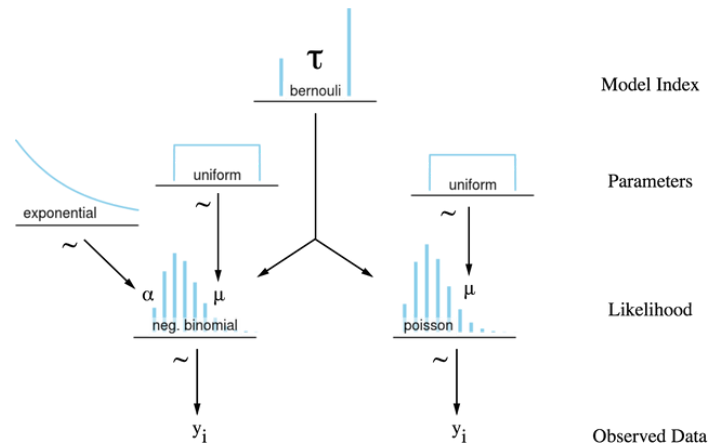

Bayes Factor Value	Interpretation
$\text{BF}(M_1, M_2) < \frac{1}{10}$	Strong evidence for M_2
$\frac{1}{10} < \text{BF}(M_1, M_2) < \frac{1}{3}$	Moderate evidence for M_2
$\frac{1}{3} < \text{BF}(M_1, M_2) < 1$	Weak evidence for M_2
$1 < \text{BF}(M_1, M_2) < 3$	Weak evidence for M_1
$3 < \text{BF}(M_1, M_2) < 10$	Moderate evidence for M_1
$\text{BF}(M_1, M_2) > 10$	Strong evidence for M_1

The value that we obtain for Bayes factor is 1.89 which is greater than 1 which means that the M1 is more likely and thus M1 which in our case was that negative binomial distribution fits better with our original data is more likely.