sSs Machine Learning Model

Contents

4.1 Introduction: 2

4.2 Problem Statement: 2

4.3 Mathematical Modeling and Notations: 3

4.4 First Solution Idea, Anomaly Detection: 7

4.5 Second Solution Idea, Data Generation: 9

4.6 Choosing a PDF: 15

4.6.1 Discrete PDF: 15

4.6.2 Normal Distribution Function: 19

4.6.3 Multimodal Normal Distribution Function 22

4.7 Testing and Results 31

4.8 Case-study, Kafr El-sheikh University 41

4.9 Summary 45

4.1 Introduction:

In this chapter, we go deep into the problem, come up with ideas to solve it, evaluate these ideas and choose a proper one. We start by stating the problem, and then define mathematical notations to deal with it abstractly. Afterwards, we propose a model that uses a Machine Learning technique called Anomaly Detection to solve the problem partially. Then, we derive a better model that generates data using Probability Distribution Functions (PDFs) to solve the problem entirely. The last thing we do in the chapter is to choose a PDF that suits the problem. We go through the discrete PDF, normal PDF and the multimodal normal PDF.

4.2 Problem Statement:

Many organizations try to collect data from their people to gain insights that support their decision making process. There is a plenty of effective techniques that are used to achieve that task of data collection. The survey-based technique is so common that much research was directed towards the creation of effective surveys. However, the volume of data that organizations need to collect (and can collect) is increasing. However, if you want more data out of a survey, you have to increase the length of the survey, and if you do this, the response rate decreases as well as the data-quality. This problem is what our system is supposed to solve. *The system is supposed to enable organizations to collect much quality-data from people without boring people with long and time-consuming surveys.*

4.3 Mathematical Modeling and Notations:

The first step of solving any problem (especially if it is to be solved with computers) is to mathematically model the problem. This moves the problem from the fuzzy real world domain into the powerful abstract mathematical domain, which gives us endless tools to solve the problem.

4.3.1 Representing Responses:

We want to represent the response to a survey in a mathematical form. A survey is a group of questions. However, questions have many types. There are:

- Behavior or fact questions.
- Knowledge questions.
- Psychological state or attitude questions.

In our problem, we stick to developing a solution for attitude questions that are measured using rating scales (e.g. on a scale from 1 to 5, how much do you like Pizza?). This makes it possible to represent the response to an entire survey with n questions as follows:

$$r = \begin{pmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_n \end{pmatrix} \tag{1}$$

where ρ_i is the value of the response to the i^{th} question in the survey. If we have m responses to the survey, we can stack them horizontally to have the following matrix:

$$R = \begin{pmatrix} \rho_1^1 & \cdots & \rho_1^m \\ \vdots & \ddots & \vdots \\ \rho_n^1 & \cdots & \rho_n^m \end{pmatrix} \tag{2}$$

where ρ_i^j is the j^{th} response to the i^{th} question.

Example 1:

If we have a survey that measures on a scale from 1 to 5 how much people love Pizza and how much people love Pasta. Someone may respond 5 for Pizza and 3 for Pasta. Someone else may respond 2 for Pizza and 4 for Pasta. Represent these responses in the matrix form.

Solution:

These responses can be represented as follows:

$$R = \begin{pmatrix} 5 & 2 \\ 3 & 4 \end{pmatrix}$$

where the first column represents the response of the first responder, and the second column represents the response of the second. The first row represents all the responses to the Pizza question, and the second row represents all the responses to the Pasta question.

4.3.2 Representing Answers and Answer Counts:

We expect the answers of any questions in the survey to be limited and defined. The vector that contains the allowed answers is defined as:

$$a_i = \begin{pmatrix} \alpha_i^1 \\ \alpha_i^2 \\ \vdots \\ \alpha_i^l \end{pmatrix} \quad (3)$$

where α_i^j is the j^{th} answer to the i^{th} question of the survey.

Histograms are useful to visualize and summarize responses (Fig. 4.1). They can be used to visually see how many times each answer has been chosen in a sample.

We can represent this histogram data in a vector that is defined as follows:

$$h_i = \begin{pmatrix} \eta_i^1 \\ \eta_i^2 \\ \vdots \\ \eta_i^l \end{pmatrix} \quad (4)$$

where η_i^j is the appearance count of the j^{th} answer to the i^{th} question in the survey responses.

Example 2:

If a question is answered on a scale from 1 to 5, and 500 respondents has answered the question. The results were:

- 154 responder answered 1.
- 57 responder answered 2.
- 34 responder answered 3.
- 46 responder answered 4.
- 209 responder answered 5.

Find the answer vector, find their histogram vector, and plot a histogram for the response data.

Solution:

- The answer vector is

$$a = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}$$

- The histogram vector is

$$h = \begin{pmatrix} 154 \\ 57 \\ 34 \\ 46 \\ 209 \end{pmatrix}$$

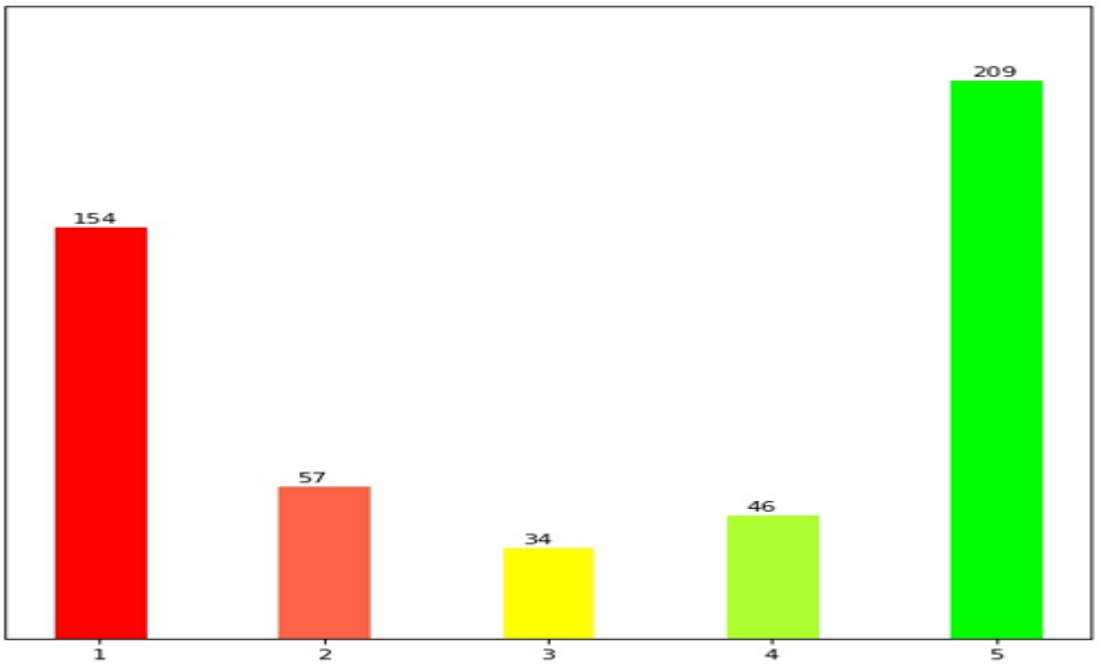

Figure 1: A histogram that shows that 154 responders gave a rating of 1, 57 responders gave a rating of 2, 34 responders gave a rating of 3, 46 responders gave a rating of 4 and 209 responders gave a rating of 5.

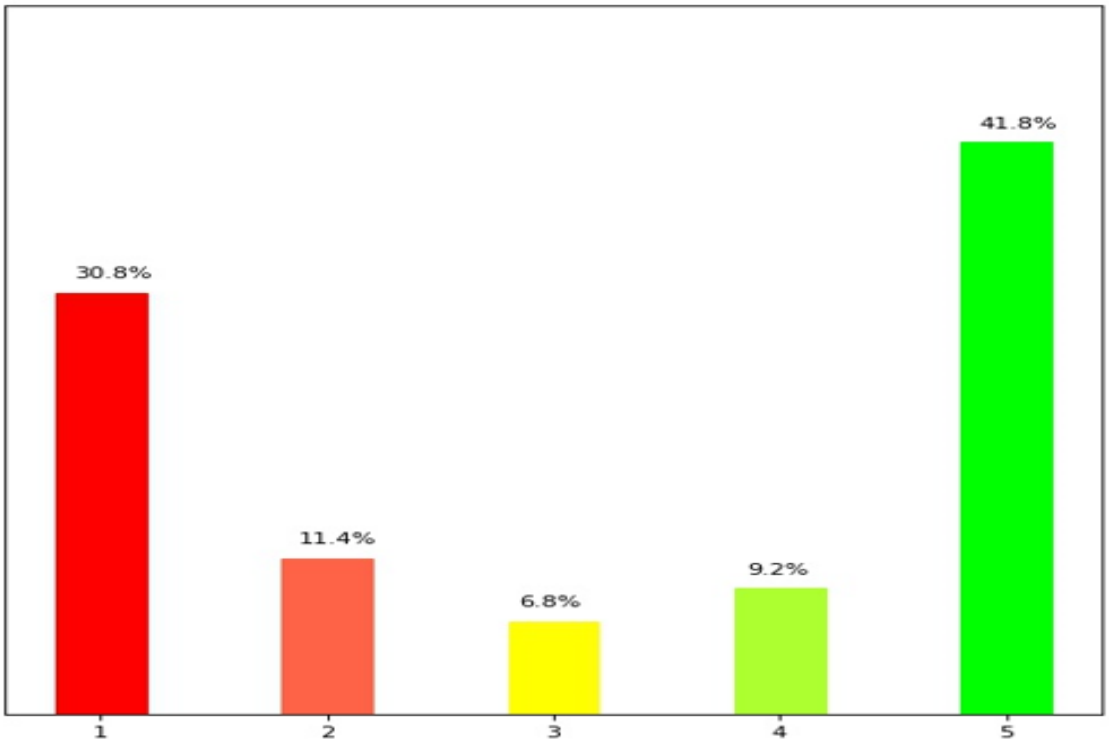

Figure 2: A histogram that shows percentages instead of counts it indicates that 30.8% of the responders gave a rating of 1, 11.4% of the responders gave a rating of 2, 6.8% of the responders gave a rating of 3, 9.2% of the responders gave a rating of 4 and 41.8% of the responders gave a rating of 5.

4.4 First Solution Idea, Anomaly Detection:

After stating the problem and defining the mathematical notations to deal with it, in this section, we go through a thought process that aims to find a proper solution to the problem. Before we begin, let us restate the problem.

> *The system is supposed to enable organizations to collect much quality-data from people without boring people with long and time-consuming surveys.*

In this statement, we can divide the problem into 2 parts:

1- *Enable organizations to collect much quality-data from people.*
2- Do not bore people with long and time-consuming surveys.

Let us focus on **the first part of the problem**. Organizations can always throw long surveys at people to collect much data. If the response rate decreases, they can force people to take their surveys. For example, a college could force students to take a survey before they can access their final grades. However, there is little organizations can do to ensure that the data they collect is of quality. **If you force people to take a long survey, they will probably fabricate their answers for the sake of getting to the end of the survey.**

If there exists a method to measure how reliable a response is, organizations could use it to ensure that their data is of quality. Actually, we could utilize a Machine Learning (ML) technique called **Anomaly Detection** to do this.

Anomaly Detection, also called outlier detection, is the identification of rare items, events or observations that raise suspicions by differing significantly from the majority of data. There are broad categories of anomaly detection techniques. One category that we can use is the unsupervised Anomaly Detection technique, which detects anomalies in an unlabeled test dataset under the assumption that the majority of the instances in the dataset are normal. In other words, we look for instances that seem to fit least to the remainder of the data set.

Hence, using the aforementioned technique, we could detect anomalous (fabricated) responses assuming that the majority of the responses are normal (not fabricated). One algorithm to apply this technique is to build a probability distribution that maps each response to a probability density value using a probability distribution function (PDF). Hence, we will have a model that utilizes a PDF for each question to measure the reliability of the response to this question. Appendix A provides background on PDFs.

This algorithm could be utilized in 2 different ways:

1- At the data aggregation process, assuming that the majority of responses are not fabricated, the model could find fabricated responses and drop them.

2- Assuming that the majority of the previous responses are not fabricated, the model could find whether the current response is fabricated, and prevent the responder from submitting a fabricated response.

If it is the case that the assumptions come true, both solutions look acceptable. However, in most cases, these assumptions don’t come true.

Moreover, the first solution may result in dropping many responses, and hence, contradict the requirement of acquiring much quality-data (we will have quality-data but not much).

As for the second solution, it may lead to the same result, if people can't submit their responses, they will abandon the survey, which decreases the response rate. On the other hand, if you force people to take the survey and prevent them from submitting fabricated answer, this worsens **the second part of the problem** at which we don't want to bore people with long and time-consuming surveys.

4.5 Second Solution Idea, Data Generation:

In probability theory, there are two interpretations for the concept of probability:

1- **Bayesian interpretation:** probability is interpreted as reasonable expectation or as a quantification of degree of belief.
2- **Frequentist interpretation:** probability is interpreted as the frequency of an outcome if the experiment is repeated a large number of times.

In the anomaly detection technique, we find the probability of a given response using a PDF. This probability can be interpreted, using the Bayesian interpretation, as how much we believe that response.

On the other hand, we could use the Frequentist interpretation to derive a new solution idea. Instead of using a PDF to measure the probability of a response, we could use a PDF to generate responses. An answer will be generated at a frequency proportional to its probability. For example, if we have a question to be answered on a scale from 1 to 5, and the probability that someone responds 5 is 75%, then, if we generate a hundred responses, about 75 of the generated responses will have the answer 5.

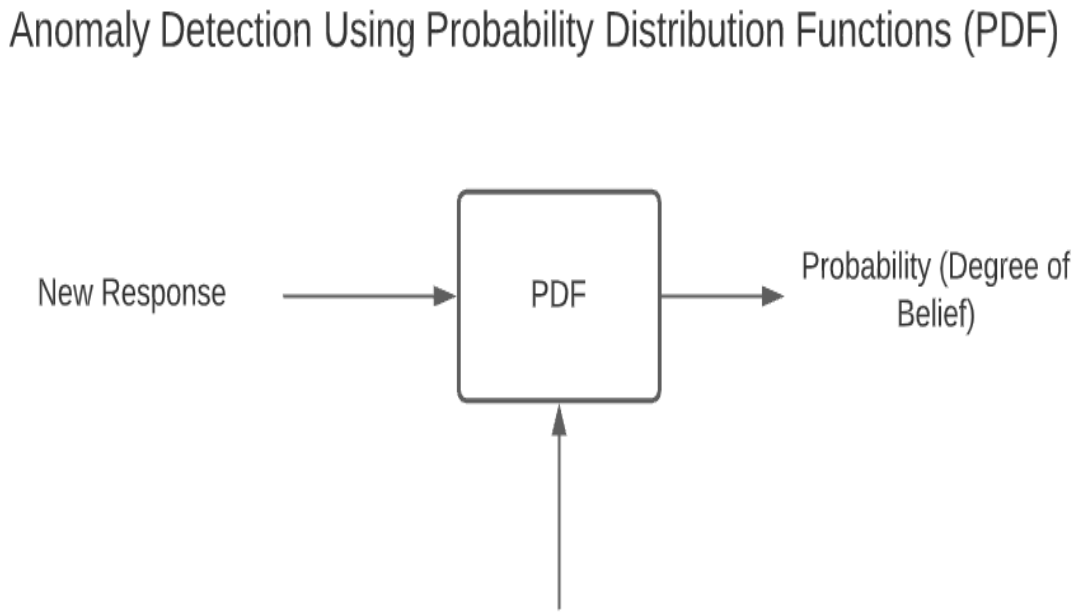

Figure 3: Anomaly Detection using Probability Distribution Functions (PDF).

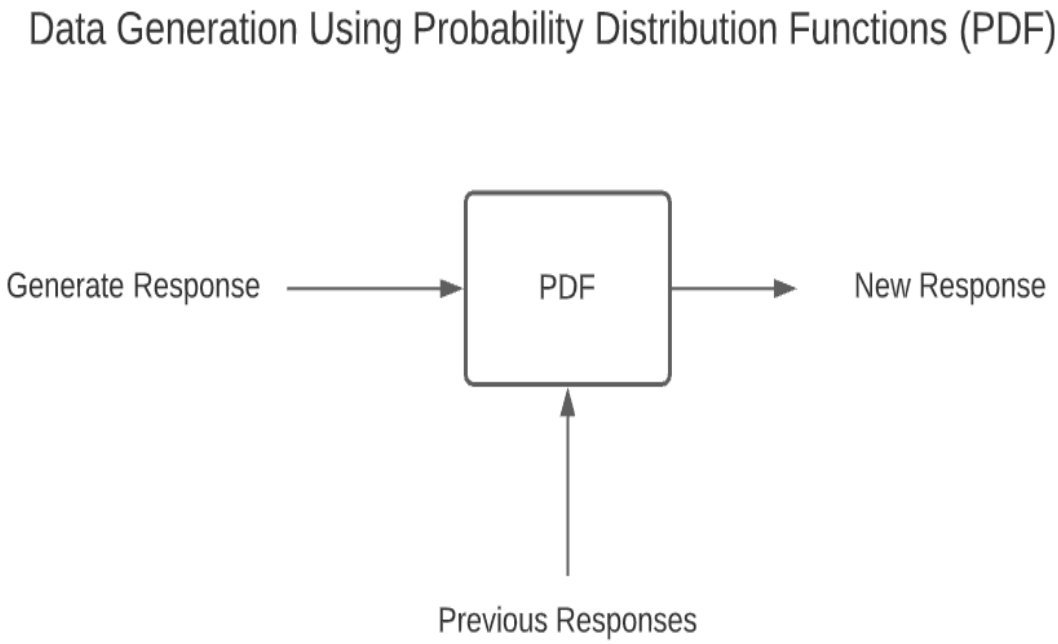

Figure 4: Data Generation Using Probability Distribution Functions (PDF).

This idea can be utilized as follows:

- After a number of respondents (first respondents) have submitted their responses, assuming that the responses are not fabricated, build a PDF using these responses.
- Use the PDF to generate an automatic response to each question for the current respondent.
- Show this respondent the generated responses and allow him to change them as he likes.
- After the respondent have submitted the response, the PDF is updated using it.

If it is the case that the assumptions come true, this idea could be what we are looking for. The idea only depends on one assumption, that is, make sure that the first people to take the survey don't fabricate their responses and leave the rest to the system. It is a middle ground that almost solves both parts of the problem. As for the first part of the problem, organizations could conduct long surveys and only worry about the veracity of the first responses because these responses propagate to the following responses. As for the second part of the problem, respondents can take the long surveys and choose which questions to answer and which questions to ignore. If they ever get bored in the middle, they can submit the survey without worrying about the rest of the questions because they are already answered. If the data quality from first respondents is high, it will almost remain high along all the following responses.

One hyper-parameter for which we need to pick a value is the number of first respondents. In fact, it should be proportional to the number of expected respondents.

Hence,

$$F \propto E$$

$$F = C * E \tag{5}$$

where F is the number of first respondents, E is the number of expected respondents and C is a constant. Logically, $F \leq E$, and hence, $C \in [0,1]$. We can interpret C as the fraction of expected respondents to be considered first respondents. If $C = 0$, there will be no first respondents. If $C = 1$, all of the expected respondents will be considered first respondents. The recommended value for this constant is 0.2. This value is derived from the 20/80 rule, which states that 20% of people control the opinion of the remaining 80%.

One crucial part in the model is the PDF. Bad PDF won't conserve the high data quality and will cause it to deteriorate. So, in the coming sections, we will discuss different PDFs to try and decide the one that copes with our problem nature.

Another crucial part of the model is the human correction. We do not want the responses of the first respondents to entirely propagate through the latent responses and become the final result with no effect from the latent respondents. In fact, they have to correct the responses according to their opinions so that the final result properly expresses both groups. In our model, we will go with the assumption that if a respondent submits a response, then, this response expresses his opinion entirely. In other words, if the model generates a response for a responder and he submits it even without reading, we assume that he agrees with all the answers in the response.

4.5 Data Generation Model Notations:

In this section, we define some notations that we will use along the coming sections.

When testing, we will get the full survey data of a test survey, take the responses of the first respondents, feed them into our model to generate data that, hopefully, assembles the rest of the responses of the full data. We need to give each of these sets a name for reference. We will refer to the full survey data as full data (F), the responses of the first respondents as input data (I), the generated responses as generated data (G) and the data after combining the input data and the generated as the expected data (E).

Symbol	Meaning	Explanation
F	full data	The Full survey data.
I	input data	The responses of the first respondents. They are fed as Input for the model.
G	generated data	The model Generated responses.
E	expected data	The Expected full survey data that comes from combining the input data and the generated data.

Table 1

To measure the accuracy of our model, we conduct the following procedure.

- Get the full data (F) of the test survey and extract the input data (I).
- Feed the model with the input data (I) to get the generated data (G), also, correct a percentage of the generated data, simulating human correction.
- Combine (I) and (G) to get the expected data (E).
- Get the histogram vectors for the full data (h_f) and the histogram vectors (h_e) for the expected data.

- Use the following equation to measure the accuracy of the model:

$$Accuracy = 1 - \frac{distance(h\_f_i,\ h\_e_i)}{\max\_distance(h\_f_i,\ h\_e_i)} \quad (6)$$

where $distance(h\_f_i,\ h\_e_i)$ is the distance between h_f_i & h_e_i. Any distance measure would do, however, we use the Euclidean distance. $\max\_distance(h\_f_i,\ h\_e_i)$ is the maximum possible distance between the 2 vectors. The maximum distance happens when one answer is chosen by 100% of the respondents in the full data, and another answer is chosen by 100% in the expected data. If we normalize the histogram data, then the maximum distance is $\sqrt{2}$. We could also express the accuracy in percentages by multiplying it by 100. The equation of the accuracy becomes:

$$Accuracy = \left(1 - \sqrt{\frac{\sum_{j=1}^{m}\left(\eta\_f_i^j - \eta\_e_i^j\right)}{2}}\right) * 100 \quad (7)$$

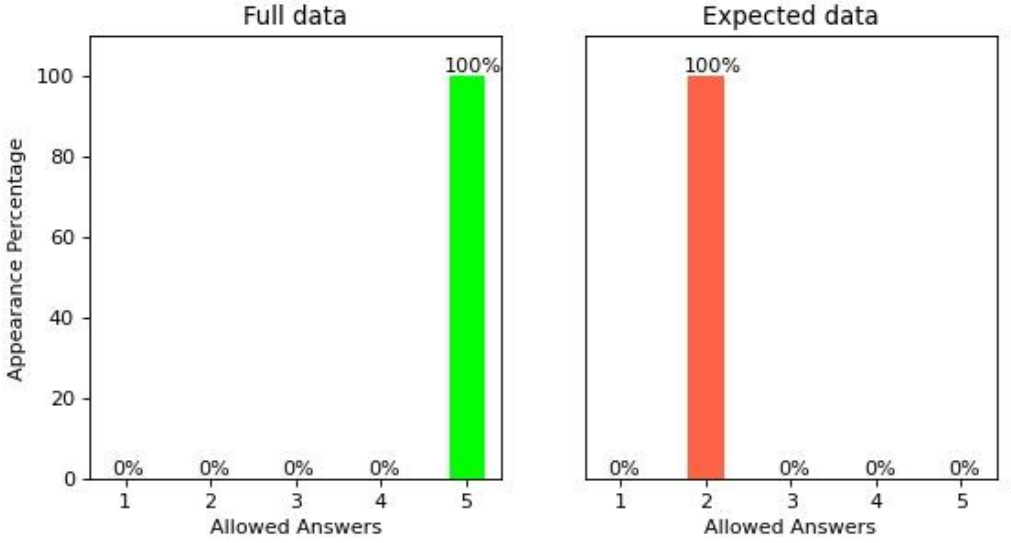

Figure 5: The case at which the maximum distance is reached.

4.6 Choosing a PDF:

We can use any PDF as the core of our model. However, we want a PDF that

1- Conserves the data distribution of the input data.
2- Understands the relation between answers and extrapolates their counts.

The first specification means that if the frequency of an answer in the input data is 50%, we want its frequency to be almost the same in the generated data. However, the second specification means that, in the input data, if the frequency of an answer is high, and the frequency of another answer related to the first one is low, then the frequency of the second one should increase in the generated data, and its count should be extrapolated. It is obvious that the two specifications somehow work against each other. The first tries to conserve the data distribution and the other tries to alter it. So, we will have to make a trade-off between the two specifications.

We will get the appropriate PDF that satisfies both specifications by finding a PDF that only satisfies the first specification and finding a PDF that only satisfies the second specification. Afterwards, we will find the PDF that satisfies the two specifications.

4.6.1 Discrete PDF:

The response values are assumed to be discrete, that is, if we have a question that is answered on a scale of 1 to 5, only (1, 2, 3, 4, 5) are allowed. Some questions allow values in between, however, these values are always limited and can be dealt with as discrete values. This discrete nature makes it possible to use the simplest form of PDFs which is the discrete PDF.

The following is the definition of the PDF:

$$p(\alpha_i^j) = \frac{\eta_i^j}{\sum_{k=1}^{m} \eta_i^j} \tag{8}$$

where $p(\alpha_i^k)$ is the probability of the k^{th} answer to the i^{th} question.

Example 3:

If the allowed answers for a question are

$$a = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}$$

And, the histogram vector is

$$h = \begin{pmatrix} 154 \\ 57 \\ 34 \\ 46 \\ 209 \end{pmatrix}$$

What is the probability of generating a response that answers 5?

Solution:

The probability of generating a response with the answer 5 is

$$p(5) = \frac{209}{154 + 57 + 34 + 46 + 209} = \frac{209}{500} = 0.418$$

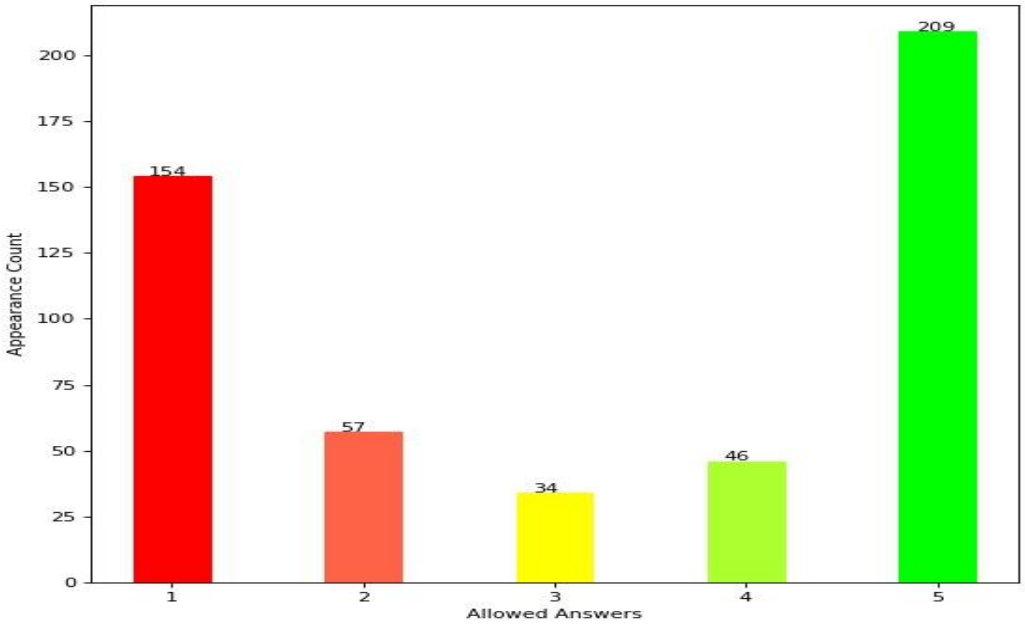

Figure 6: The histogram of the question.

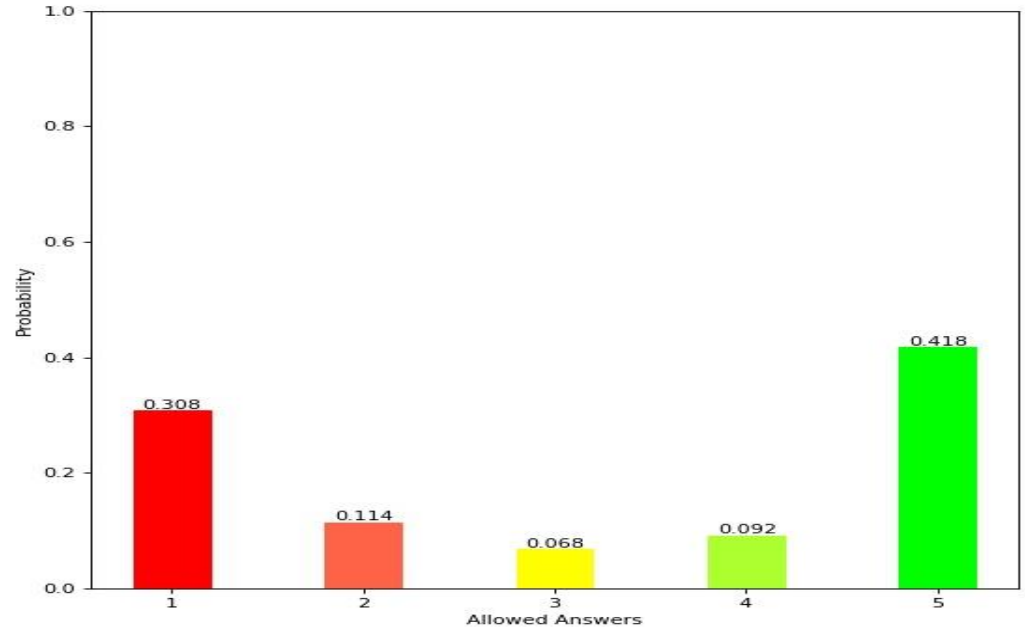

Figure 7: The discrete PDF of the question.

Returning to our specifications, this PDF clearly conserves the data distribution in the input data and generates answers with almost the same frequency as the input data. We performed a test with a sample of 500 responses using this PDF. We made sure that the distribution of the input data was similar to that of the full data. This resulted in expected data with an accuracy of 96.97%. The results are shown in (figure 8).

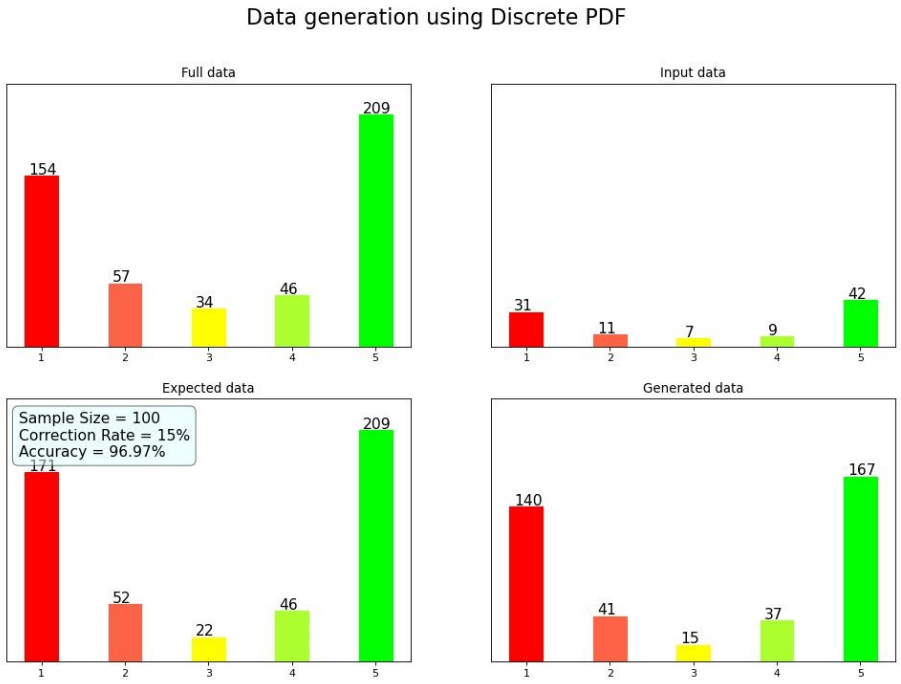

Figure 8: The results of testing data generation using discrete PDF. The input data has a data distribution similar to that of the full data, which results in high accuracy expected data.

However, if the distribution of the input data differs from the full data (misleading input data), the accuracy will decrease. Moreover, if an answer was never selected in the input data, it will never be selected in the generated data because its probability would be zero. (Figure 9) shows a test that has these two assumptions applied to the input data. This resulted in an accuracy of 79.56%.

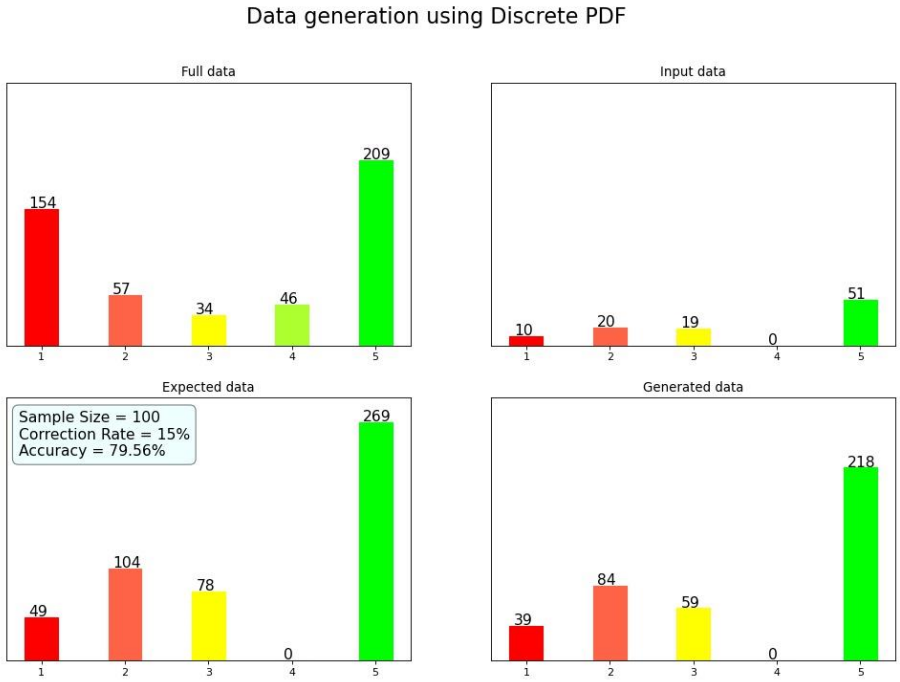

Figure 9: The results of testing data generation using discrete PDF. The input data has a data distribution different from that of the full data, which results in low accuracy expected data.

4.6.2 Normal Distribution Function:

As for the second specification, we need to understand the relation between answers. In deed, in rating questions, nearby answers are related. If something is rated 4, there is a good chance that someone would rate it 3 and someone else would rate it 5. So, if the probability of an answer increases, our PDF should also increase the probability of its nearby answers.

Almost the most famous PDF is the Normal Distribution Function. It allows us to concentrate the probability density at one point about which the density decreases exponentially. (Figure 10) shows the graph of a general normal distribution function.

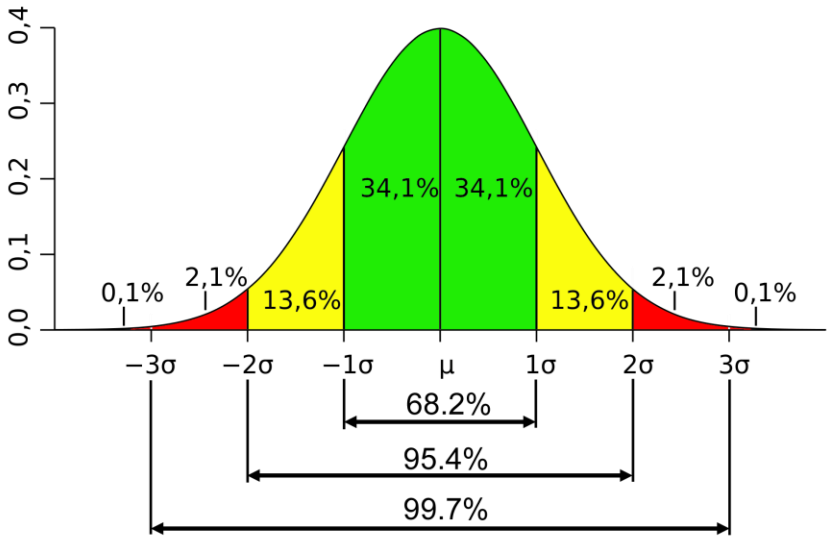

Figure 10: General form of Gaussian distribution. The distribution is centered around (μ) and about 68.2% of its area lies between $(\mu - \sigma)$ and $(\mu + \sigma)$

The general equation of the normal distribution function is:

$$f(\alpha) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\alpha-\mu}{\sigma}\right)^2} \quad (9)$$

where α is the value of the answer we want to compute its probability density, μ is the value of the mean answer for the question and σ is the standard deviation of the answers.

To calculate the probability of a specific answer, we integrate (Equation 9) from $\alpha - D$ to $\alpha + D$ w.r.t α to get the following equation.

$$p(\alpha_i^j) = \frac{1}{\sigma\sqrt{2\pi}} \int_{\alpha-D}^{\alpha+D} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} . dx \quad (10)$$

$$p(\alpha_i^j) = \frac{1}{2}\left(erf\left(\frac{\alpha + D - \mu}{\sigma\sqrt{2}}\right) - erf\left(\frac{\alpha - D - \mu}{\sigma\sqrt{2}}\right)\right) \quad (11)$$

The value of D is another hyper-parameter that we have to choose. However, it doesn't have a great impact on the performance of the model. We choose a value of $\frac{1}{2}$ for it and stick to this value along this section and the coming ones.

Example 4:

For the following histogram vector find the mean and the standard deviation of the answers and use them to plot a normal distribution:

$$h = \begin{pmatrix} 31 \\ 11 \\ 7 \\ 9 \\ 42 \end{pmatrix}$$

Solution:

$$\mu = \frac{31 * 1 + 11 * 2 + 7 * 3 + 9 * 4 + 42 * 5}{31 + 11 + 7 + 9 + 42} = 3.2$$

$$\sigma = \frac{31*(1-3.2)^2+11*(2-3.2)^2+7*(3-3.2)^2+9*(4-3.2)^2+42*(5-3.2)^2}{100} = 3.08$$

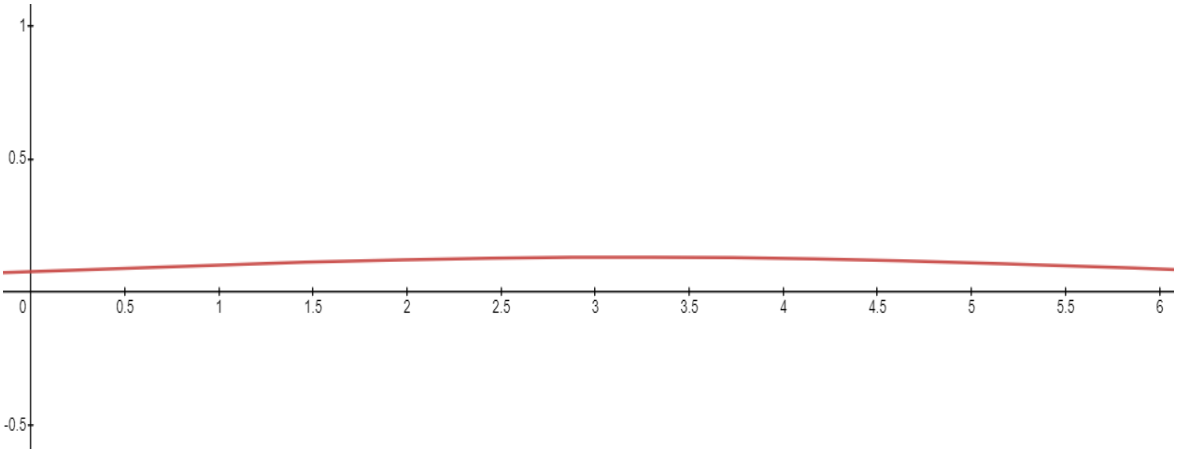

Figure 11: The normal PDF of the question. The distribution is centered at point 3.2, however it is almost flat because of the high standard deviation value.

We could build a normal PDF by finding the mean answer for each question in the input data and the standard deviation of the answers. Afterwards, we use the distribution to generate responses. Therefore, the model will generate the mean answer at a high frequency. The frequency of nearby answers decreases as you go further away from the mean answer.

This PDF clearly satisfies the second specification. However, it can not satisfy the first one. The data distribution of the input could change entirely, resulting in expected data with very little accuracy. However, in the coming section, we will see how we can combine the discrete PDF and the normal PDF to get a PDF that satisfies the two specifications.

4.6.3 Multimodal Normal Distribution Function

Up until now, we saw how the discrete PDF could conserve the data distribution of the input data, and how the normal PDF could use the relation between answers to extrapolate the counts of answers. Multimodal PDF is a mixture of the PDFs. (Figure 12) shows an example of a multimodal distribution.

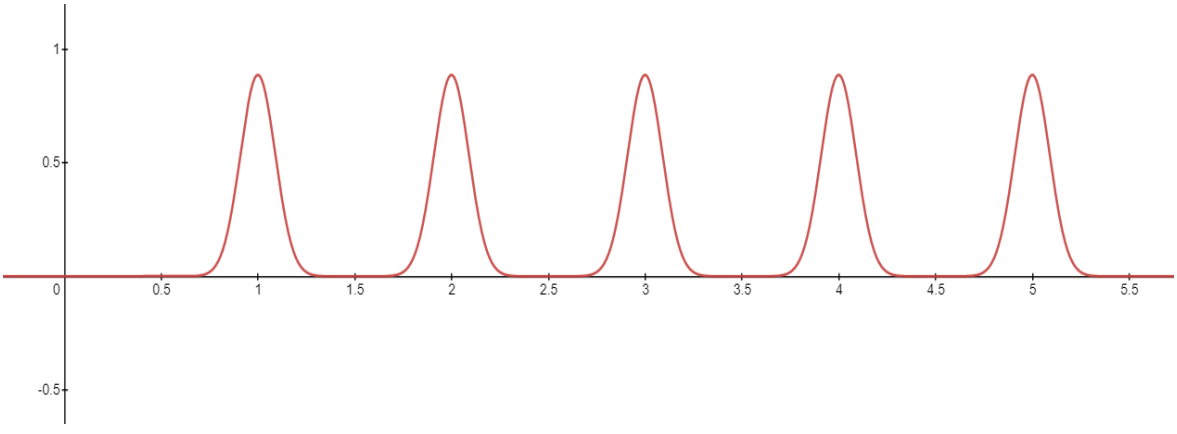

Figure 12: An example of a multimodal normal PDF.

The following is the general equation of the PDF:

$$f(\alpha) = \frac{1}{m\sigma\sqrt{2\pi}}\left(\eta_1 e^{-\frac{1}{2}\left(\frac{\alpha-\mu_1}{\sigma}\right)^2} + \eta_2 e^{-\frac{1}{2}\left(\frac{\alpha-\mu_2}{\sigma}\right)^2} + \eta_3 e^{-\frac{1}{2}\left(\frac{\alpha-\mu_3}{\sigma}\right)^2} + \cdots + \eta_l e^{-\frac{1}{2}\left(\frac{\alpha-\mu_l}{\sigma}\right)^2}\right) \quad (12)$$

As for our model, for the following answer vector

$$a = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}$$

The multimodal PDF is

$$f(\alpha) = \frac{1}{m\sigma\sqrt{2\pi}}\left(\eta_1 e^{-\frac{1}{2}\left(\frac{\alpha-1}{\sigma}\right)^2} + \eta_2 e^{-\frac{1}{2}\left(\frac{\alpha-2}{\sigma}\right)^2} + \eta_3 e^{-\frac{1}{2}\left(\frac{\alpha-3}{\sigma}\right)^2} + \eta_4 e^{-\frac{1}{2}\left(\frac{\alpha-4}{\sigma}\right)^2} + \eta_5 e^{-\frac{1}{2}\left(\frac{\alpha-5}{\sigma}\right)^2}\right) \quad (13)$$

where η_j is the appearance count of the j^{th} answer and m is a normalizing term, and is the total number of responses where

$$m = \sum_{j=1}^{5} \eta_j \quad (14)$$

To compute the probability of a given answer, we integrate (Equation 13) from $\alpha - D$ to $\alpha + D$ w.r.t α to get the following equation:

$$p(\alpha) = \frac{1}{2m}\Big(\eta_1 \operatorname{erf}\Big(\frac{\alpha + D - 1}{\sigma\sqrt{2}}\Big) - \eta_1 \operatorname{erf}\Big(\frac{\alpha - D - 1}{\sigma\sqrt{2}}\Big) + \eta_2 \operatorname{erf}\Big(\frac{\alpha + D - 2}{\sigma\sqrt{2}}\Big) - \eta_2 \operatorname{erf}\Big(\frac{\alpha - D - 2}{\sigma\sqrt{2}}\Big) + \eta_3 \operatorname{erf}\Big(\frac{\alpha + D - 3}{\sigma\sqrt{2}}\Big) - \eta_3 \operatorname{erf}\Big(\frac{\alpha - D - 3}{\sigma\sqrt{2}}\Big) + \eta_4 \operatorname{erf}\Big(\frac{\alpha + D - 4}{\sigma\sqrt{2}}\Big) - \eta_4 \operatorname{erf}\Big(\frac{\alpha - D - 4}{\sigma\sqrt{2}}\Big) + \eta_5 \operatorname{erf}\Big(\frac{\alpha + D - 5}{\sigma\sqrt{2}}\Big) - \eta_5 \operatorname{erf}\Big(\frac{\alpha - D - 5}{\sigma\sqrt{2}}\Big)\Big) \quad (15)$$

Assuming that

$$d_1 = \begin{pmatrix} \operatorname{erf}\left(\frac{\alpha-D-1}{\sigma\sqrt{2}}\right) \\ \operatorname{erf}\left(\frac{\alpha-D-2}{\sigma\sqrt{2}}\right) \\ \operatorname{erf}\left(\frac{\alpha-D-3}{\sigma\sqrt{2}}\right) \\ \operatorname{erf}\left(\frac{\alpha-D-4}{\sigma\sqrt{2}}\right) \\ \operatorname{erf}\left(\frac{\alpha-D-5}{\sigma\sqrt{2}}\right) \end{pmatrix} \quad \& \quad d_2 = \begin{pmatrix} \operatorname{erf}\left(\frac{\alpha+D-1}{\sigma\sqrt{2}}\right) \\ \operatorname{erf}\left(\frac{\alpha+D-2}{\sigma\sqrt{2}}\right) \\ \operatorname{erf}\left(\frac{\alpha+D-3}{\sigma\sqrt{2}}\right) \\ \operatorname{erf}\left(\frac{\alpha+D-4}{\sigma\sqrt{2}}\right) \\ \operatorname{erf}\left(\frac{\alpha+D-5}{\sigma\sqrt{2}}\right) \end{pmatrix} \quad (16)$$

Then,

$$p(\alpha) = \frac{1}{2m} h^T (d_2 - d_1) \quad (17)$$

We could build a multimodal PDF for a question and use it by plugging the histogram vector of the question in (Equation 17). In fact, the histogram vector is the trainable parameter, which is very straightforward. Another parameter in this PDF is the standard deviation σ. Its value could be updated dynamically as the model

goes, however, keeping it constant and dealing with it as a hyper-parameter has a benefit.

The standard deviation controls the width of each modal, in other words, it controls how much the probability density of each central answer affects the nearby answers. Because the area under the curve of any PDF must sum up to 1, the increase in the probability density of nearby answers makes the probability density of the central answer decrease. We can think of σ as a hyper-parameter that controls how much probability density you take from the central answer to the nearby answers. If $\sigma = 0$, the multimodal PDF turns into a discrete PDF and the probability density of the nearby answers is never affected by the probability density of the central answer. (Table 2) and (Figures 13, 14, 15, 16, 17) show how different values of σ affects the probability densities of the central answer and nearby answers.

σ	**Probability density percentage for the central answer**	**Probability density percentage for nearby answers**
0	100%	0%
0.3	90%	10%
0.39	80%	20%
0.48	70%	30%
0.59	60%	40%
0.73	50%	50%

Table 2

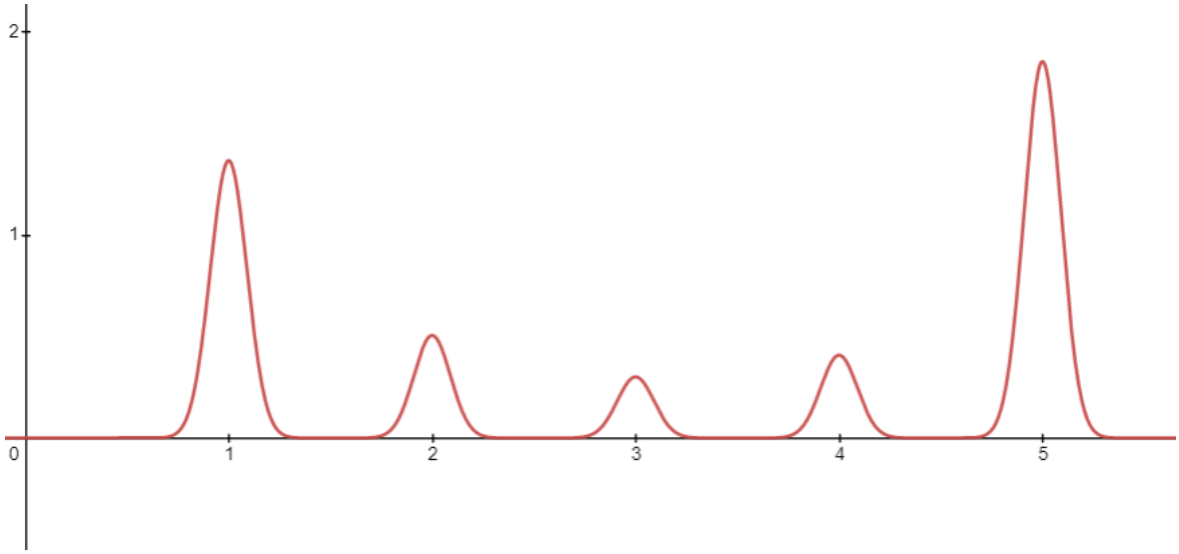

Figure 13: Shows the effect of the hyper-parameter σ on the multimodal normal PDF, $\sigma = 0.3$.

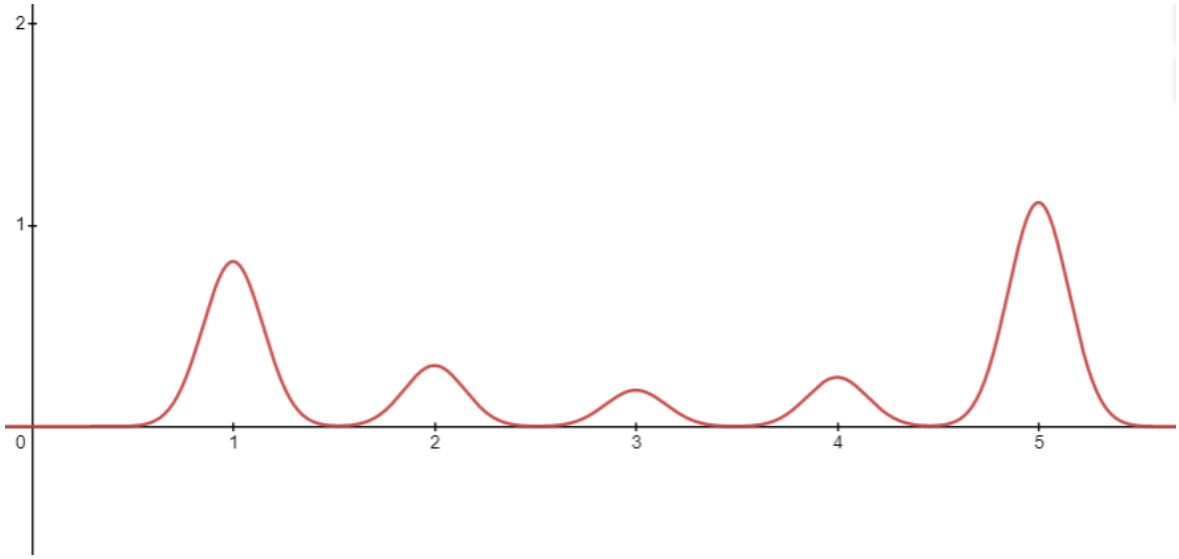

Figure 14: Shows the effect of the hyper-parameter σ on the multimodal normal PDF, $\sigma = 0.39$.

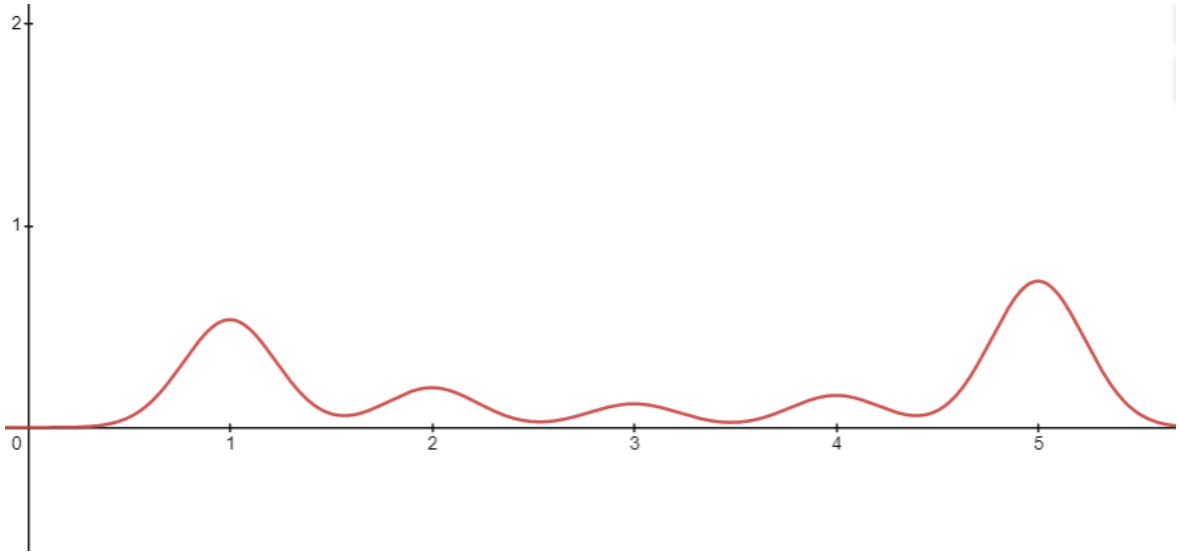

Figure 15: Shows the effect of the hyper-parameter σ on the multimodal normal PDF, $\sigma = 0.48$.

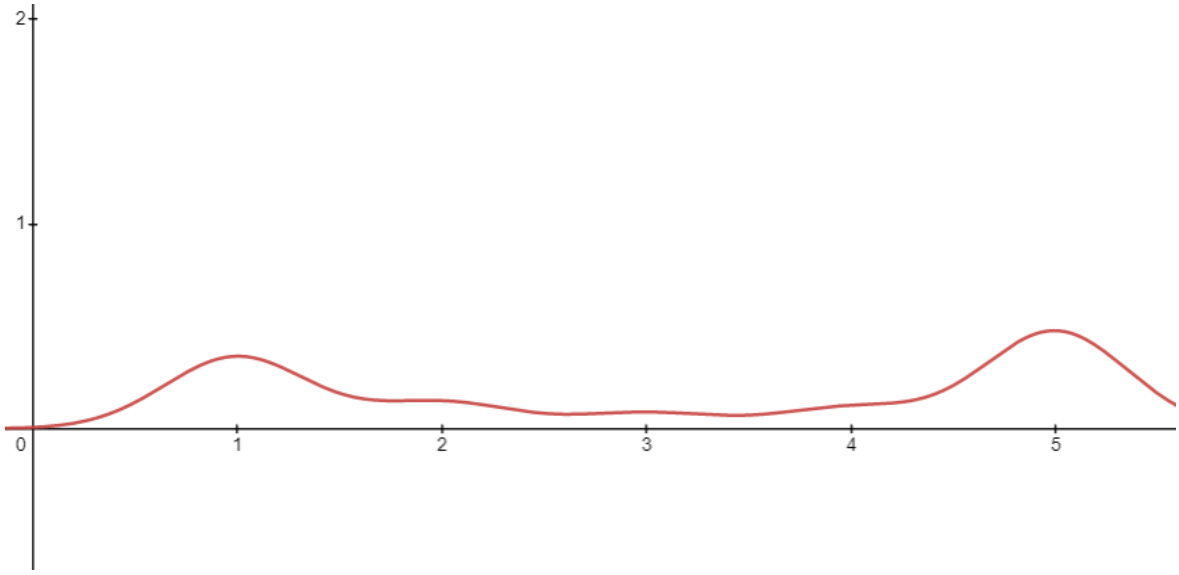

Figure 16: Shows the effect of the hyper-parameter σ on the multimodal normal PDF, $\sigma = 0.59$.

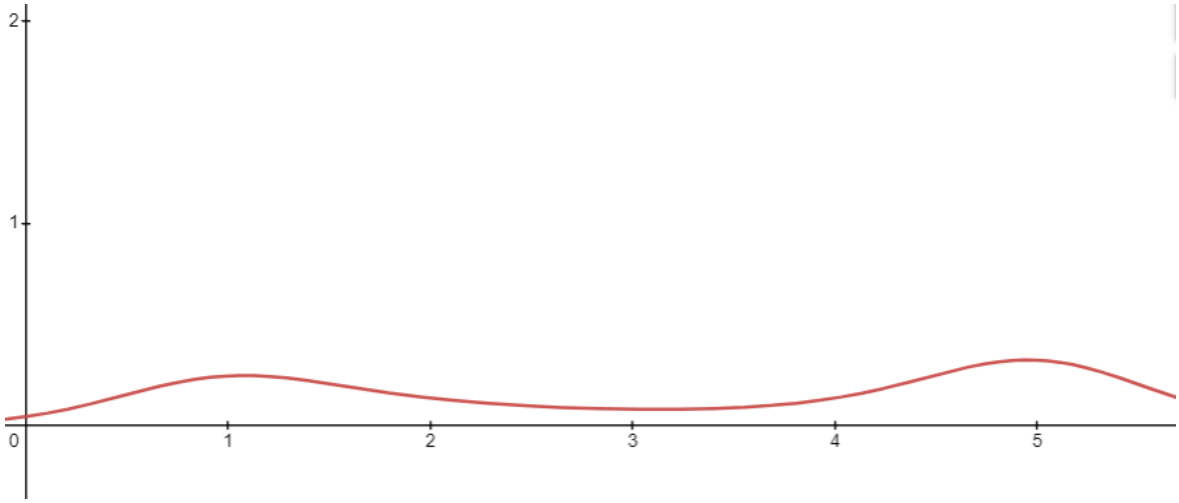

Figure 17: Shows the effect of the hyper-parameter σ on the multimodal normal PDF, $\sigma = 0.73$.

The value of σ controls the trade-off between the data distribution conservation and the data extrapolation. The greater the value of σ, the greater the extrapolation. As a result, its value should be proportional with the expected correction rate. If you expect many latent respondents to read the generated answers, you could safely increase σ. If this is not the case, you should keep σ low. The recommended value is 0.3, which almost keeps the data distribution the same, with little extrapolation.

(Figure 18) shows the result of a test conducted with an input data distribution similar to the full data distribution. The test is conducted 5 times and the average expected answer is used to calculate the accuracy.

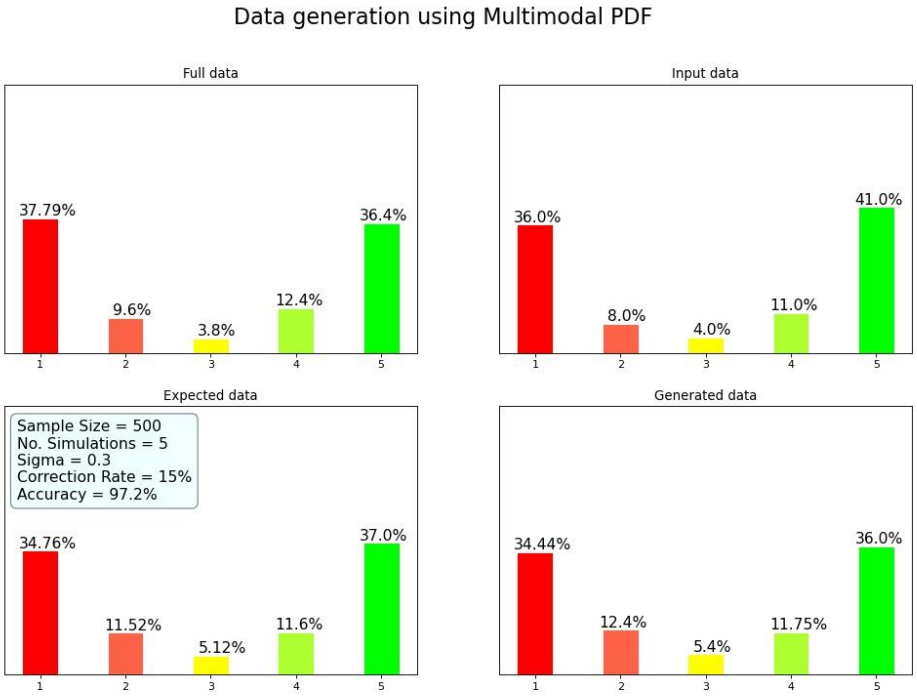

Figure 18: The results of testing data generation using multimodal PDF. The input data has a data distribution similar to that of the full data, which results in high accuracy expected data.

(Figure 19) Shows another test, but, this time the input data distribution is slightly different from the full data distribution. Specifically, answer 4 in the input data has be chosen only 7% of the time while it has been chosen about 18.6% of the time in the full data. However, because it is close to the dominant answer which is 3, it has a percentage of 14.6% in the generated data and in turn has a percentage of 13.08% in the expected data. This extrapolation can't be achieved using the discrete PDF.

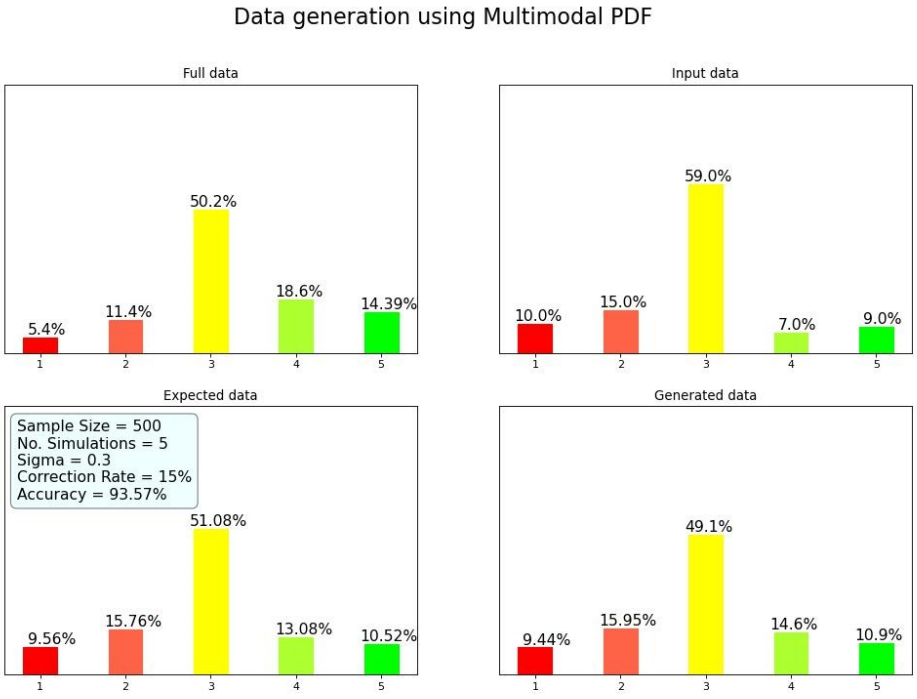

Figure 19: The results of testing data generation using multimodal PDF. The input data has a data distribution slightly different from that of the full data, however, the generated data extrapolates the answer counts and the expected data becomes of an acceptable accuracy.

4.7 Testing and Results

We follow the same testing procedure we followed along the chapter. This time, we test the effect of different variants on the performance of the model. We variate:

- σ: The standard deviation of the modals.
- Dataset sizes.
- Data distribution patterns.

We test using datasets generated by a computer algorithm that generates data assembling those found in rating scale surveys. For rating questions, there is always one or two dominant answers with the highest choice count. The choice count of answers decreases the further away you go from the dominant answers. We use this fact to generate quality data that assembles rating scale survey data.

4.7.1 The effect of varying σ:

(Figures 40, 41, 42, 43, 44) show how the model could conserve the data distribution of any shape if $\sigma = 0.3$, and input data distribution is similar to the full data distribution. Increasing the value of σ would result in more extrapolation and less conservation.

(Figures 45, 46, 47, 48, 49) show the same samples but with $\sigma = 0.48$. The accuracy of the results has decreased due to the increase in extrapolation.

One might argue that extrapolation is not necessary. This is not true because extrapolation is how we model the general pattern of rating scale survey responses. It is a fact that answer counts always follow a normal distribution around the dominant answer. If this pattern doesn't appear in our sample, it will appear if we increase its size. In fact, the multimodal normal distribution of any sample would eventually converge to a unimodal normal distribution if we have a big enough sample (more responses). Extrapolation gives us this power of provisioning how our sample would look like if

it goes big enough, and generates data that copes with this vision. Hence, extrapolation is extremely necessary for our model and, arguably, it is what makes it as powerful.

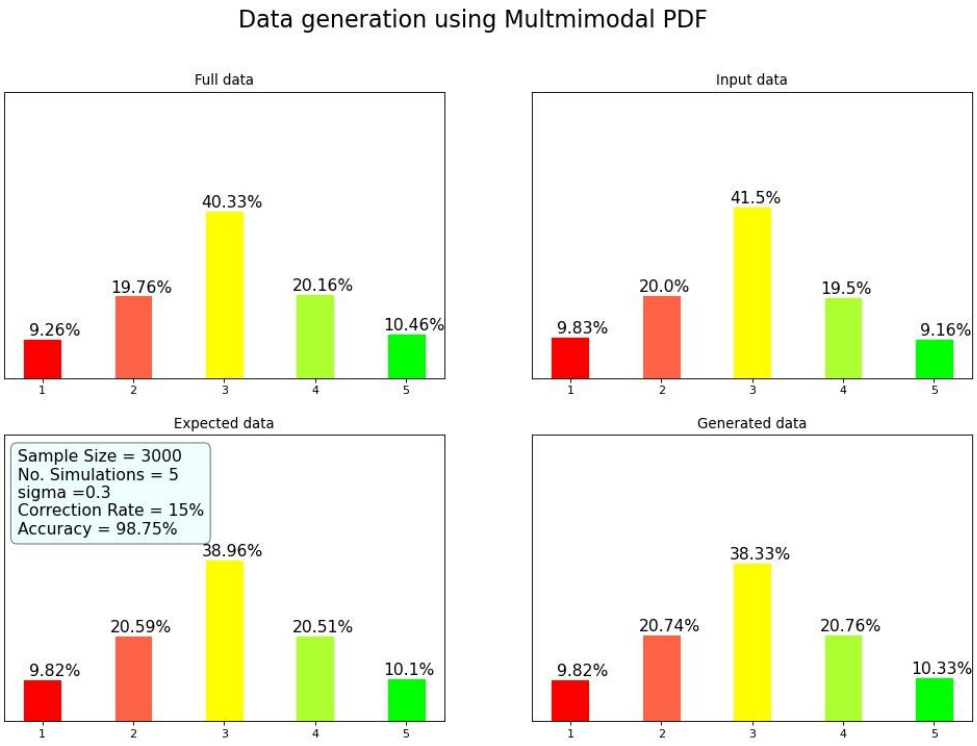

Figure 40

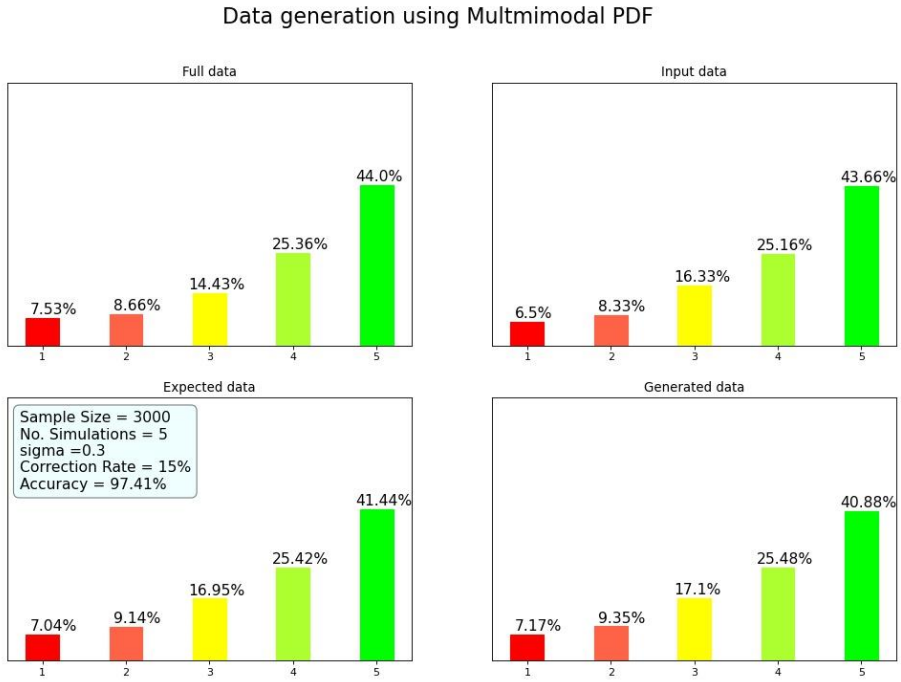

Figure 420

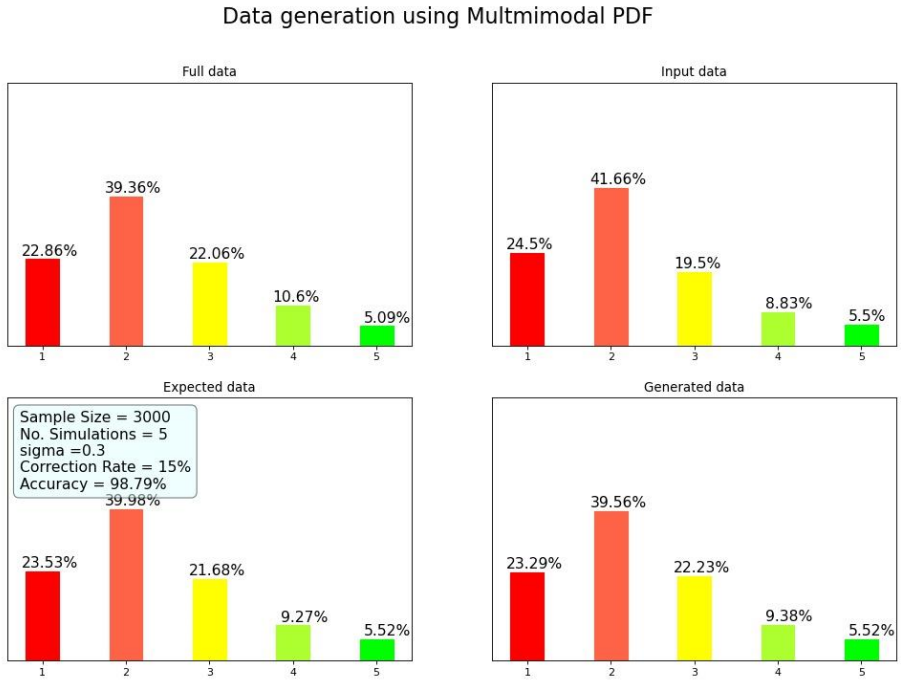

Figure 421

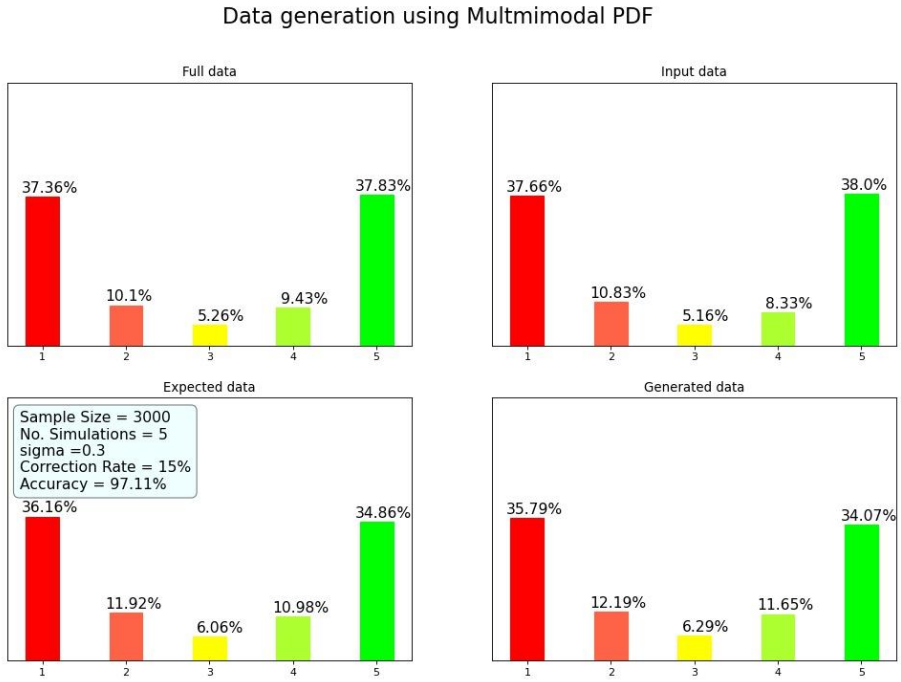

Figure 422

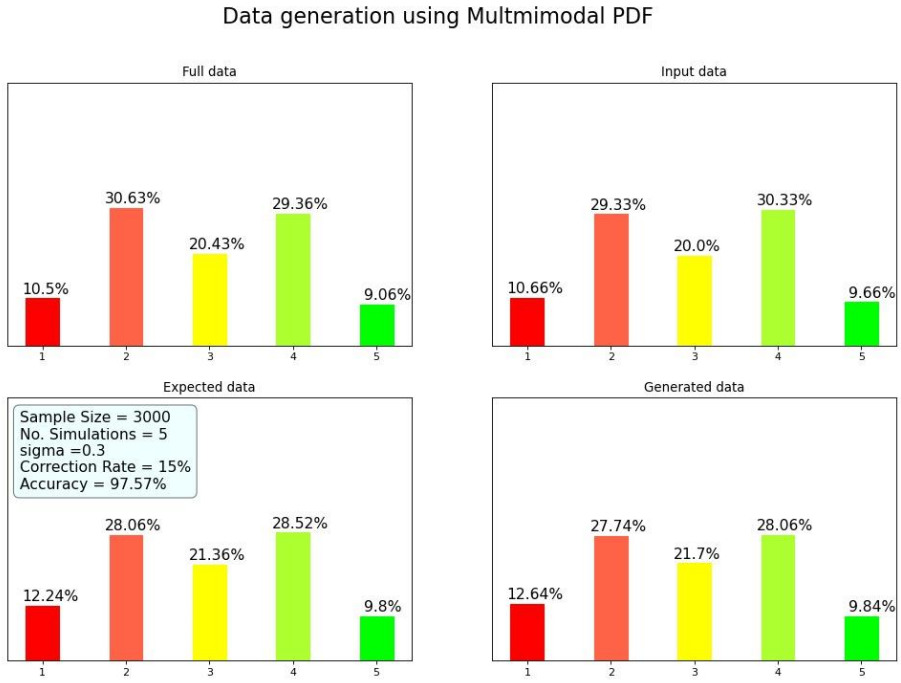

Figure 423

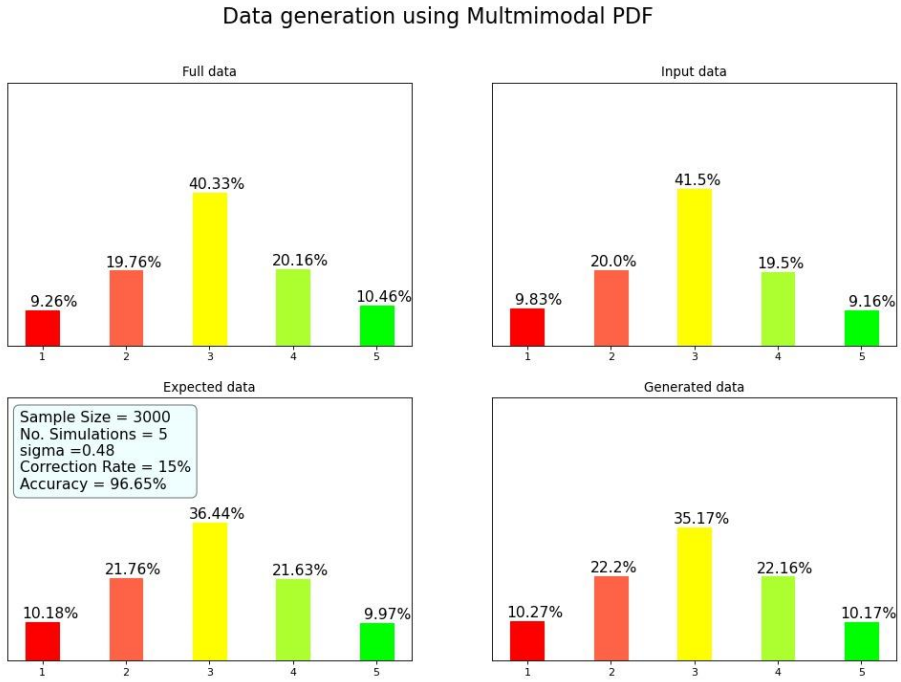

Figure 424

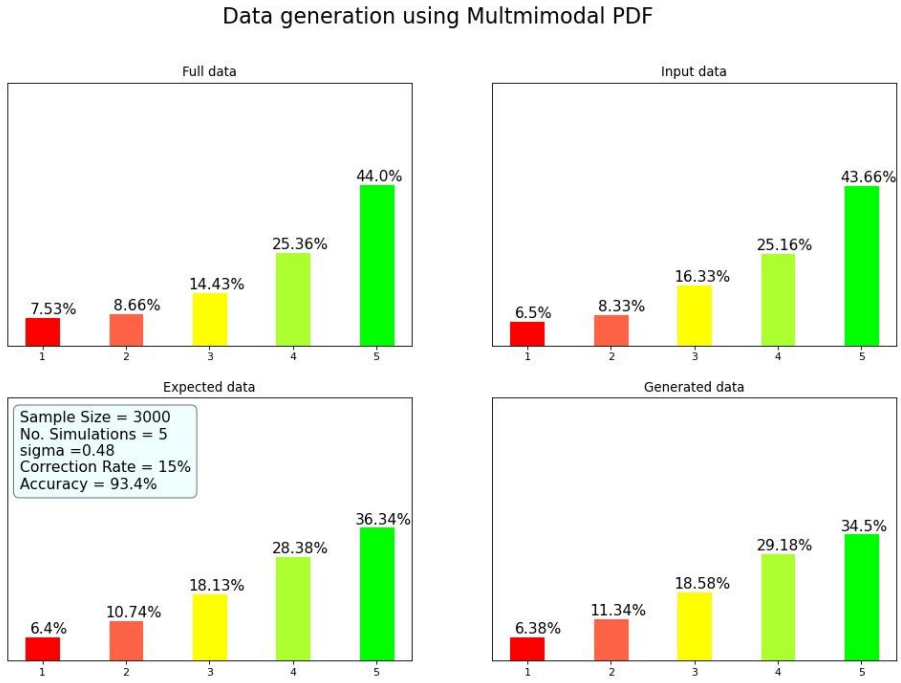

Figure 425

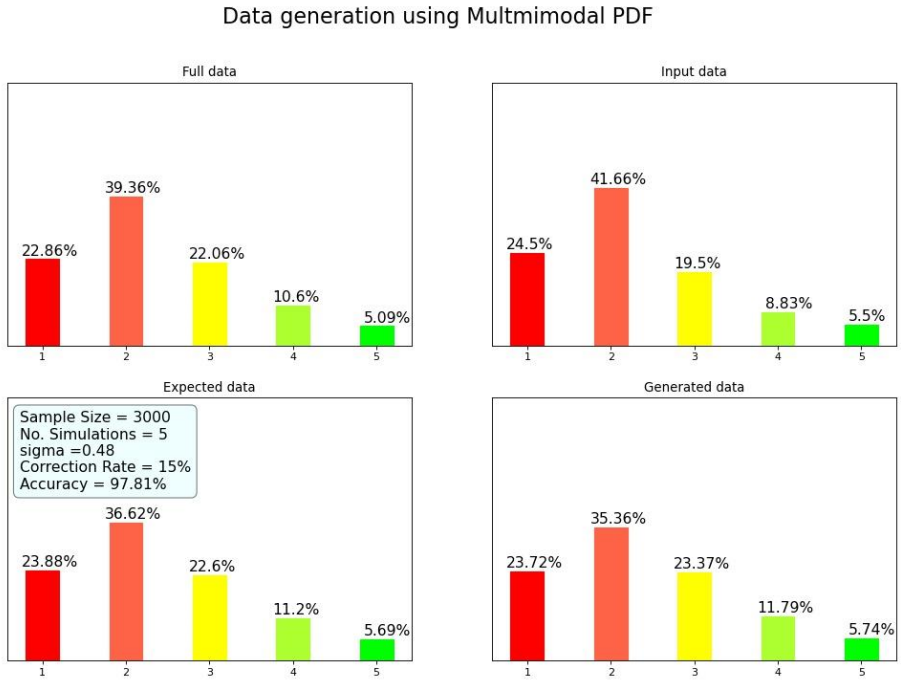

Figure 426

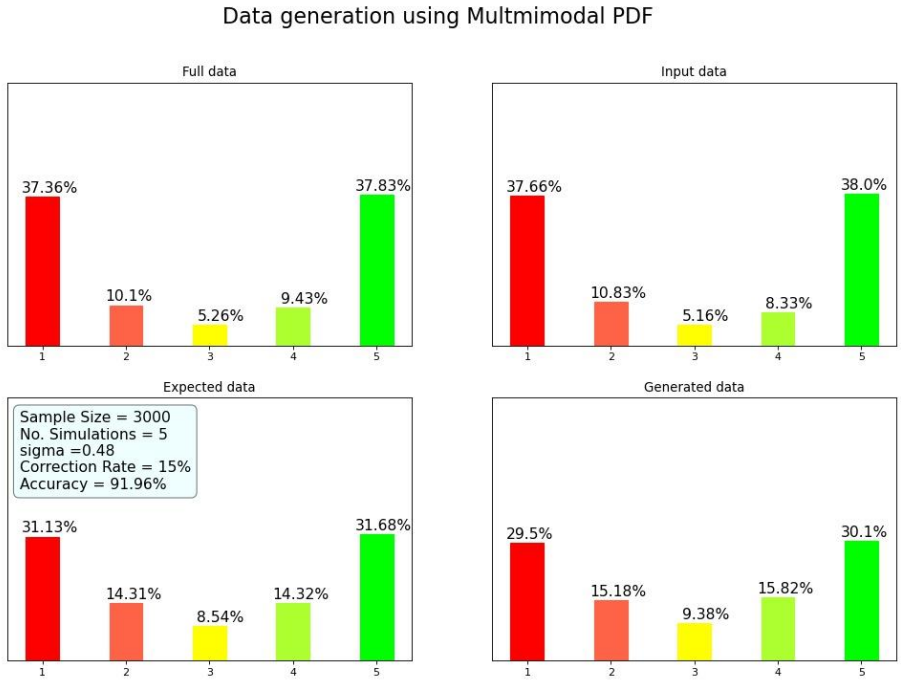

Figure 427

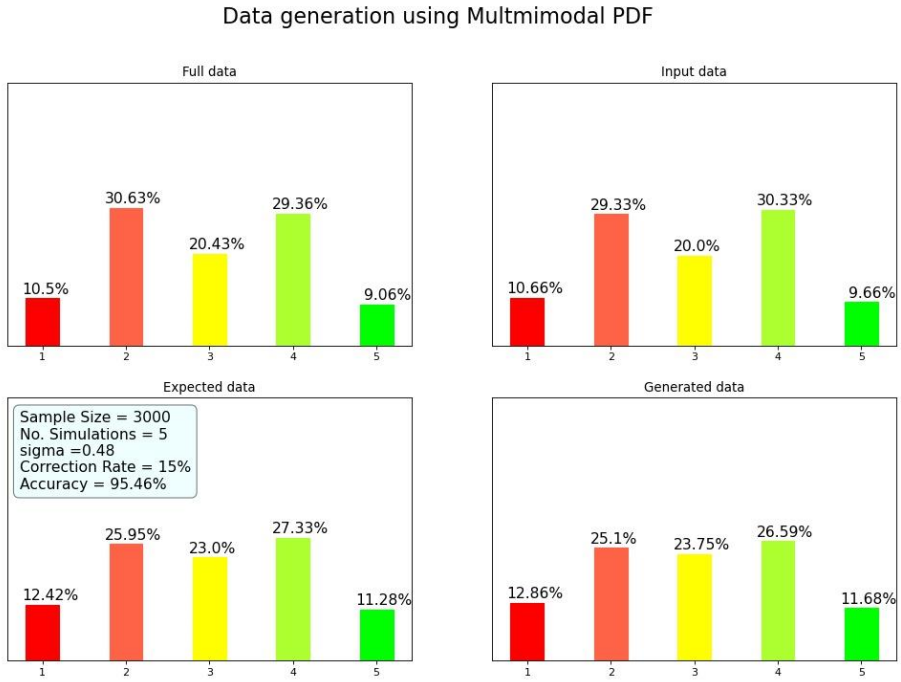

Figure 428

4.7.2 The Effect of Varying the Dataset Size:

Most Machine Learning models require much data to learn and to function but our model doesn't. The model could learn from very little data and function. However, the problem involves the data itself. We argued that any rating scale survey sample would converge to a normal distribution if it were big enough. Otherwise, the data might be skewed and the model might make wrong extrapolation. This is why organizations should make sure that the number of first respondents is enough, and that they deliver quality responses.

(Figure 50) shows how little full data can be skewed and lead to wrong expectations.

(Figure 51) shows how increasing the sample size leads the full data to converge to some form of a normal distribution.

(Figures 52, 53, 54) show how the accuracy increases as the sample size increases. This is the result of the convergence of the full data.

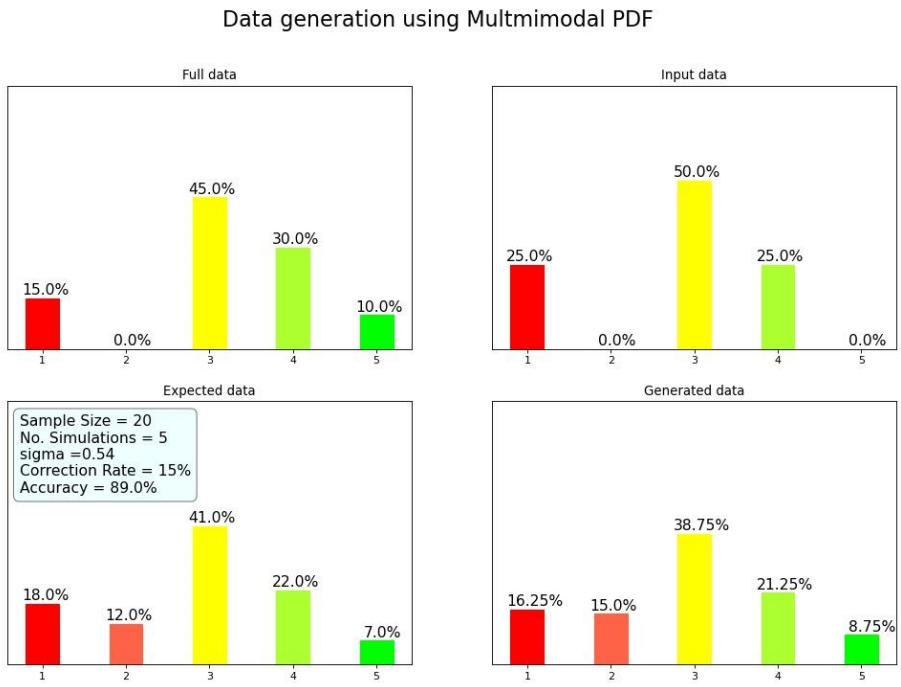

Figure 29: The sample size is 20, which results in skewed data.

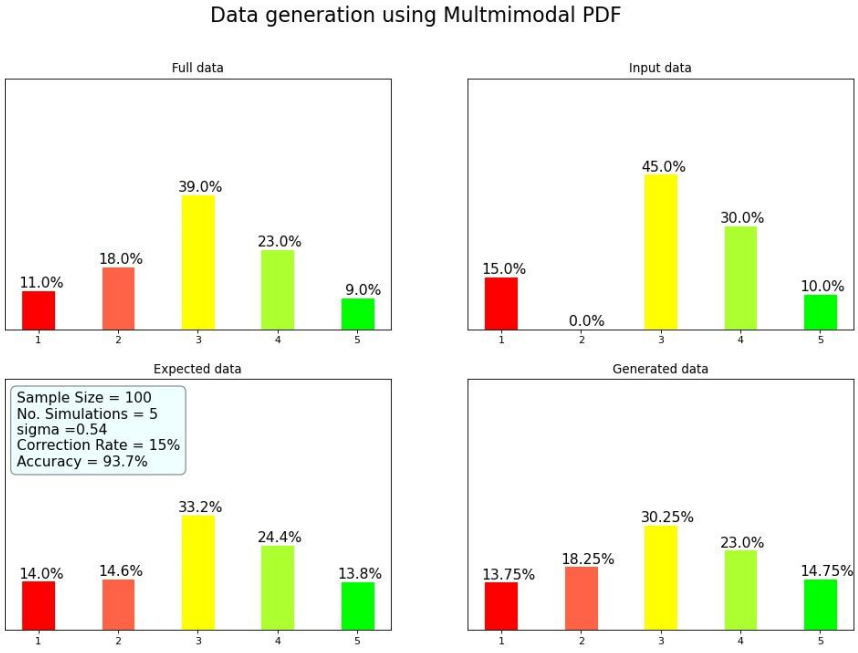

Figure 530: The sample size is increased, which results in full data that starts to converge. Extrapolation made the model eliminate the misleading input data and generate data with high accuracy.

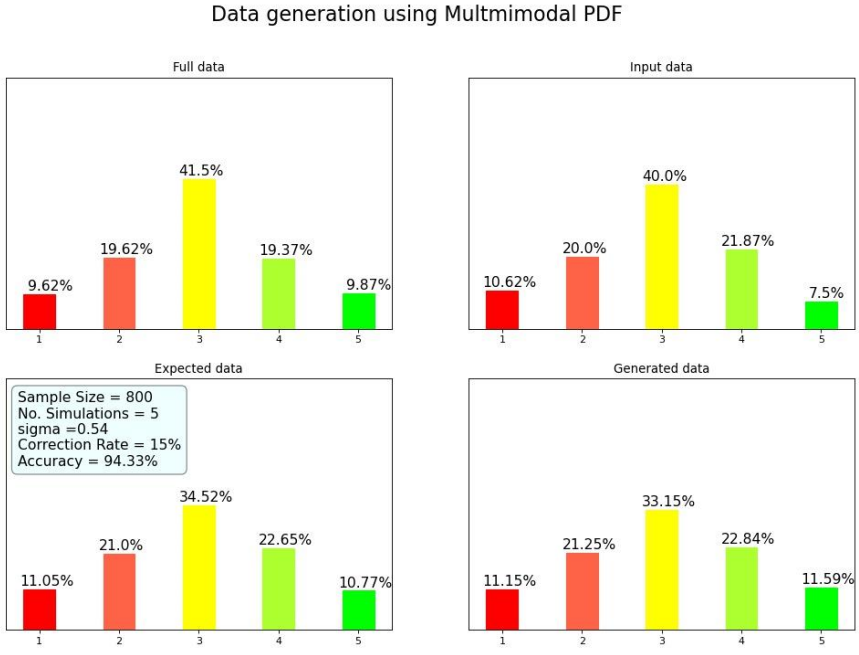

Figure 31: In the full data, increasing the sample size made the probability densities balanced on both sides of the dominant answer.

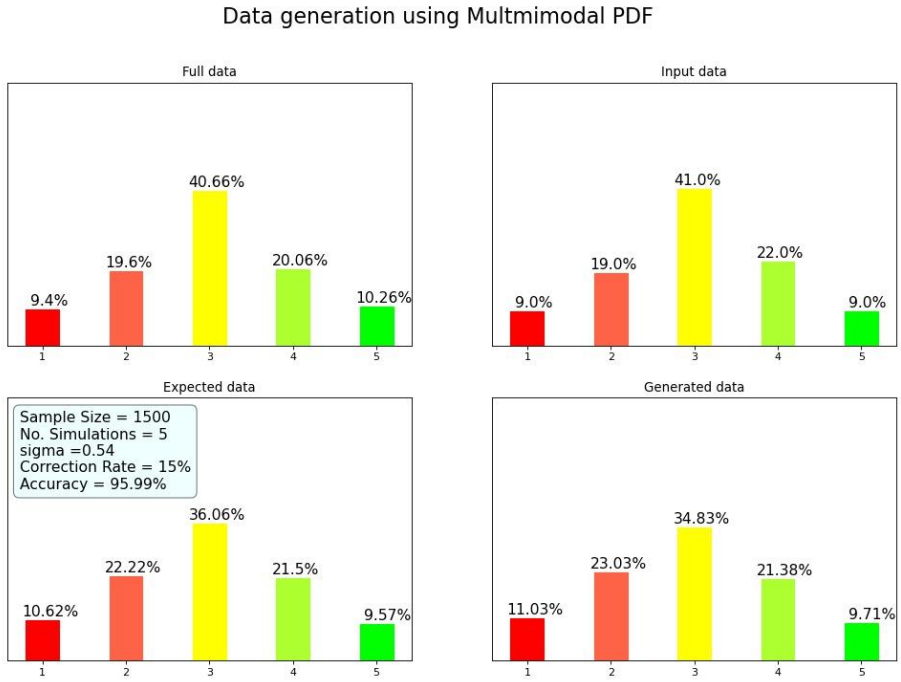

Figure 32

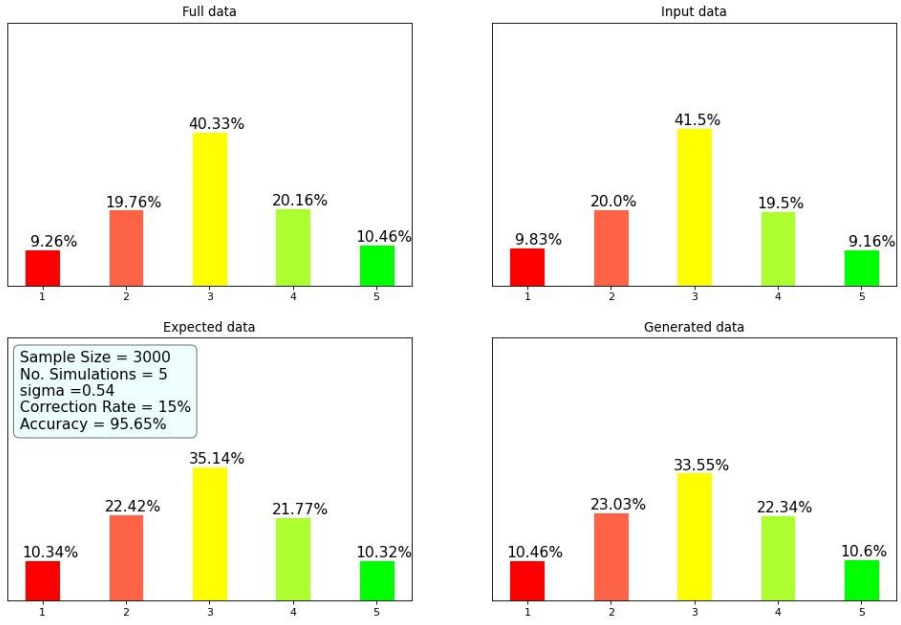

Figure 533

4.8 Case-study, Kafr El-sheikh University

4.8.1 Problem Discussion:

The Faculty of Engineering in Kafr El-sheikh University conducts a course rating survey at the end of each term. Students of the faculty take the survey before they are allowed to access their final grades online. If a student doesn't take the survey, he can't access his grades. This rule aims to increase the response rate to the survey. However, each course rating survey has 58 questions that are answered on a scale from 1 to 5. All courses have the same questions. If a student studies 6 courses in a term he has to take 6 surveys, and hence, he has to answer 348 questions. The problem is that most students procrastinate taking the surveys and only take them when their grades are available. Their eagerness to know their grades makes them fabricate their survey answers. This results in unreliable data that cannot support decision making about courses. We could utilize our model to solve this problem.

4.8.2 Proposed Solution:

We propose the following solution:

- From the 58 questions, choose a small number to be required questions.
- Let 20% of the students take the full survey to train the model.
- Afterwards, make it possible for students to take quick surveys.
- A quick survey is a survey at which required questions are mandatory and the rest of the questions are optional.
- Optional questions are answered by the model and displayed to the student as answered questions.
- The student can change the answer of any question as he likes (human correction) before submitting his answer.

- The model is, then, trained using this newly submitted answer.

The data quality of the final data depends on the data quality of the first respondents. In our case, this is achievable if we assume that students who take the surveys early answer them with more veracity.

4.8.3 Testing the Model

To ensure that our model could actually be applied in this problem, we acquired real data from the faculty and performed our test procedure. (Figures 55, 56, 57) show the results of 3 question of a course survey from the college. The average expectance accuracy of this survey is 95.903%.

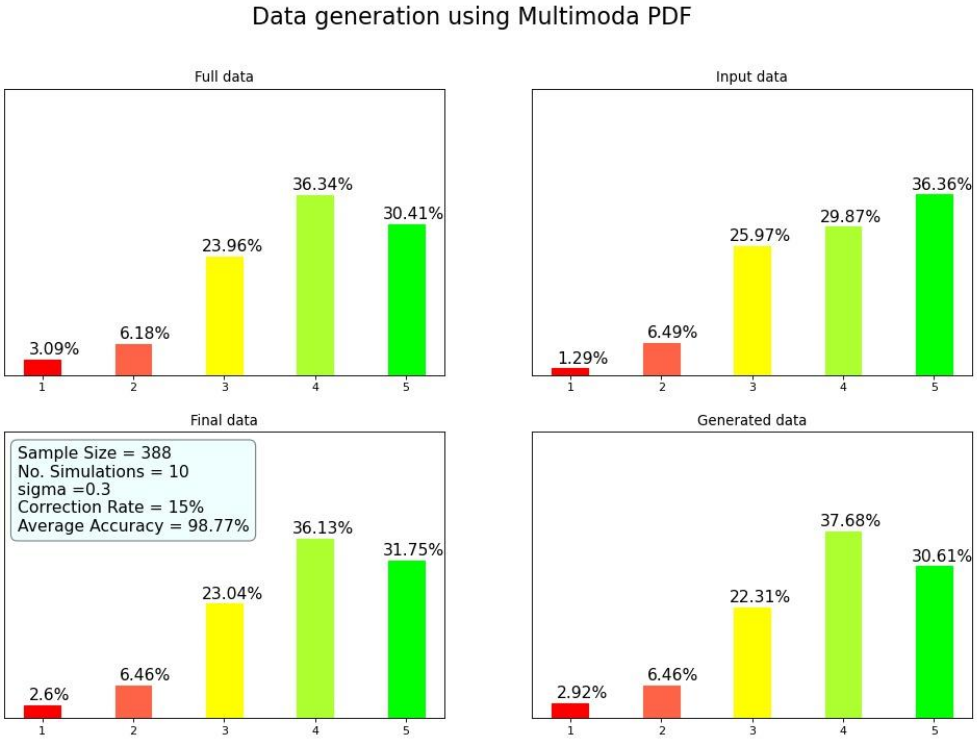

Figure 34

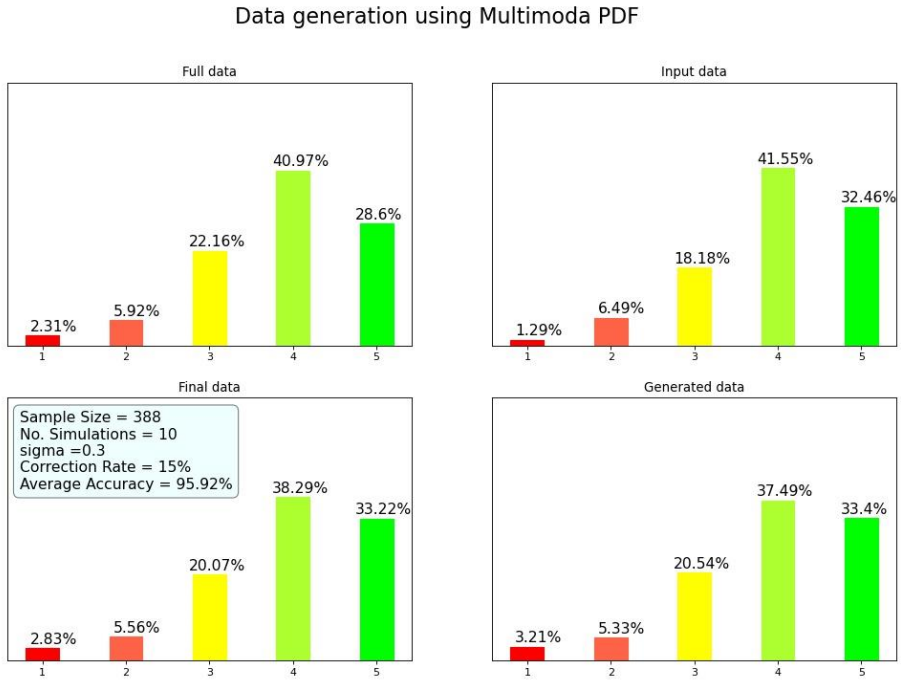

Figure 35

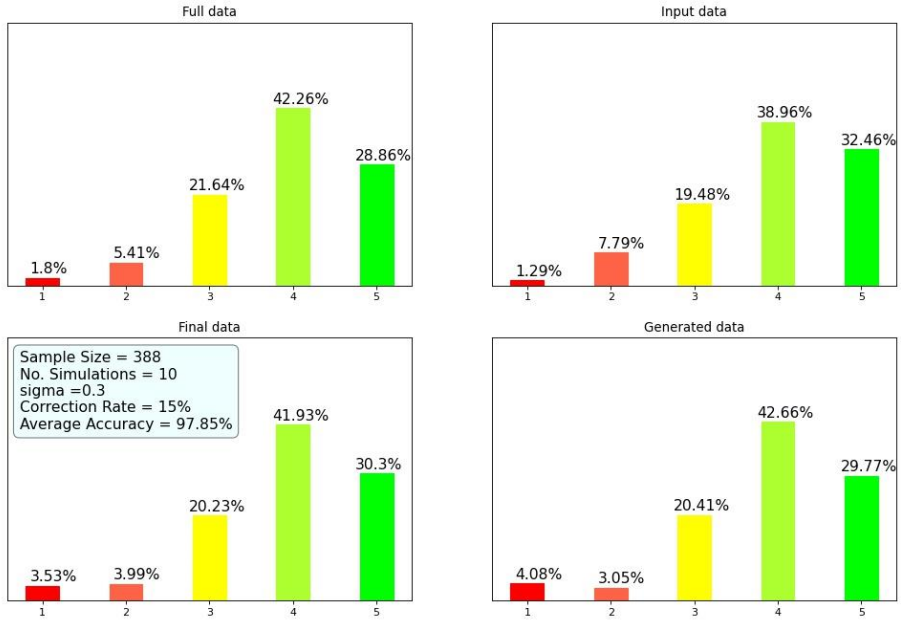

Figure 36

4.8.4 The Model's Impact on the System:

As for the impact of the model on the system. It depends on the implementation and the hardware specifications. If the parameters are updated at each submission, the time taken to generate an answer becomes independent from the number of responses. According to (Equation 15), the time complexity of generating an answer is $O(n * l)$, where n is the number of questions in a survey and l is the number of answers for each question. Hence, for each survey, the model generates an answer in a constant time. For the hardware specifications shown in (Figure 58), and for a survey with 58 questions that contain 5 answers each, it took the model an average of $138\mu s$ to generate an answer.

Operating System:	Windows 10 Pro 64-bit (10.0, Build 19041)
Language:	English (Regional Setting: English)
System Manufacturer:	Dell Inc.
System Model:	Inspiron 5559
BIOS:	BIOS Date: 03/01/16 14:53:59 Ver: 1.1.9.00
Processor:	Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz (4 CPUs), ~2.6GHz
Memory:	16384MB RAM
Page file:	8682MB used, 9832MB available
DirectX Version:	DirectX 12

Figure 37

4.9 Summary

We began the chapter by stating that our problem had two parts; organizations want to get huge amounts of quality data from people, and people don't want to take long and time-consuming surveys. We proposed a model to the problem using Anomaly Detection, however, it turned out to be inefficient. We derived another model that uses PDFs to generate answers. We went on to choose a suitable PDF that can conserve the data distribution and extrapolate the answer counts. We settled on the Multimodal Normal PDF, and saw the impact of many variants on the model. We concluded the chapter by a case-study for the course rating surveys of the Faculty of Engineering, Kafr El-sheikh University. We, finally, proposed a solution and studied the impact of it on the system.