In this deliverable, you will discuss your progress and report your preliminary results. Be precise in your explanation and report.

1. **Problem statement: Restate the initial project that you proposed in deliverable one in 2 - 3 sentences. Be sure to refer back to this problem statement in the following questions.**
   We plan to predict maternal health risks using inputs of user data, such as age, blood pressure, heart rate, and blood sugar. Using a classification model, we aim to enable early risk detection for timely interventions.

2. **Data Preprocessing: Confirm the dataset you are working with. State any changes from the initial dataset you chose. Discuss the content of the dataset (number of samples, labels, etc.). Describe and justify your data preprocessing methods (did you delete or modify any data? If so, why?).**
   This dataset includes key maternal health indicators like age, vital signs, BMI, and blood glucose levels, which are crucial for predicting high-risk pregnancies.
   To preprocess the data, first, we removed any missing values either by mean imputation. Then, we checked for outliers in variables with medical data like blood pressure or glucose levels as if any are too extreme and out of a typical range this could indicate that the data is erroneous removed them. Finally, we performed data normalization (rescaling values to be between 0 and 1) and standardization (transforms data to have a mean of 0 and a standard deviation of 1) for the blood pressure, glucose levels, and heart rate data.

3. **Machine learning model: In the first deliverable, you proposed a model for your project. If you decided to change your model, explain why. Restate your chosen model and elaborate on the design decisions. Report the following:**
   a. **Specify the framework and tools that you used to implement your model. (For instance, did you use any libraries such as PyTorch, Keras, etc. to implement the model? Any other tools? What does the architecture of your model look like? How many layers/modules? etc.) Explain and provide architecture graphs as appropriate.**

We implemented a multiclass logistic regression model with the help of the NumPy library. Our model consists of a single-layer architecture where an input matrix X (with bias added) is multiplied by a weight matrix W to produce logits Z, which are then passed through the softmax function to generate class probabilities. The model is trained using gradient descent, optimizing the cross-entropy loss with weight updates computed as $W=W-\alpha \nabla W$, where ∇W is the gradient of the loss function. The one-hot encoding of target labels Y ensures correct gradient computation.

b. **Justify any decision about training/validation/test splits, regularization techniques, optimization tricks, setting hyper-parameters, etc.**

To ensure balanced evaluation and prevent overfitting while providing enough data for learning, the dataset was split into 80% training and 20% testing. One-hot encoding was applied to categorical labels to make them compatible with the softmax function. Additionally, skewed features such as Blood Sugar and Age were transformed using log and Box-Cox transformations to make them more normally distributed, improving the model's stability.

c. **Description of validation methods How did you test your model? Is your model overfitting or underfitting?**

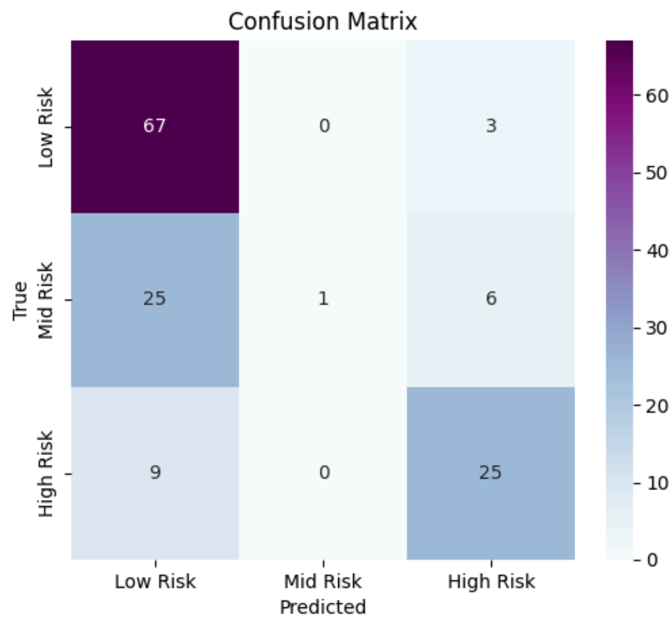
We split the data to be 70% training data and 30% testing data. After training the model, we calculated its accuracy score, confusion matrix, and classification report. Although there might not be extreme overfitting or underfitting due to the model's imbalance toward predicting low and high risk versus mid risk.

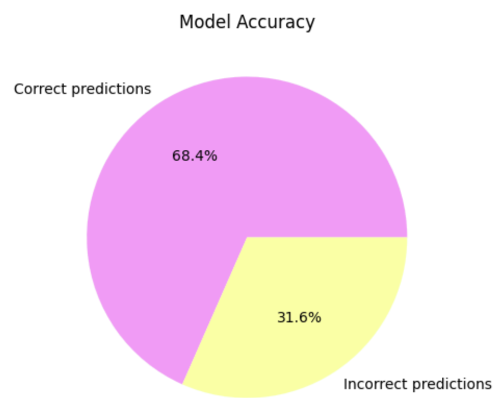
d. **Did you face any challenges implementing the model? If so, how did you solve it? At this point, don't forget to save your trained weights! You will need them for the integration and/or testing of your model!**

It was hard to scale a model we learned to implement for binary classification, to multi-class linear regression, especially when trying to understand and derive the correct formulas for the gradient of the lost function or the softmax activation function. We overcame this by breaking down the formulas, verifying computations with NumPy, and debugging shape mismatches to ensure proper learning.

4. **Preliminary results: In this section, you will focus on the performance of your model. Confirm the metric discussed in Deliverable 1. Present a detailed analysis of your results, providing graphs as appropriate. In addition to an evaluation metric, discuss the overall performance of the model and the feasibility of the project with these results. Remember, graphs are beautiful and we love them!**

Below are graphs for the confusion matrix and accuracy results using logistic regression:





	precision	recall	f1-score	support
0	0.66	0.96	0.78	70
1	1.00	0.03	0.06	32
2	0.74	0.74	0.74	34
accuracy			0.68	136
macro avg	0.80	0.57	0.53	136
weighted avg	0.76	0.68	0.60	136

Although the precision shows the model performs well for mid-risk and decent for low and high risk classification, recall is significantly worse for mid risk. For example, precision is 100% for mid-risk, meaning every point it predicted as mid-risk actually was mid-risk. However, a low recall score indicates it was not able to classify most true mid-risk test points. This could indicate that there are fewer mid-risk samples, or that features might overlap substantially between low and high risk. Moreover, the accuracy is 68%, which is not terrible but could be improved with hyperparameter tuning or choosing a different model. Given our first results, tweaking our data processing and perhaps exploring other models could improve the performance of our model.

5. **Next steps: Discuss your next steps. Describe the pros/cons of your approach and future work. Will you be altering your model? For example, will you be fine-tuning it? At this point, if you think that your model is not performing well and/or does not work, please reach out to your assigned TPM to see what you can do to improve it.**

We currently have a static learning rate of 0.01, a static number of iterations of 500, and a static regulation strength (epsilon) of 0.0001. In the future, we plan to make each of these hyperparameters, and test - using methods like random search, grid search, or cross validation - for which value the model is most accurate.

Currently, our model exhibits 31.6% incorrect predictions which we will try to minimize through the hyperparameter tuning. We observe that among the incorrectly predicted labels, there is a large amount of predicted low risk when the true label is mid risk which suggests issues with that decision boundary.