

Group 6 Contributions

DataSet used: [Iris Dataset](#) + [IMDb Movie Reviews Dataset](#)

Tool used: [Jupyter Notebook](#)

GitHub Repository: <https://github.com/Muen1/probability-bayes-gradient-descent>

Group Members and Roles

Member	GitHub Username	Contribution Summary	Commits
David Akintayo	dav1d001	<ul style="list-style-type: none">Loaded the Iris dataset using Pandas.Selected two features (<code>sepal_length</code> and <code>sepal_width</code>) for analysis.Implemented the Bivariate Normal Distribution from scratch.Generated both Contour and 3D Surface plots to visualize the probability density.	<ul style="list-style-type: none">Uploaded notebook <code>part1_bivariate.ipynb</code> to <code>notebooks/</code> directory.Implemented code to load and clean the Iris dataset using pandas.Computed mean vector and covariance matrix for selected features (<code>sepal_length</code> and <code>sepal_width</code>).Plotted 2D contour and 3D surface visualizations to represent the probability density.Added explanations, markdown descriptions, and interpretation of results in notebook cells.Verified and documented function outputs for

			correctness.
Cynthia Mutie	Muenl	<ul style="list-style-type: none"> • Set up GitHub repository structure. • Identified keywords for positive and negative sentiment. • Calculated Prior, Likelihood, Marginal, and Posterior probabilities using Bayes' theorem. • Visualised keyword frequencies using bar plots. 	<ul style="list-style-type: none"> • Created project folders and Jupyter Notebook. • Loaded and previewed IMDb dataset using pandas • Computed prior, likelihood, marginal and posterior probabilities for positive keywords. • Implemented Bayes' function and printed posterior probabilities.
Sougnabe Payang	Sougnabe	<ul style="list-style-type: none"> • Performed manual gradient descent calculations for linear regression. • Used dataset: (1,3), (3,6), starting with $m=-1$, $b=1$, learning rate $\alpha=0.1$. • Completed the symbolic derivatives and the first iteration, showing step-by-step predictions, gradient calculations, and parameter updates • Made the progressions summary and the interpretations. • Verified results with Part 4 code. 	<ul style="list-style-type: none"> • Added the symbolic derivatives • Added the first iteration • Added the iteration 2 • added the third iteration • Added the forth iteration • Added the summary and interpretations.
Elvis Kayonga	Elvis-Kayonga	<ul style="list-style-type: none"> • Implemented gradient descent in Python for the same dataset. • Used function-based modular code (DRY principle). • Recorded parameter updates and generated plots: error reduction (MSE vs 	<ul style="list-style-type: none"> • Create manual calculations • Rename manual calculations to manual calculations.ipynb • Rename manual calculations.ipynb to manual calculations.py • Add prediction and

		Iterations) and parameter convergence (m and b vs Iterations).	<p>MSE computation functions</p> <ul style="list-style-type: none">• Add gradient descent function for 4 iterations• Implement visualization for gradient descent history• Added visualization functions to plot parameter updates and error over iterations.• Add gradient descent and visualization modules• Also added the manual calculations turned into code in the notebook
--	--	--	--