**REPORT**

**Multinomial Logit Model Implementation and Visualisation using Python**

**Introduction**

Thie report summarizes the assumptions, implementation, findings and visualizations of the function ‘calculate\_probabilities’, developed to compute probabilities of each alternative in a multinomial choice setting using the logistic function. The multinomial logit model is a staple in discrete choice modelling, used to estimate the probability of selecting a particular choice from a set of alternatives based on various independent variables.

**Assumptions**

1. Linear deterministic utility models: The utilities for each alternative are assumed to be a linear combination of independent variables and their respective coefficients (b).

2. Data structure for input data: It is assumed that the input data is provided in a structured dictionary format, with equal length list of values for each variable.

3. Parameters structure: The parameters (b coefficients) are assumed to be provided in a dictionary format.

4. Probability calculation: The probability of each alternative is the exponential of its utility divided by the sum of exponentials of all utilities.

**Methodology**

The ‘calculate\_probabilities’ function follows these steps:

1. Data preparation: Convert input data into a matrix for vectorized operations using numpy arrays.

2. Utility calculation: Compute deterministic utilities for each alternative using defined utility functions.

3. Probability computation: Applies the logistic function (the exponential of its utility divided by the sum of exponentials of all utilities) to transform utilities into probabilities.

**Key Features and Error Handling**

1. Dynamic Handling: The function can accommodate any number of alternatives and independent variables due to vectorization.

2. Error Handling: Error handling implementations include:

* Validation of non-empty inputs for parameters, data, and utilities.
* Consistency in data length across all independent variables.
* Proper working of utility functions with the data and parameter.
* Handling of any other exceptions.

**Findings**

Upon applying the function to sample data, the following key observations were made:

1. Probability Distribution: The computed probabilities for each alternative were observed to be consistent with the logistic function and input data.

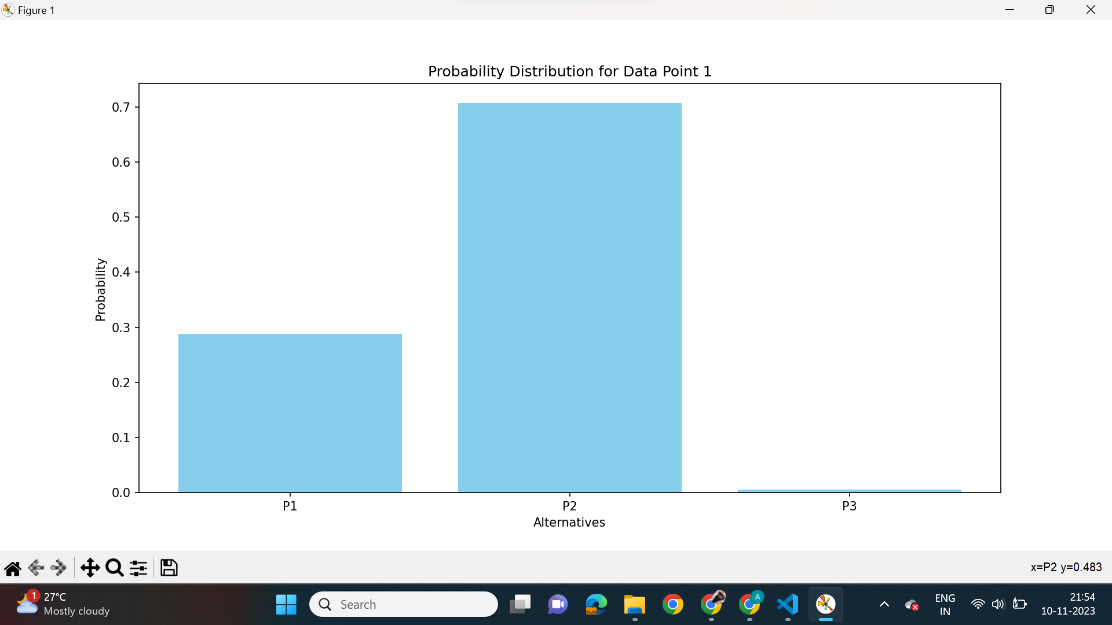
2. Sensitivity to Parameters: Changes in b coefficients significantly influence the probability outcomes, mainly as it a linear model and is linearly proportional to utility functions.

3. Data Dependency: Variations in independent variables led to noticeable shifts in probabilities, highlighting the model's sensitivity to input data.

4. Error Handling: Robust error handling allowed for easy detection of the area with error and the cause.

**Visualizations:**

1. Probability Distribution Bar Charts: For each data point, bar charts were plotted to visually represent the probability distribution across different alternatives.



**Conclusion**

The ‘calculate\_probabilities’ function effectively models multinomial choices, providing valuable insights into how different factors influence the probability of selecting each alternative. Its flexibility to handle various scenarios and robust error handling make it a reliable tool for discrete choice modelling. The visualizations further aid in interpreting the model's outcomes.