**Logistic Regression**

Logistic regression is a [supervised machine learning](https://www.geeksforgeeks.org/supervised-unsupervised-learning/) algorithm mainly used for [classification](https://www.geeksforgeeks.org/getting-started-with-classification/) tasks where the goal is to predict the probability that an instance of belonging to a given class. It is used for classification algorithms its name is logistic regression. it’s referred to as regression because it takes the output of the [linear regression](https://www.geeksforgeeks.org/ml-linear-regression/)function as input and uses a sigmoid function to estimate the probability for the given class. The [difference between linear regression and logistic regression](https://www.geeksforgeeks.org/ml-linear-regression-vs-logistic-regression/) is that linear regression output is the continuous value that can be anything while logistic regression predicts the probability that an instance belongs to a given class or not.

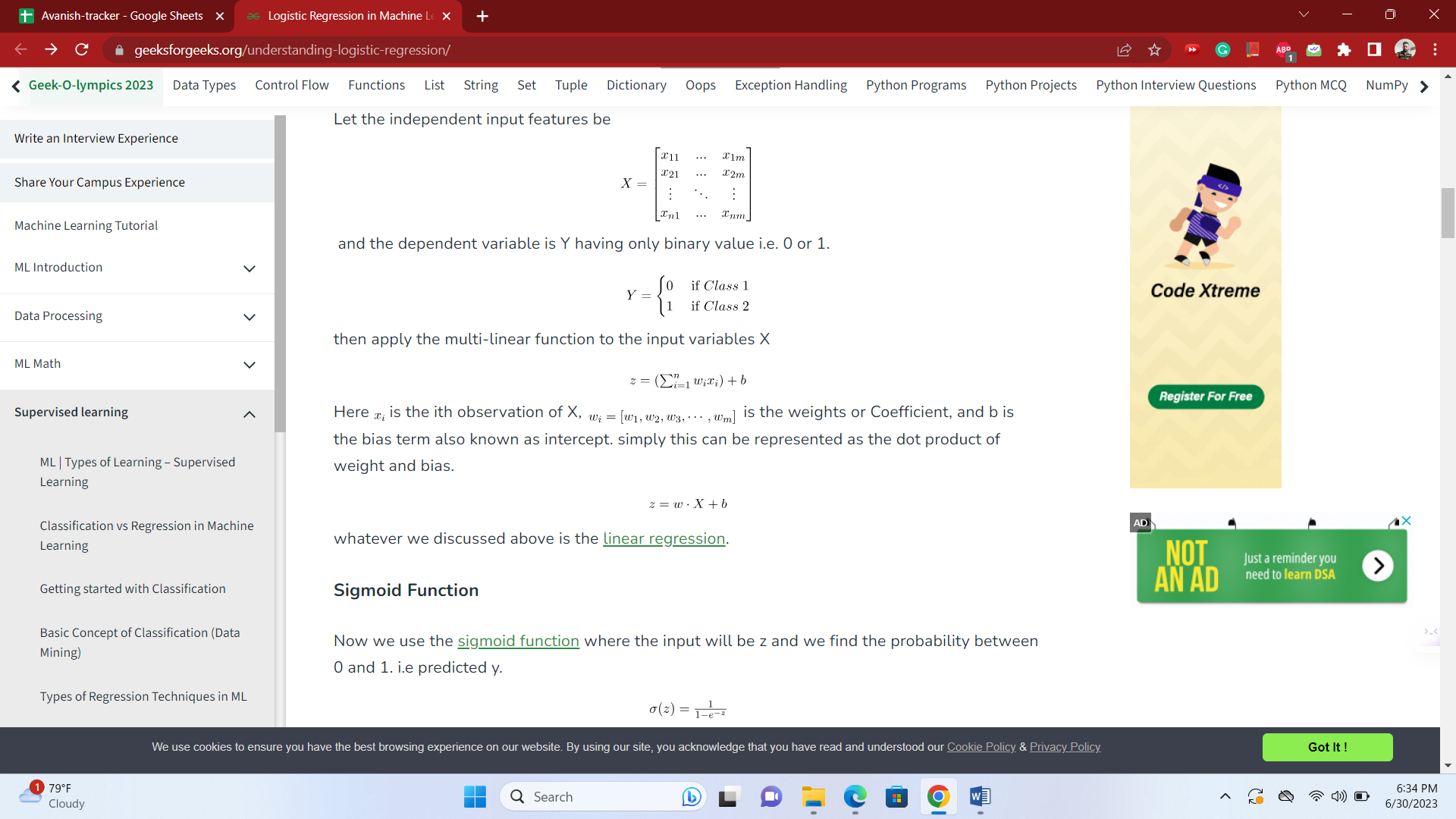
**Terminologies involved in Logistic Regression:**

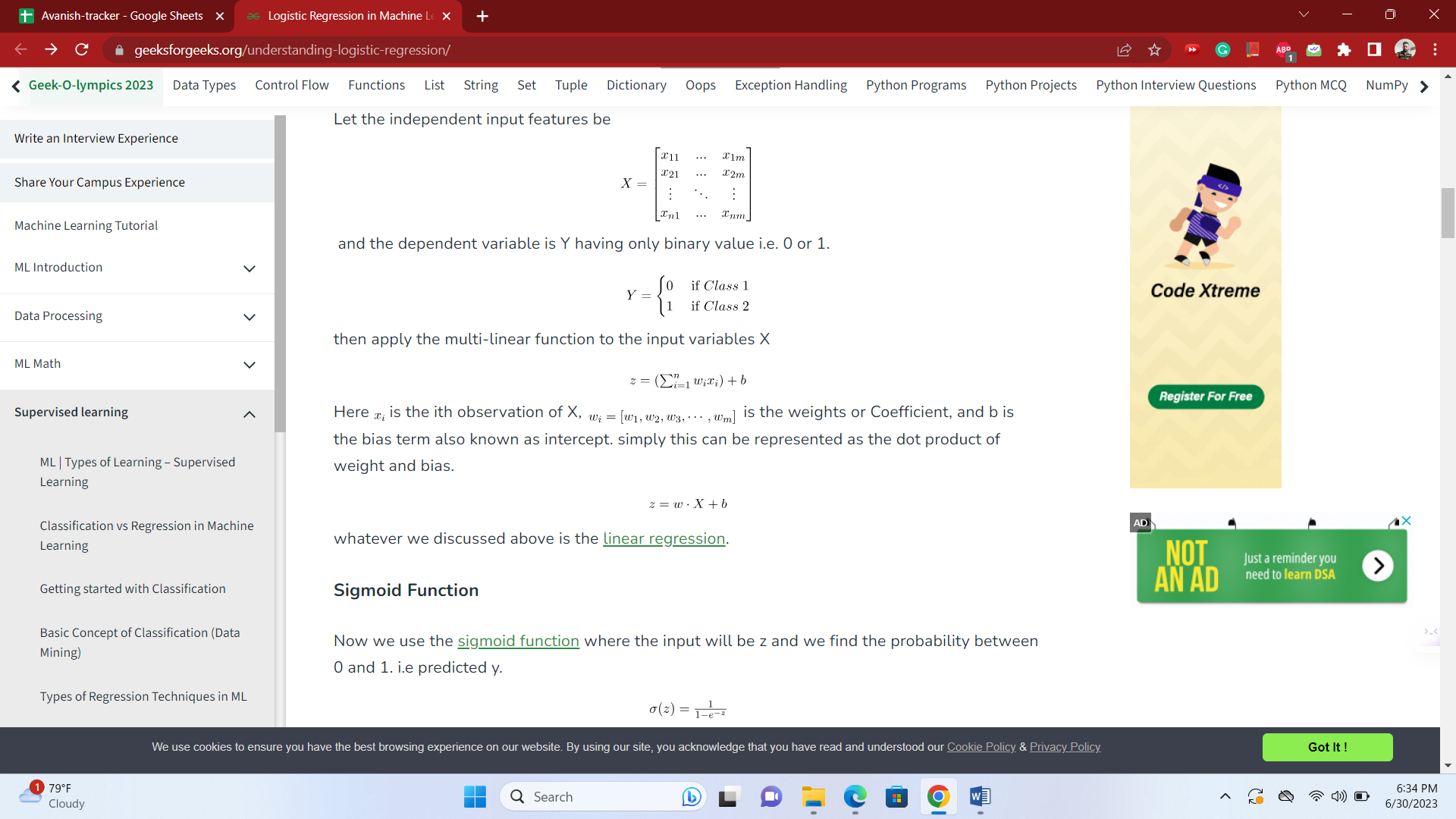
Here are some common terms involved in logistic regression:

* **Independent variables:** The input characteristics or predictor factors applied to the dependent variable’s predictions.
* **Dependent variable:** The target variable in a logistic regression model, which we are trying to predict.
* **Logistic function:** The formula used to represent how the independent and dependent variables relate to one another. The logistic function transforms the input variables into a probability value between 0 and 1, which represents the likelihood of the dependent variable being 1 or 0.
* **Odds:**It is the ratio of something occurring to something not occurring. it is different from probability as the probability is the ratio of something occurring to everything that could possibly occur.
* **Log-odds:**The log-odds, also known as the logit function, is the natural logarithm of the odds. In logistic regression, the log odds of the dependent variable are modeled as a linear combination of the independent variables and the intercept.
* **Coefficient:**The logistic regression model’s estimated parameters, show how the independent and dependent variables relate to one another.
* **Intercept:**A constant term in the logistic regression model, which represents the log odds when all independent variables are equal to zero.
* [**Maximum likelihood estimation**](https://www.geeksforgeeks.org/probability-density-estimation-maximum-likelihood-estimation/)**:** The method used to estimate the coefficients of the logistic regression model, which maximizes the likelihood of observing the data given the model.

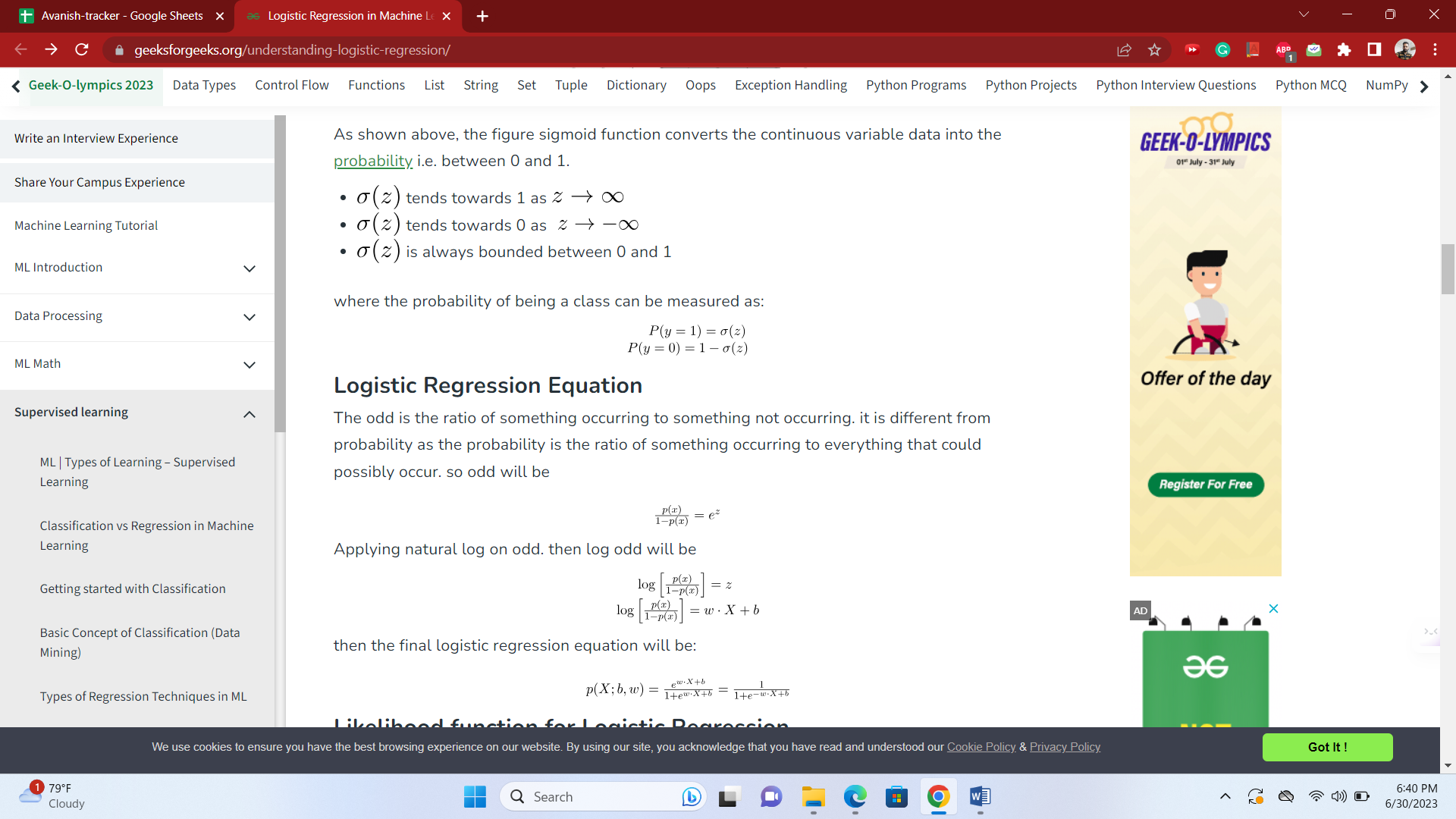
**How does Logistic Regression work?**

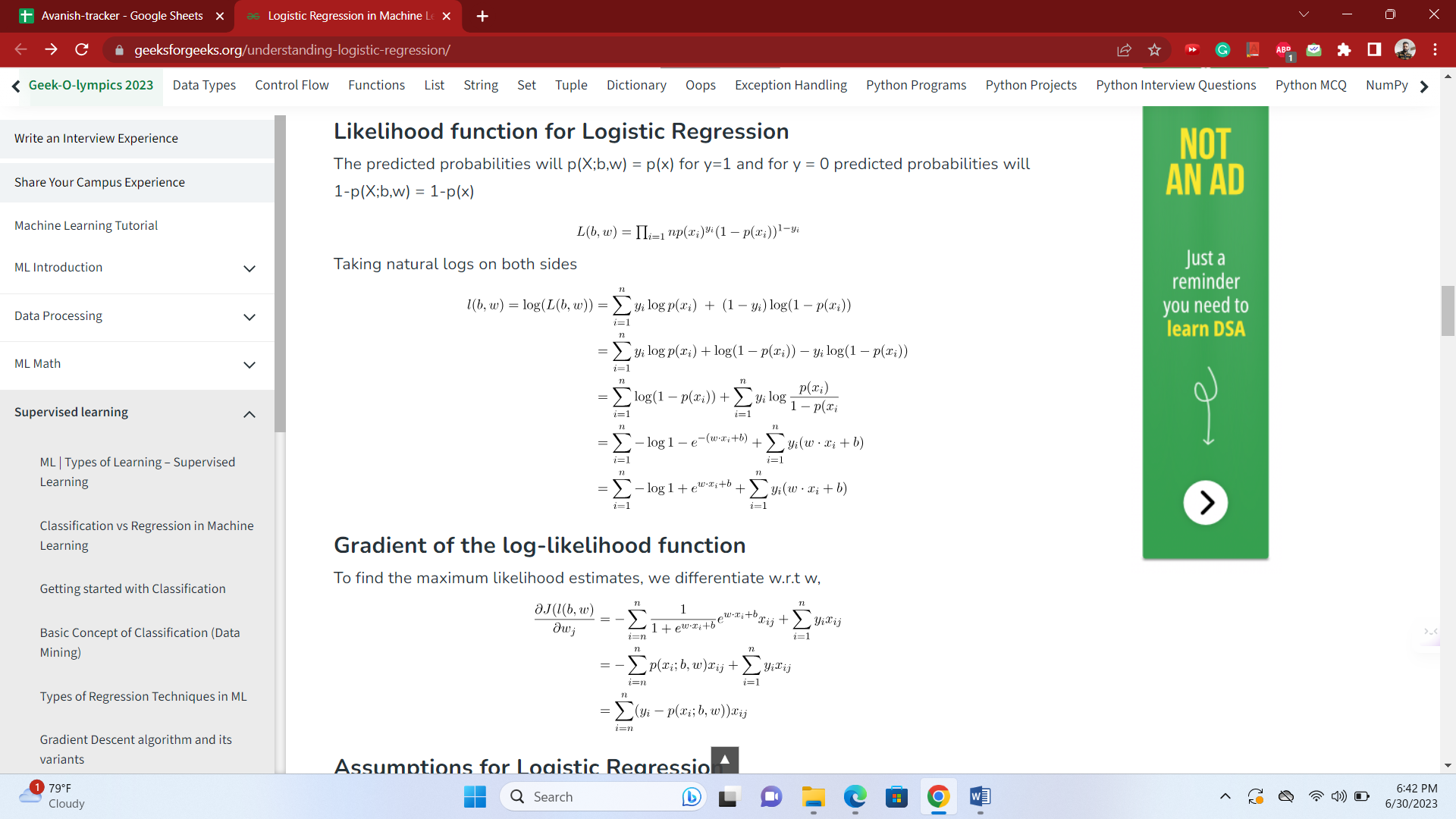
The logistic regression model transforms the [linear regression](https://www.geeksforgeeks.org/ml-linear-regression/) function continuous value output into categorical value output using a sigmoid function, which maps any real-valued set of independent variables input into a value between 0 and 1. This function is known as the logistic function.











## Assumptions for Logistic Regression

The assumptions for Logistic regression are as follows:

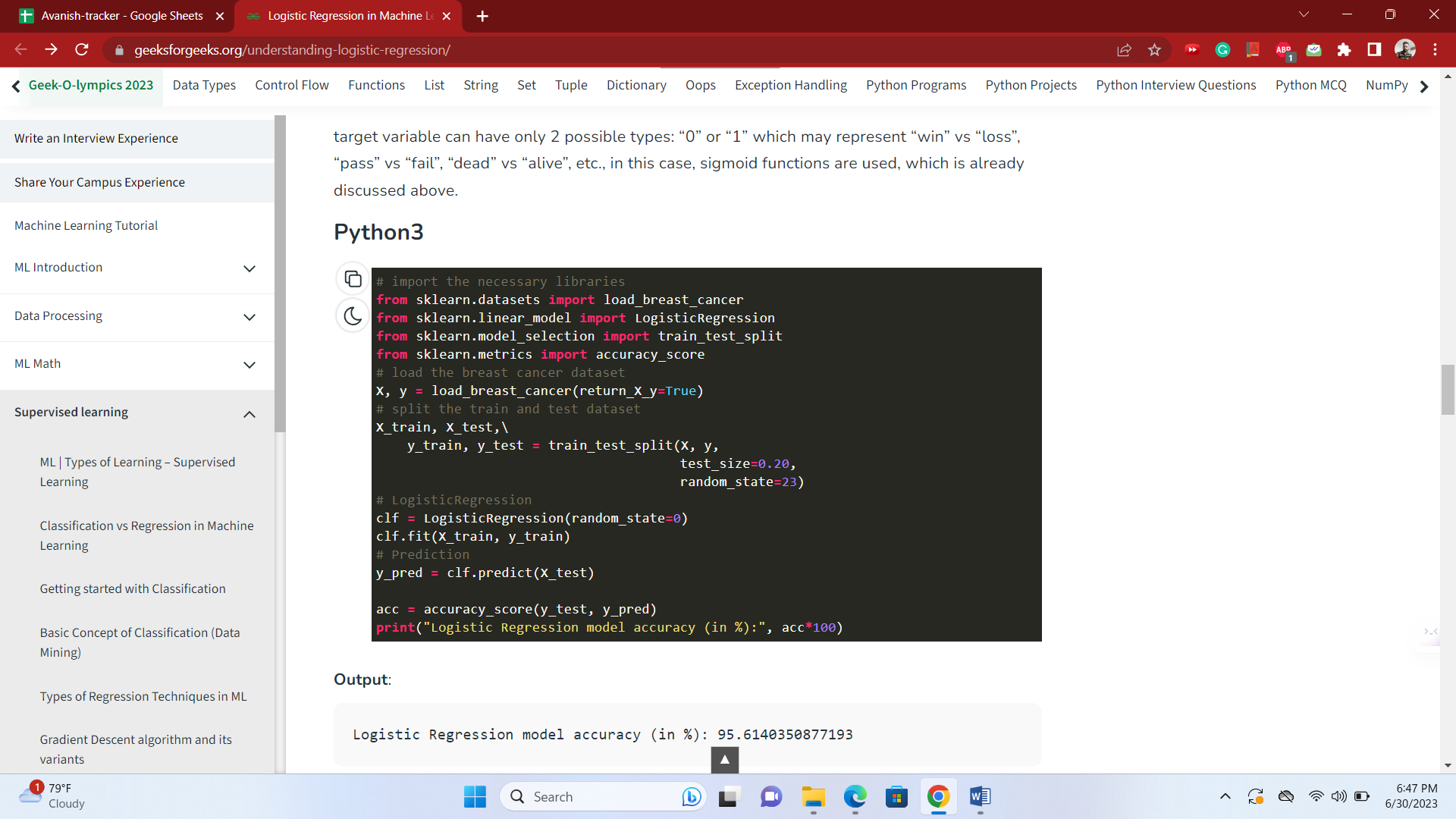
* **Independent observations:**Each observation is independent of the other. meaning there is no correlation between any input variables.
* **Binary dependent variables:**It takes the assumption that the dependent variable must be binary or dichotomous, meaning it can take only two values. For more than two categories softmax functions are used.
* **Linearity relationship between independent variables and log odds:** The relationship between the independent variables and the log odds of the dependent variable should be linear.
* **No outliers:** There should be no outliers in the dataset.
* **Large sample size:**The sample size is sufficiently large

## Types of Logistic Regression

Based on the number of categories, Logistic regression can be classified as:

### ****Binomial Logistic regression:****

target variable can have only 2 possible types: “0” or “1” which may represent “win” vs “loss”, “pass” vs “fail”, “dead” vs “alive”, etc., in this case, sigmoid functions are used, which is already discussed above.

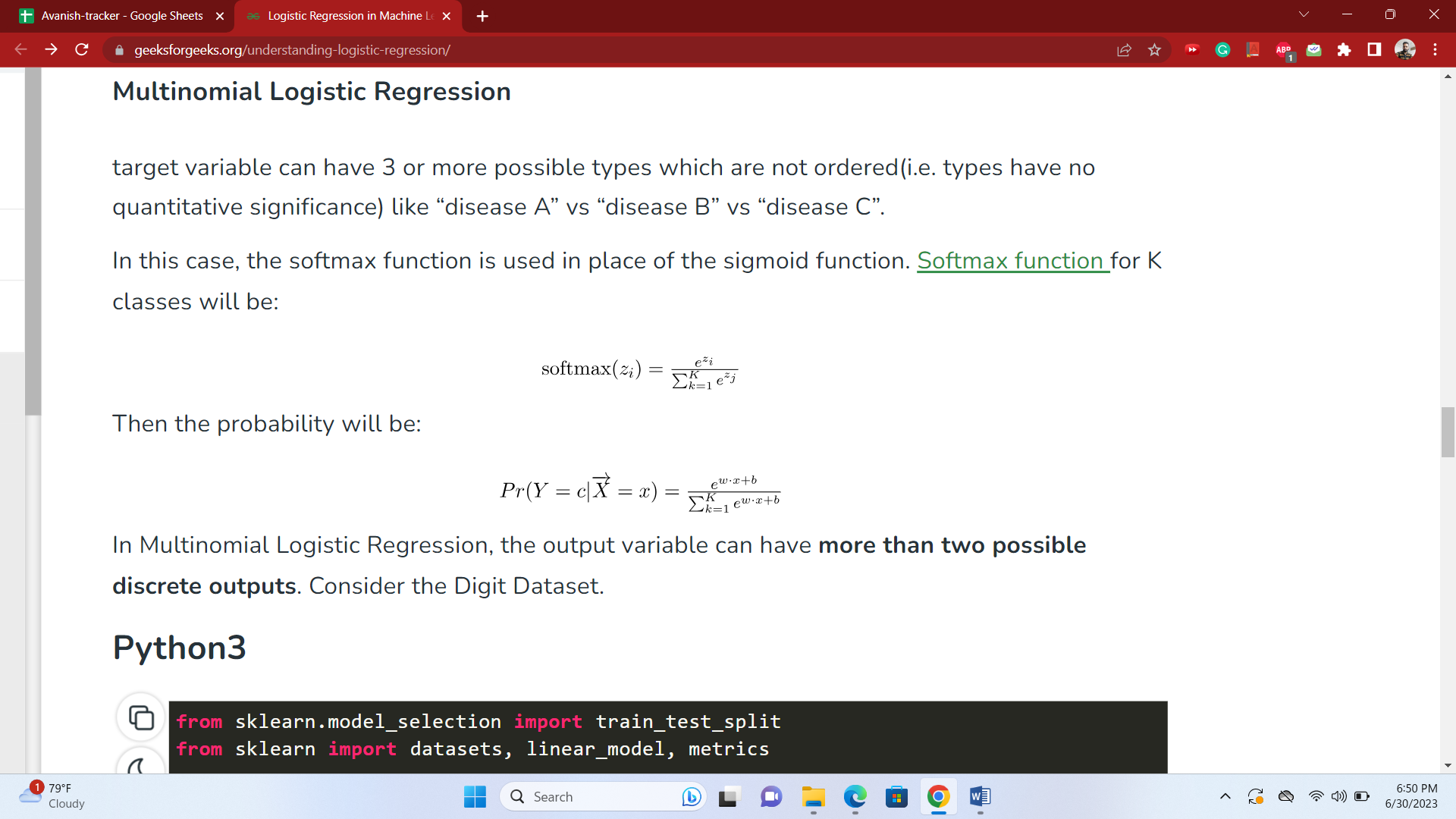


Logistic Regression model accuracy (in %): 95.6140350877193

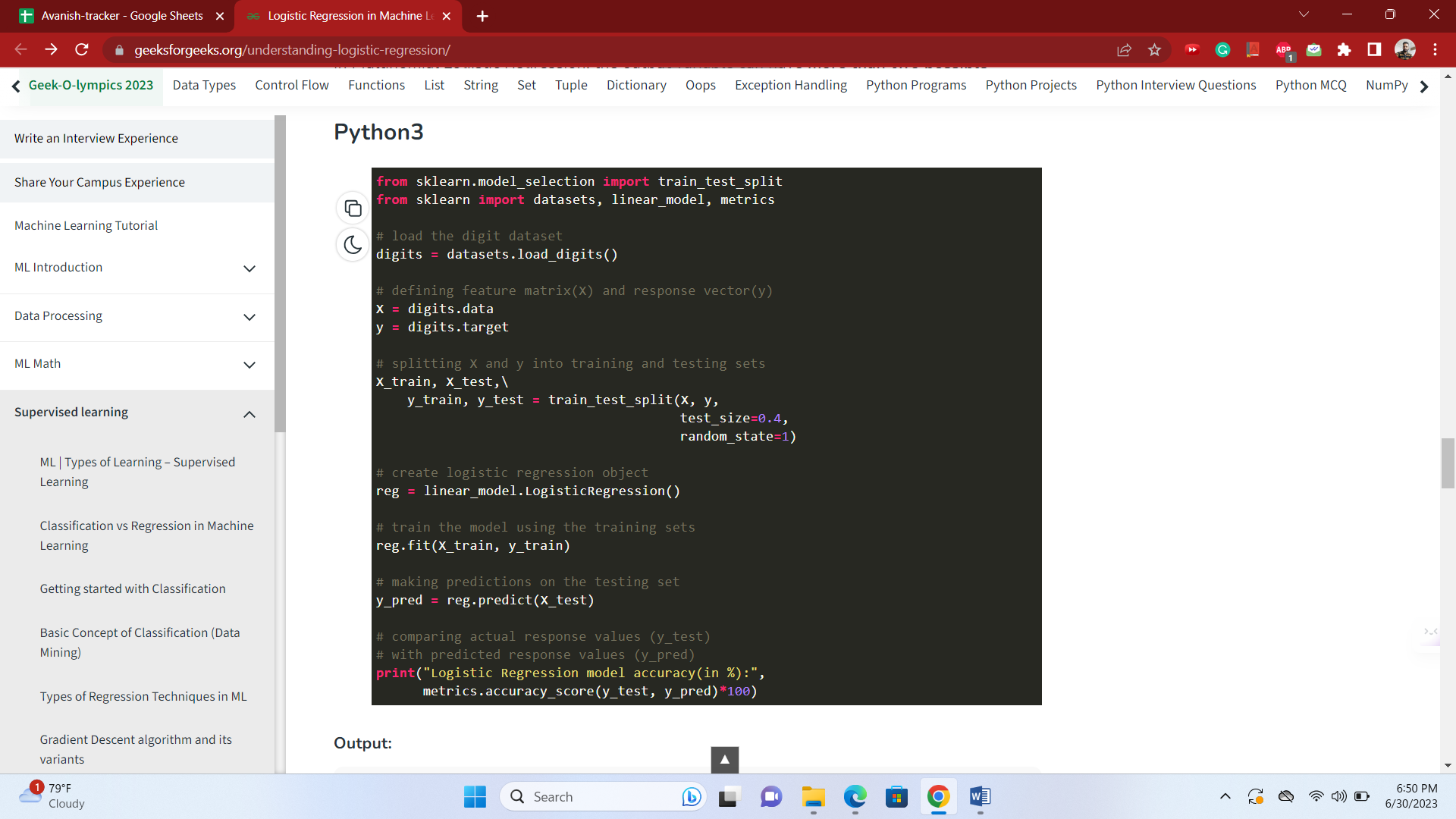
### ****Multinomial Logistic Regression****

target variable can have 3 or more possible types which are not ordered(i.e. types have no quantitative significance) like “disease A” vs “disease B” vs “disease C”.

In this case, the softmax function is used in place of the sigmoid function. [Softmax function](https://www.geeksforgeeks.org/understanding-activation-functions-in-depth/)for K classes will be:



In Multinomial Logistic Regression, the output variable can have **more than two possible discrete outputs**. Consider the Digit Dataset.



Logistic Regression model accuracy(in %): 96.52294853963839

**Loss function and cost function**

When calculating loss we consider only a single data point, then we use the term loss function.

Whereas, when calculating the sum of error for multiple data then we use the cost function. There is no major difference.

In other words, the loss function is to capture the difference between the actual and predicted values for a single record whereas cost functions aggregate the difference for the entire training dataset.

The Most commonly used loss functions are Mean-squared error and Hinge loss.

Mean-Squared Error(MSE): In simple words, we can say how our model predicted values against the actual values.

MSE = √(predicted value - actual value)2  
Hinge loss: It is used to train the machine learning classifier, which is

L(y) = max(0,1- yy)

Where y = -1 or 1 indicating two classes and y represents the output form of the classifier. The most common cost function represents the total cost as the sum of the fixed costs and the variable costs in the equation y = mx + b

<https://www.analyticsvidhya.com/blog/2020/11/binary-cross-entropy-aka-log-loss-the-cost-function-used-in-logistic-regression/>

The terms "cost function" and "loss function" are often used interchangeably in the context of machine learning, but they can have slightly different interpretations depending on the specific context. In general, both terms refer to a mathematical function that measures the error or discrepancy between predicted and actual values. However, there can be subtle differences in their usage.

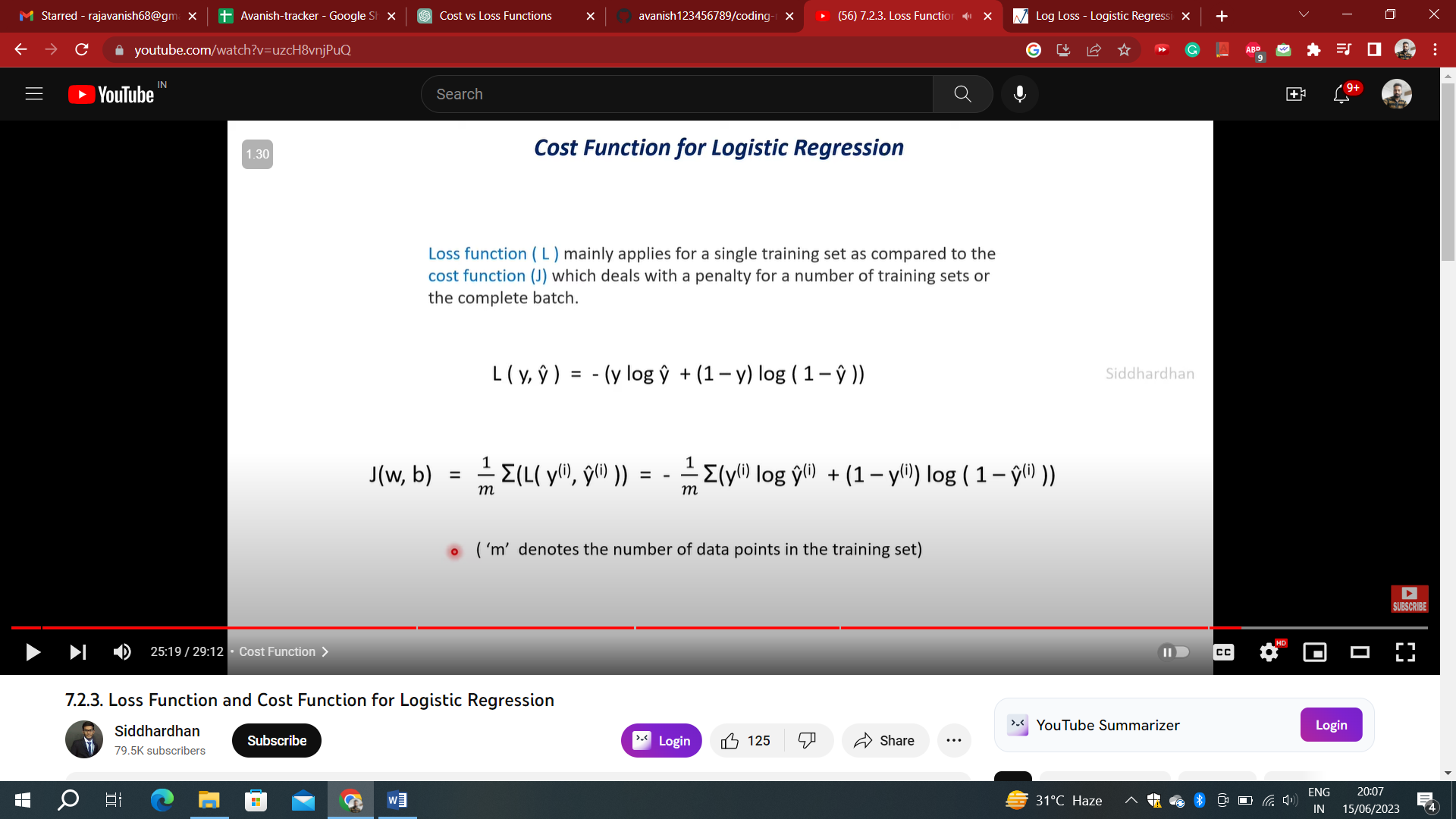
Loss Function: A loss function, also known as an error function or objective function, quantifies the error between the predicted output of a machine learning model and the true output or target value. It measures how well the model is performing on a specific training example. The purpose of the loss function is to provide a single scalar value that represents the error, which can be used to optimize the model's parameters during the training process. The goal is to minimize this error to improve the model's performance.

Examples of loss functions include mean squared error (MSE), binary cross-entropy, categorical cross-entropy, and hinge loss. The choice of loss function depends on the problem at hand and the type of data being predicted.

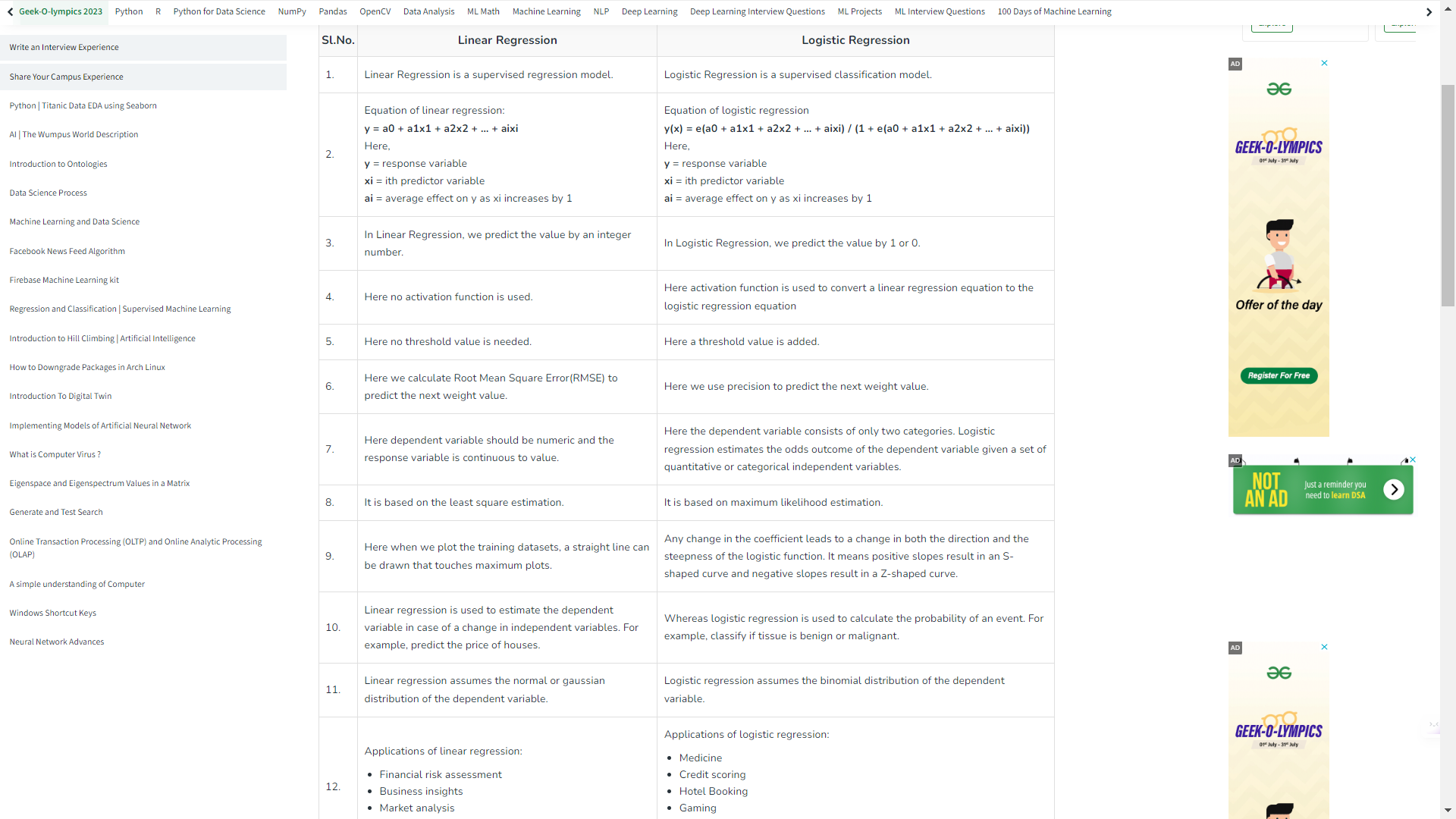
Cost Function: The term "cost function" is often used in the context of optimization algorithms, such as gradient descent, which aim to minimize the error or loss of a machine learning model. In this context, the cost function represents the average loss or error over the entire training dataset. It calculates the overall performance of the model by summing up or averaging the individual losses computed by the loss function across all training examples.

While the loss function focuses on the error for a single training example, the cost function provides a broader perspective by considering the average error across the entire dataset. By minimizing the cost function, the model aims to generalize well and make accurate predictions on unseen data.

In summary, the loss function measures the error for a single training example, while the cost function represents the overall performance of the model by aggregating the losses across the entire training dataset. However, in many cases, the terms "loss function" and "cost function" are used interchangeably, and the distinction between them may not be critical in practice.



# **Linear Regression vs Logistic Regression**



Top of Form

**How does logistic regression vary when datasets are large and small**

Logistic regression can exhibit different behaviors when applied to large and small datasets. Here are some ways in which logistic regression can vary based on the dataset size:

The sample size can have several effects on logistic regression:

1. Model Stability: In general, as the sample size increases, logistic regression estimates become more stable and reliable. With a larger sample size, the estimated coefficients tend to converge to the true population values, resulting in more accurate parameter estimates.
2. Generalization: A larger sample size provides a more representative and diverse set of observations from the underlying population. This improves the model's ability to generalize to new, unseen data. Models trained on larger datasets tend to have better predictive performance when applied to new samples.
3. Precision of Parameter Estimates: With a larger sample size, the standard errors associated with the estimated coefficients tend to decrease. This means that the parameter estimates become more precise, and the confidence intervals around them become narrower. As a result, statistical inferences made about the relationship between predictors and the outcome variable become more reliable.
4. Statistical Power: A larger sample size increases the statistical power of the logistic regression analysis. Statistical power refers to the ability of the model to detect true effects or relationships between predictors and the outcome variable. With more data, logistic regression can detect smaller, more subtle effects and provide more accurate estimates of the model parameters.
5. Overfitting: Overfitting occurs when a model learns the noise or idiosyncrasies of the training data, leading to poor generalization performance. In logistic regression, larger sample sizes can help mitigate overfitting by providing a more diverse set of observations, reducing the risk of fitting noise in the data.
6. Rare Events: Logistic regression is commonly used for modeling rare events or imbalanced datasets. In such cases, a larger sample size is advantageous as it increases the likelihood of capturing sufficient instances of the rare event, improving the model's ability to learn and predict the event accurately.

It's important to note that while a larger sample size generally provides benefits, there is a diminishing return effect. Once the sample size reaches a certain point, additional data may not significantly improve the model's performance or precision of parameter estimates. The optimal sample size depends on various factors, such as the complexity of the problem, the effect size, and the level of noise in the data.

In summary, a larger sample size enhances the stability, generalization, precision of parameter estimates, statistical power, and mitigates overfitting in logistic regression. Increasing the sample size can lead to more reliable and accurate models, improving the overall performance of logistic regression analysis.

Top of Form

Top of Form

Top of Form

**Logistic Regression sensitivity to outlier**

Logistic regression can be sensitive to outliers, just like other regression models. Outliers are data points that deviate significantly from the general pattern or trend in the dataset. Here's how outliers can impact logistic regression:

1. Influence on Parameter Estimates: Outliers can exert a strong influence on the estimated coefficients (parameters) of logistic regression. Logistic regression aims to find the best-fit line that maximizes the likelihood of the observed data. Outliers, being extreme observations, can pull the estimated line towards them, leading to biased coefficient estimates.
2. Impact on Decision Boundary: Logistic regression uses a decision boundary to separate the two classes in binary classification. Outliers can disrupt the decision boundary, causing it to shift or deviate from its intended position. This can lead to misclassification or decreased performance of the logistic regression model.
3. Distortion of Odds Ratios: In logistic regression, odds ratios are commonly used to interpret the impact of predictors on the outcome variable. Outliers can inflate or deflate the odds ratios, resulting in misleading interpretations of the predictor's effect on the outcome.
4. Influence on Model Fit: Outliers can affect the overall fit of the logistic regression model. The presence of outliers may lead to a poorer fit of the model to the majority of the data, reducing the model's ability to accurately represent the underlying relationship.
5. Increased Sensitivity to Overfitting: Logistic regression models can be more susceptible to overfitting in the presence of outliers. Overfitting occurs when the model captures noise or idiosyncrasies in the data, resulting in poor generalization to new data. Outliers can exacerbate overfitting by causing the model to excessively adjust to accommodate them.

To mitigate the sensitivity of logistic regression to outliers, several approaches can be employed:

1. Outlier Detection: Identify and analyze potential outliers in the dataset using techniques like box plots, scatter plots, or robust statistical methods. Carefully examine the outliers to determine if they are genuine data points or measurement errors.
2. Outlier Treatment: Depending on the nature and cause of the outliers, different approaches can be used, such as removing the outliers, transforming the outliers, or assigning them to a separate category. The appropriate treatment strategy should be based on the specific context and domain knowledge.
3. Robust Estimation: Robust regression techniques, such as robust logistic regression or M-estimators, can be employed to reduce the influence of outliers on the parameter estimates. These methods downweight the impact of outliers, leading to more robust and reliable model estimation.
4. Regularization: Regularization methods like L1 or L2 regularization (e.g., ridge regression or lasso) can help reduce the impact of outliers and prevent overfitting. These techniques introduce a penalty term that encourages simpler models and reduces the influence of extreme observations.
5. Stratified Sampling: In cases where outliers are suspected to be influential due to their class membership, stratified sampling can be used to ensure a balanced representation of the classes and mitigate the effect of outliers on the logistic regression model.

It's important to carefully evaluate the impact of outliers on logistic regression and consider appropriate strategies to handle them based on the specific characteristics of the dataset and the objectives of the analysis.

Top of Form

Top of Form

Top of Form

**Effect of missing values on Logistic Regression**

Missing values in logistic regression can have several effects on the analysis and model performance:

1. Reduced Sample Size: When missing values are present in the dataset, they lead to a reduction in the effective sample size available for analysis. This reduction in sample size can potentially limit the statistical power of the logistic regression model, making it more challenging to detect significant relationships between predictors and the outcome variable.
2. Potential Bias: If missingness is related to the outcome or predictors, it can introduce bias in the estimated coefficients. This bias occurs because the missing data may not be missing completely at random (MCAR) or missing at random (MAR), violating the assumptions of logistic regression. In such cases, the estimated coefficients may not accurately represent the true relationship between predictors and the outcome.
3. Incomplete Information: Missing values lead to incomplete information for specific observations. This can result in a loss of information and potentially affect the accuracy of the logistic regression model. The missingness may eliminate potentially valuable data that could contribute to the model's predictive power.
4. Imputation Techniques: To address missing values, imputation techniques can be employed to fill in or estimate the missing data. Different imputation methods, such as mean imputation, median imputation, or multiple imputation, can be used to replace missing values with estimated values. However, the choice of imputation method can introduce additional uncertainty and potential biases in the analysis.
5. Missing Data Mechanism: The impact of missing values also depends on the missing data mechanism. If the missing data mechanism is missing completely at random (MCAR), where the probability of missingness is unrelated to the observed or unobserved data, the impact on the logistic regression may be minimal. However, if the missing data mechanism is missing at random (MAR) or missing not at random (MNAR), where the missingness is related to the observed or unobserved data, the missing values can introduce bias and affect the model estimates.
6. Handling Missingness: Several approaches can be used to handle missing values in logistic regression. These include complete case analysis (excluding observations with missing values), imputation methods, or using specialized techniques like maximum likelihood estimation that can handle missing data directly. The choice of approach depends on the specific characteristics of the data and the assumptions made about the missingness mechanism.

It is crucial to carefully consider the implications of missing values in logistic regression and select appropriate methods to handle them. The choice of handling missingness should be guided by the missing data mechanism, the amount of missingness, and the potential impact on the model's validity and performance.

**Effect of correlation on Logistic Regression**

Correlation among predictors (also known as multicollinearity) can have several effects on logistic regression:

1. Unreliable Parameter Estimates: High correlation among predictors can make it challenging to estimate the individual effects of each predictor accurately. When predictors are highly correlated, the logistic regression model may struggle to disentangle their individual contributions to the outcome variable. This can lead to unstable and unreliable parameter estimates.
2. Impact on Variable Importance: Correlation among predictors can make it difficult to determine the relative importance or significance of each predictor in the logistic regression model. When predictors are highly correlated, it becomes challenging to attribute the observed effects solely to one predictor while accounting for the influence of other correlated predictors. This can affect the interpretation of variable importance in the model.
3. Increased Standard Errors: High correlation among predictors leads to increased standard errors of the estimated coefficients. Larger standard errors indicate higher uncertainty in the estimated effects of predictors on the outcome variable. This can make it harder to detect statistically significant relationships and may reduce the precision of the parameter estimates.
4. Unstable Model Performance: Multicollinearity can make the logistic regression model more sensitive to small changes in the data. As the correlation among predictors increases, slight changes in the input data can lead to substantial changes in the model's predicted probabilities or classification results. This instability can hinder the model's performance and make it less reliable for prediction.
5. Inflated Variance Inflation Factor (VIF): The variance inflation factor (VIF) is a measure of multicollinearity that quantifies the extent to which the variance of the estimated regression coefficients is inflated due to correlation among predictors. High VIF values indicate strong correlation among predictors, which can further compromise the reliability of the logistic regression model.

To address the effects of correlation on logistic regression, several strategies can be employed:

1. Variable Selection: If predictors are highly correlated, it may be necessary to select a subset of variables that are most relevant to the outcome. Prioritize variables that have stronger theoretical or empirical justification for their inclusion and are less correlated with other predictors.
2. Feature Engineering: Transform or combine correlated predictors to create new variables that capture the underlying information without the high correlation. For example, you can create interaction terms, polynomial features, or aggregated variables to represent the relationship between correlated predictors.
3. Regularization: Regularization techniques, such as ridge regression or LASSO (Least Absolute Shrinkage and Selection Operator), can help reduce the impact of multicollinearity by shrinking the coefficients towards zero or enforcing sparsity in the model. Regularization can effectively handle correlated predictors and improve the stability and performance of the logistic regression model.
4. Prioritize Interpretability: When correlation is present among predictors, it becomes important to prioritize the interpretability of the logistic regression model. Selecting a subset of predictors with low correlation and high interpretability can help in understanding the relationships between predictors and the outcome variable more accurately.

In summary, high correlation among predictors in logistic regression can lead to unreliable parameter estimates, increased standard errors, unstable model performance, and inflated VIF values. By carefully selecting variables, performing feature engineering, using regularization techniques, and prioritizing interpretability, the impact of correlation can be mitigated, leading to more reliable and interpretable logistic regression models.

**Feature Engineering, Feature Selection and Feature Importance for Logistic Regression algorithm**

Feature Engineering, Feature Selection, and Feature Importance are important steps in preparing data for logistic regression. Here's how each of these steps can be applied to the logistic regression algorithm:

1. Feature Engineering:
   * Creating Interaction Terms: Generate new features by combining existing features through multiplication, division, or other mathematical operations. This can capture potential synergistic effects or nonlinear relationships between predictors.
   * Polynomial Features: Create higher-order polynomial features to capture nonlinearities in the data. For example, adding squared or cubed terms of predictors.
   * Binning or Discretization: Convert continuous features into categorical features by dividing them into bins or discrete intervals. This can capture nonlinear patterns that may exist in the data.
   * One-Hot Encoding: Convert categorical variables into binary indicator variables to represent different categories.
   * Log Transformations: Apply logarithmic transformations to skewed features to make the distribution more symmetric and reduce the influence of extreme values.
2. Feature Selection:
   * Univariate Selection: Use statistical tests (e.g., chi-square, t-tests) or correlation analysis to select features that are individually predictive of the outcome variable.
   * Recursive Feature Elimination: Recursively eliminate less important features based on their coefficients or importance until the desired number of features is reached.
   * L1 Regularization (Lasso): Utilize L1 regularization, which shrinks coefficients towards zero and performs automatic feature selection by effectively setting some coefficients to zero.
   * Information Gain or Mutual Information: Apply these techniques from information theory to measure the relevance of features by quantifying their association with the target variable.
3. Feature Importance:
   * Coefficient Magnitude: In logistic regression, the absolute magnitude of the estimated coefficients indicates the importance of the corresponding features. Larger coefficient values suggest stronger relationships with the outcome variable.
   * Odds Ratio: Compute the odds ratio for each predictor, which quantifies the change in odds of the outcome variable for a unit change in the predictor. Higher odds ratios indicate more influential features.
   * Stability Selection: Assess the stability of feature selection by performing logistic regression with bootstrapped samples and measuring the frequency of feature selection across multiple iterations.
   * Random Forest or Gradient Boosting: Use ensemble-based methods like random forests or gradient boosting, which inherently provide a measure of feature importance based on how often a feature is used for splitting in the tree-based models.

It's important to note that the choice and application of these techniques depend on the specific dataset, domain knowledge, and the problem at hand. Additionally, it is advisable to validate the effectiveness of feature engineering, selection, and importance techniques through proper validation procedures such as cross-validation or holdout validation.

**Overfitting handling in Logistic Regression**

Handling overfitting in logistic regression is similar to handling it in linear regression. Logistic regression is a classification algorithm, but it shares some common strategies with linear regression for addressing overfitting. Here's how you can handle overfitting in logistic regression:

1. **Feature Selection:**
   * Choose relevant features: Include only the most relevant features in your model to avoid fitting noise. Removing irrelevant or redundant features can prevent overfitting.
2. **Regularization Techniques:**
   * Ridge (L2 regularization), Lasso (L1 regularization), and Elastic Net: These techniques add penalty terms to the logistic regression cost function, discouraging large coefficients and feature selection, similar to linear regression.
3. **Cross-Validation:**
   * Use k-fold cross-validation to assess the model's generalization performance. This helps in selecting hyperparameters and detecting overfitting by evaluating the model on unseen data.
4. **Early Stopping:**
   * Monitor the model's performance on a validation set during training and stop training when validation performance starts to degrade.
5. **Data Augmentation:**
   * Generating augmented data by adding noise or variations to the existing data can help the model generalize better.
6. **Reduce Model Complexity:**
   * Limit the number of features, interactions, or polynomial terms in the model to avoid unnecessary complexity that might lead to overfitting.
7. **Regularization Strength Tuning:**
   * Adjust the strength of regularization (lambda or alpha parameter) using techniques like grid search or random search to find the right balance between fitting the data and preventing overfitting.
8. **Collect More Data:**
   * Increasing the size of your training dataset can help improve the model's generalization and reduce overfitting.
9. **Ensemble Methods:**
   * Consider using ensemble techniques like Random Forest or Gradient Boosting, which can mitigate overfitting by combining multiple models.
10. **Feature Engineering:**
    * Transform and engineer features in a way that reduces noise and captures relevant patterns, helping the model to focus on meaningful information.
11. **Regularization-Specific Considerations:**
    * For logistic regression, L1 regularization (Lasso) can also perform feature selection by driving some coefficients to exactly zero. This can help create a more parsimonious model.

Remember that the specific approach you take will depend on the characteristics of your data, the degree of overfitting observed, and the overall goals of your analysis. Regular evaluation of the model's performance on new data and careful tuning of hyperparameters are essential to effectively handle overfitting in logistic regression.

Top of Form