Assignment- Logistic Regression (Theretical)

#Q1. What is Logistic Regression, and how does it differ from Linear Regression?
Logistic Regression is a classification algorithm that predicts the probability of an instance belonging to a particular class.
It differs from Linear Regression as it outputs probabilities (bounded between 0 and 1) instead of continuous values.

#Q2. What is the mathematical equation of Logistic Regression?

The mathematical equation of Logistic Regression is based on the sigmoid function, which maps any real-valued number to a probability between 0 and 1.

- $P(Y=1|X) = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\ldots+\beta_n X_n)}}$

#Q3. Why do we use the Sigmoid function in Logistic Regression

The **Sigmoid function** (also called the **logistic function**) is used in Logistic Regression because it converts any real-valued input into a probability between **0 and 1**. This makes it ideal for **binary classification problems**, where we need to determine whether an instance belongs to class **1 (positive)** or **0 (negative)**.

#Q4. What is the cost function of Logistic Regression

In Logistic Regression, we use the **Log Loss (Logarithmic Loss)** or **Binary Cross-Entropy** as the cost function instead of Mean Squared Error (MSE) used in Linear Regression. This is because MSE is not suitable for classification problems, as it results in a **non-convex** function, making optimization difficult.

#Q5. What is Regularization in Logistic Regression?
Regularization is a technique used to **prevent overfitting** by adding a penalty term to the cost function. It discourages large coefficients, making the model more **generalizable**.

#Q6 Why is it needed Explain the difference between Lasso, Ridge, and Elastic Net regression

Types of Regularization:

1. **Lasso (L1 Regularization)** – Shrinks coefficients and forces some to become **zero** (feature selection).
2. **Ridge (L2 Regularization)** – Shrinks coefficients but **does not** force them to zero.
3. **Elastic Net (L1 + L2)** – A combination of **Lasso and Ridge**, balancing feature selection and coefficient shrinkage.

#Q7. When should we use Elastic Net instead of Lasso or Ridge

Elastic Net is preferred when:

- Features are **highly correlated**: Lasso may randomly select one feature and ignore others, but Elastic Net keeps a balance.
- Data is **high-dimensional** (many more features than samples): It combines L1 (Lasso) and L2 (Ridge) penalties to provide stability.
- You need **automatic feature selection** (Lasso-like effect) while preventing **over-shrinking** of coefficients (Ridge-like effect).

#Q8. What is the impact of the regularization parameter (λ) in Logistic Regression

The regularization parameter λ\lambdaλ (or C=1λC = \frac{1}{\lambda}C=λ1 in scikit-learn) controls the complexity of the model:

- **Higher λ** → Stronger penalty → Simpler model (less overfitting) → Smaller coefficients.
- **Lower λ** → Weaker penalty → More flexible model (but risk of overfitting) → Larger coefficients.

#Q9. What are the key assumptions of Logistic Regression?

1. **Linear Relationship (with Logit):** The independent variables should have a linear relationship with the log-odds.
2. **No Multicollinearity:** Features should not be highly correlated; otherwise, it affects coefficient stability.
3. **Independent Observations:** Data points should be independent of each other.
4. **Large Sample Size:** Logistic Regression performs best when there are enough samples, especially for rare classes.

#Q10. What are some alternatives to Logistic Regression for classification tasks?

- **Decision Trees** (handles non-linearity well).
- **Random Forests** (ensemble learning, reduces overfitting).
- **Support Vector Machines (SVMs)** (works well with high-dimensional data).
- **Naive Bayes** (good for text classification and probabilistic modeling).
- **Neural Networks** (for complex, non-linear relationships).
- **Gradient Boosting (XGBoost, LightGBM, CatBoost)** (for structured data).

#Q11. What are Classification Evaluation Metrics?

- **Accuracy** = (TP + TN) / (TP + TN + FP + FN)
- **Precision** = TP / (TP + FP) (how many predicted positives are correct).
- **Recall (Sensitivity)** = TP / (TP + FN) (how many actual positives are identified).
- **F1-score** = 2 × (Precision × Recall) / (Precision + Recall) (harmonic mean of precision and recall).
- **ROC-AUC Score** (discrimination between classes).
- **Log Loss** (penalizes wrong probability estimates).

#Q12. How does class imbalance affect Logistic Regression?

- Logistic Regression may **favor the majority class**, predicting the minority class poorly.
- Metrics like **accuracy become misleading**.
- **Solutions:**
  - **Resampling techniques** (oversampling minority or undersampling majority class).
  - **Use class weights** (`class_weight='balanced'` in sklearn).
  - **Use different metrics** like Precision-Recall AUC instead of accuracy.

#Q13. What is Hyperparameter Tuning in Logistic Regression?

Hyperparameter tuning optimizes model performance. Key parameters:

- **Regularization parameter (λ or C)**: Controls overfitting.
- **Penalty type (L1, L2, Elastic Net)**: Chooses the right regularization method.
- **Solver selection** (affects optimization performance).
- **Grid Search / Random Search**: Common tuning methods.

#Q14. What are different solvers in Logistic Regression? Which one should be used?

- **liblinear**: Good for small datasets, supports L1 & L2.
- **lbfgs**: Default for small to medium datasets, supports only L2.
- **saga**: Good for large datasets, supports L1, L2, and Elastic Net.
- **newton-cg** & **cg**: Works well for only L2 regularization.

Which one to use?

- For **small datasets** → `liblinear`.
- For **large datasets** → `saga`.
- For **L2 regularization** → `lbfgs` or `newton-cg`.

#Q15. How is Logistic Regression extended for multiclass classification?

- **One-vs-Rest (OvR):** Fits separate binary classifiers for each class.
- **Softmax (Multinomial Regression):** Assigns probabilities to multiple classes simultaneously.
- **Which one to use?**
  - **OvR**: Works well with fewer classes, interpretable.
  - **Softmax**: Preferred when classes are mutually exclusive.

#Q16. What are the advantages and disadvantages of Logistic Regression?

Advantages:

- Simple, interpretable, and efficient.
- Works well with linearly separable data.
- Probabilistic output (confidence scores).
- Less prone to overfitting with proper regularization.

Disadvantages:

- Struggles with non-linear relationships.
- Sensitive to outliers.
- Requires feature scaling for best performance.

#Q17. What are some use cases of Logistic Regression?

- **Healthcare**: Disease prediction (e.g., diabetes, cancer detection).
- **Finance**: Credit risk analysis, fraud detection.
- **Marketing**: Customer churn prediction.
- **Human Resources**: Employee attrition prediction.
- **Political Science**: Predicting election outcomes.

#Q18. What is the difference between Softmax Regression and Logistic Regression?

- **Logistic Regression**: Used for **binary classification**, predicts probability for two classes.
- **Softmax Regression**: Generalization of logistic regression for **multiclass classification**, assigns probability to each class.

#Q19. How do we choose between One-vs-Rest (OvR) and Softmax for multiclass classification?

Use OvR when:

- The number of classes is large.
- Faster training time.

Use Softmax when:

- Classes are mutually exclusive.
- Probabilistic interpretation is needed.

#Q20. How do we interpret coefficients in Logistic Regression?

Each coefficient βi\beta_iβi represents the change in log-odds for a one-unit increase in the corresponding feature.
$e\beta_i$ gives the odds ratio (how much more/less likely an event is).
If $\beta_i>0$, the feature increases the probability of the positive class.
If $\beta_i<0$, the feature decreases the probability of the positive class.