📘 Logistic Regression — Revision Notes

🔹 1. Role of Logistic Regression

- Acts as a **classifier** (Depressed vs Not Depressed).
- Learns **weights (coefficients)** for each feature (word/phrase).
- Uses **sigmoid function** to convert linear sum → probability.
- Decision rule:
  - P≥0.5P ≥ 0.5P≥0.5 → Class 1 (Depressed)
  - P<0.5P < 0.5P<0.5 → Class 0 (Not Depressed)

🔹 2. Pipeline

1. **Dataset** → sentences + labels (0/1).
2. **TF-IDF Vectorization** → text → numbers (features).
   - Rare, meaningful words → higher score.
   - Common words → lower score.
3. **Logistic Regression Training**

3. Logistic Regression Training

- Equation:

$$z = w_1x_1 + w_2x_2 + \ldots + b$$

- Apply sigmoid:

$$P(y = 1|x) = \frac{1}{1 + e^{-z}}$$

- Learns coefficients w_i.

4. **Prediction**
   - Compute probability.
   - Apply threshold (0.5).

🔹 3. Example

Sentence: `"I feel sad and empty"`

- TF-IDF vector → `[sad=0.7, empty=0.6]`
- Weights → `[sad=2.1, empty=1.8]`
- Linear score:

$$z = 2.1(0.7) + 1.8(0.6) + b = 2.55 + b$$

- Sigmoid:

$$P \approx 0.95$$

- Predict → **Depressed (1)**

↓

🔹 4. Visualization Insights

- **Sigmoid curve (S-shape):**
  - Small z → P near 0
  - Large z → P near 1
- **Word "sad"** → probability ↑ (Depressed side).
- **Word "happy"** → probability ↓ (Not Depressed side).
- Logistic regression learns **positive vs negative signals** from words.

🔹 5. Key Points to Remember

- Logistic regression = **linear model + sigmoid function**.
- It doesn't just say "yes/no" → it gives **probability**.
- Works well with **TF-IDF features**.
- Easy to interpret (weights show importance of words).
- Used as a **baseline model** in NLP before trying SVM, Naive Bayes, or deep learning.

🔹 Linear Combination (z) = $w_1x_1 + w_2x_2 + \ldots + b$

1) x_i (features / values) kahan se aate hain?

- Ye tumhare **sentence ke TF-IDF vector** se aate hain.
- Example sentence: `"I feel sad and empty"`

Vocabulary = 'sad', '$empty$', '$amazing$', '$grateful$', '$anxious$', '$sleep$'

TF-IDF vector:

- `"sad"` word sentence me hai aur important hai ⇒ TF-IDF score = 0.7
- `"empty"` bhi hai ⇒ TF-IDF score = 0.6
- `"amazing"`, `"grateful"` nahi hain ⇒ TF-IDF = 0.0

👉 Matlab:

x=[sad=0.7,empty=0.6,amazing=0.0,grateful=0.0,anxious=0.0,sleep=.0]

2) w_i (weights) kahan se aate hain?

- Ye **Logistic Regression training ke dauraan sikhta hai**.
- Jab tum `log_reg.fit(X_train_tfidf, y_train)` chalate ho:
  - Model ko ground-truth labels diye jaate hain (Depressed = 1, Not Depressed = 0).
  - Model har word ke liye ek coefficient (weight) learn karta hai jo batata hai us word ka **Depression vs Not Depressed me kitna contribution hai**.

Example (maan lo model ne sikha):

- `sad` → +2.1 (Depressed ki taraf push karta hai)
- `empty` → +1.8 (Depressed ke signal deta hai)
- `amazing` → −2.0 (Not Depressed ke sig deta hai)
- `grateful` → −1.7 (Not Depressed ke liye strong word)

3) Bias (b) kya hai?

- Ye ek **extra constant** hota hai jo model ke decision boundary ko adjust karta hai.
- Matlab agar sare words zero ho tab bhi model ek base probability predict kare

Example

Sentence: `"I feel sad and empty"`

Vector (x):

- sad = 0.7, empty = 0.6, baki sab 0.

Weights (w):

- sad = 2.1, empty = 1.8

Calculation:

$$z = (2.1 \times 0.7) + (1.8 \times 0.6) + b$$

$$z = 1.47 + 1.08 + b = 2.55 + b$$

↓

👉 Yehi tumne dekha tha as `≈ 2.97 + b` (agar TF-IDF ya weights thoda alag hote).

📌 **Summary**

- xi_ = **sentence ka TF-IDF score** for word *i*.
- wi = **Logistic Regression ne training se sikha hua coefficient** for word *i*.
- Bias = ek constant adjustment.
- Saath milke → **linear sum (z)** banta hai → sigmoid → probability.

🔹 Step 4: Sigmoid Function

Formula:

$$P(y=1|x) = \frac{1}{1+e^{-z}}$$

- **Input** = linear sum $z = w_1x_1 + w_2x_2 + \ldots + b$
- **Output** = probability (always between 0 and 1)

🔹 Example Calculation

Suppose `"I feel sad and empty"` sentence ke liye:

- Calculated linear sum $z = 3.0$

Now:

$$P(y=1|x) = \frac{1}{1+e^{-3.0}}$$

$$= \frac{1}{1+0.0498} = \frac{1}{1.0498} \approx 0.95$$

🔹 Interpretation

- $P = 0.95$ → **95% chance sentence is Depressed (class 1)**
- Decision rule:
  - If P ≥ 0.5 → Predict 1 (Depressed)
  - If P < 0.5 → Predict 0 (Not Depressed)

🔹 Sigmoid Intuition

- Jab **z** bahut bada positive (≫ 0) ho → P → 1 (almost surely Depressed)
- Jab **z** bahut bada negative (≪ 0) ho → P → 0 (almost surely Not Depressed)
- Jab **z = 0** ho → P = 0.5 (model unsure).