Lecture-5

- Regression vs Classification

Feature	Classification	Regression
Task	Predicts a discrete label from input variables (X)	Predicts a continuous value from input variables (X)
Output Variable	Discrete (categorical values like "spam" or "not spam")	Continuous (numerical values like price or size)
Example	Classifying emails as "spam" or "not spam"	Predicting house prices between $100,000 and $200,000
Problem Types	- Two-class (binary) classification - Multi-class classification - Multi-label classification	- Multivariate regression - Time-series forecasting

- Linear vs Logistic

Linear Regression	Logistic Regression
Regression problem (predicts continuous output)	Classification problem (predicts discrete output)
Continuous (real-valued, e.g., house prices, weight)	Discrete (binary or categorical, e.g., "spam" or "not spam")
Models a linear relationship between input features and output	Models the probability of a categorical outcome using a logistic function
$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n$	$P(y=1) = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\ldots+\beta_n X_n)}}$
Mean Squared Error (MSE)	Log-Loss or Cross-Entropy
Exact values (e.g., $200,000 for house prices)	Probabilities that convert to binary outcomes (e.g., 0 or 1 for classification)
Predicting a person's weight based on height and age	Predicting whether an email is "spam" or "not spam"

- Using linear regression for classification is problematic due to the following reasons:
    1. **Output Range**: Linear regression produces continuous values, which can be negative or greater than one, whereas classification requires discrete class labels (e.g., 0 or 1).
    2. **Interpretation**: The outputs of linear regression are not easily interpretable as probabilities for class membership, making it difficult to determine which class an observation belongs to.
    3. **Decision Boundary**: Linear regression creates a linear decision boundary that may lead to outputs falling outside the valid class range, while classification models are designed to provide clear boundaries.
    4. **Loss Function**: Linear regression uses Mean Squared Error, which is unsuitable for categorical outcomes. Classification models use Log-Loss or Cross-Entropy, which better measures classification performance.
    5. **Prediction Probabilities**: Linear regression does not provide probabilities for class membership, while logistic regression and other classifiers do, facilitating better decision-making.

Lecture-5

Linear Regression: $Y = A + BX$; $\hat{y} = a + bx$

$A = \bar{y} - B\bar{x}$; $SS_{xy} = \sum (x_i - \bar{x})(y_i - \bar{y})$

$B = \frac{SS_{xy}}{SS_{xx}}$; $SS_{xx} = \sum (x_i - \bar{x})^2$

Goodness of the fitting Model:

Variance of Y values is $\frac{1}{N} \sum_{i=1}^{N} (Y_i - \bar{Y})^2$

$$\frac{1}{N} \sum_{i=1}^{N} (Y_i - \bar{Y})^2 = \frac{1}{N} \sum_{1}^{N} (Y_i - \hat{Y} + \hat{Y} - \bar{Y})^2$$

$$= \frac{1}{N} \sum_{1}^{N} (Y_i - \hat{Y}_i)^2 + \frac{1}{N} \sum_{1}^{N} (\hat{Y}_i - \bar{Y})^2$$

$SST = SSE + SSR$

$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$; R^2 closer to 1, fitted model is good.

R^2 closer to 0, fitted model is bad.

Quiz-2 (Solve)

Set-A

X	Y	$x_i - \bar{x}$	$y_i - \bar{y}$	$(x_i - \bar{x})^2$	$(x_i - \bar{x})(y_i - \bar{y})$	$\hat{y}$	$(y_i - \bar{y})^2$	$(y_i - \hat{y}_i)^2$	$(\hat{y}_i - \bar{y})^2$
55	4040	-941	-146918	885481	138249838	62693.51	21584898724	3440235367	7790627315
77	120120	-919	-30838	844561	28340122	64757.08	950982244	3065051937	7430508220.65
1212	180180	216	29222	46656	6311952	171218.49	853925284	80308534.5	410487360.24
1616	210210	620	59252	384400	36736240	209113.13	3510799504	1203122.654	3382020663.82
2020	240240	1024	89282	1048576	91424768	247007.76	7971275524	45802628.98	9225553749.63
$\bar{x} = 996$	$\bar{y} = 150958$			$SSxx = 3209674$	$SSxy = 301062920$		$SST = 6974376256$	$SSE = 1326520318$	$SSR = 28236762933.34$

$$\bar{y} = A + BX \quad ; \quad B = \frac{SSxy}{SSxx} = \frac{301062920}{3209674}$$

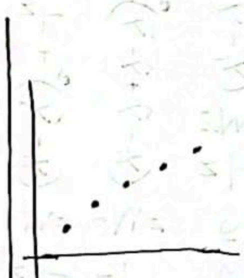
$$= 93.79$$

$$A = \bar{y} - B\bar{x} = 150956 - (93.79 \times 996)$$

$$= 57534.59$$

The regression line: $\hat{y} = 57534.59 + 93.79X$

(Ans)



$$R^2 = 1 - \frac{SSE}{SST}$$

$$= 1 - \frac{1326520318}{6974376256}$$

$$= 0.80$$

(Ans)

So, the model is good fit.

Lecture-6 (See the association math using the following youtube link)

Apriori algorithm example-2 in data mining in bangla/Data mining tutorial in …

Lecture-6

Association Rules are used to find interesting relationships, patterns, and associations among a set of items in a dataset.

Unlike supervised learning where we have a target variable to predict, association rules do not have a predefined target. Instead they explore data to uncover patterns. It describe relationships within the data but do not predict future outcomes.

* Prescriblive analytics goes beyond predicting future outcomes by recommending actions to acheive desired goals. It suggests the best course of action based on the analysis.

Support: measures how frequently the items X and Y appear together in the dataset.

$$Support(x \to y) = \frac{frq(x,y)}{N}$$

; $frq(x,y)$, measures how frequently the item X and Y appears in the dataset.

; N= total transactions

Confidence: measures how often items in Y appears in ~~these~~ transactions that contain X.

$$confidence(x \to y) = \frac{frq(x,y)}{frq(x)}$$

; $frq(x,y)$ is the frequency count of transaction that contain both X and Y.

; $frq(x)$ is the frequency ~~that~~ count that contains X.

$$Lift(x \to y) = \frac{Support(x \to y)}{\underbrace{Supp(x)}_{\frac{f(x)}{N}} \times \underbrace{Supp(y)}_{\frac{f(y)}{N}}}$$

Lift: Lift ~~means~~ measures the strength of a rule over the random cooccurence of X and Y. It shows how much more likely Y is to occur when X occurs compared to when X and Y are independent

Lift value near 1 indicates X and Y appear almost together.
Lift > 1, ~~value~~ means they appear together more than expected.
Lift < 1, means they appear together less than expected.

Example 2

Rule: {Milk, Diaper} ⇒ {Beer}

$$\text{Support} = \frac{2}{5} = 0.4$$

$$\text{confidence} = \frac{2}{3} = 0.67$$

$$\text{Lift} = \frac{0.67}{3/5}$$

$= 1.11 > 1$;

So, appear together more than expected.

$$\text{Lift} = \frac{\text{support}(x \to y)}{\text{supp}(x) \times \text{supp}(y)}$$

$$= \frac{frq(x,y)/N}{frq(x)/N \times frq(y)/N}$$

$$= frq(x,y) \times \frac{1}{freq(x)} \times \frac{N}{freq(y)}$$

$$= \frac{frq(x,y)}{freq(x)} \times \frac{N}{freq(y)}$$

$$= \text{confidence}(x \to y) \times \frac{1}{\text{sup}(y)}$$

$$= \frac{\text{confidence}(x \to y)}{\text{supp}(y)}$$

Apriori Algorithm:

Input:

D = Database Transaction

I = Items

L = Large Itemset

S = Support

α = Confidence

Output:

R = association rules satisfying s and α

(1) $R = \emptyset$

Initialize R with empty set.

(2) for each $I \in L$ do

Assume = {Bread, Peanut Butter}

(3) for $x \in I$ such that $x \neq 0$ do

Subset x are {Bread} and {Peanut Butter}

Now for x = {Bread}

compute confidence

$= \frac{\text{support (Bread, Peanut)}}{\text{Support (Bread)}}$

$= \frac{60}{80} = 0.75 > 50\%$ [let $\alpha = 0.5$]

$\cdot$ if $\frac{\text{Support}(I)}{\text{Support}(x)} \geq \alpha$

$R = R \cup \{x \Rightarrow (I - x)\}$

Add rule to R.

... Repeat the step 3 for Peanut

(Quiz-2)

Set-C

ID	Items
1	{A,C,D}
2	{B,C,D}
3	{A,B,C,D}
4	{B,D}
5	{A,B,C,D}

S = minimum support = 30% = $\frac{30}{100} \times 5 = 1.5$

α = confidence = 50%

C_1:

item	Support
{A}	3/5 = 60%
{B}	4/5 = 80%
{C}	4/5 = 80%
{D}	5/5 = 100%

C_2:

item	Support
{A,B}	2/5 = 40%
{A,C}	3/5 = 60%
{A,D}	3/5 = 60%
{B,C}	3/5 = 60%
{B,D}	4/5 = 80%
{C,D}	4/5 = 80%

C_3:

item	Support
{A,B,C}	2/5 = 40%
{A,B,D}	2/5 = 40%
{A,C,D}	3/5 = 60%
{B,C,D}	3/5 = 60%

C_4:

item	Support
A,B,C,D	1/5 = 20% ✗

Set of Large Itemset:

L = {{A}, {B}, {C}, {D}, {A,B}, {A,C}, {A,D}, {B,C}, {B,D}, {C,D}, {A,B,C}, {A,B,D}, {A,C,D}, {B,C,D}}

Rule	Support	Confidence
A,B → C	40%	40/40 = 100%
A,C → B	40%	40/60 = 66.67%
B,C → A	40%	40/60 = 66.67%
C → A,B	40%	40/80 = 50%
B → A,C	40%	40/80 = 50%
A → B,C	40%	40/60 = 66.67%
A,B → D	40%	40/40 = 100%
A,D → B	40%	40/60 = 66.67%
B,D → A	40%	40/80 = 50%
D → A,B	40%	40/100 = 40% ×
B → A,D	40%	40/80 = 50%
A → B,D	40%	40/60 = 66.67%
A,C → D	60%	60/60 = 100%
A,D → C	60%	60/60 = 100%
C,D → A	60%	60/80 = 75%
D → A,C	60%	60/100 = 60%
C → A,D	60%	60/80 = 75%
A → C,D	60%	60/60 = 100%
B,C → D	60%	60/60 = 100%
C,D → B	60%	60/80 = 75%
B,D → C	60%	60/80 = 75%
D → B,C	60%	60/100 = 60%
B → C,D	60%	60/80 = 75%
C → B,D	60%	60/80 = 75%

* All the Rules having confidence above 50% is Accepted.

Lecture-7

Content-Based Filtering (CBF) is a recommendation technique where items are suggested to a user based on the content of items they have previously interacted with. Here's a simple breakdown:

1. **Content Matching**: The system looks at the attributes or tags (like keywords) of items a user has liked or interacted with.
2. **Keyword Analysis**: Items are tagged with specific keywords, and the system identifies the preferences of the user based on these tags.
3. **Recommendation**: Using the understanding of the user's preferences (like preferred genres, topics, etc.), the system recommends new items that share similar content with those the user liked before.

Example: In a movie recommendation system, if a user watches and likes action movies, the system will analyze the keywords (like "action," "thriller") and recommend more action-related films based on that content.

Collaborative Filtering (CF) is a recommendation technique that suggests new items to a user based on the preferences and behaviors of similar users. Here's a simple explanation:

1. **User Similarity**: The system looks at what users with similar tastes have liked or interacted with and suggests those items to each other.
2. **Improvement Over Content-Based Filtering**: Unlike content-based filtering, which only looks at the items, collaborative filtering focuses on user behavior and interactions, making the recommendations more personalized and accurate.
3. **Predicting Preferences**: By analyzing what users have liked in the past, the system can predict what they might like in the future.

Two Types of Collaborative Filtering:

1. User-Based Collaborative Filtering (UBCF):

- **What it does**: It recommends items to a user based on the preferences of similar users.
- **How it works**: The system finds users who have similar interests or behaviors (like watching the same movies or buying similar products). Then, it recommends items that those similar users liked but the current user hasn't interacted with yet.
- **Example**: If User A and User B both love action movies and User B watches a new action movie, the system might recommend that movie to User A because their preferences are similar.

2. Item-Based Collaborative Filtering (IBCF):

- **What it does**: It recommends items based on the similarity between items themselves, focusing on the user's own history.
- **How it works**: The system looks at the items a user has interacted with and finds similar items to recommend. It compares items based on features like ratings or user behavior and suggests items that other users have rated or interacted with in a similar way.
- **Example**: If a user watches and enjoys an action movie, the system might recommend other action movies that are rated similarly or have similar characteristics.

Key Difference:

- **User-Based** focuses on finding similar users to make recommendations.
- **Item-Based** focuses on finding similar items based on the user's own past preferences.

Use gpt to learn about matrix factorization