USN | | | | | | | | | |

Internal Assessment Test 2 – July 2024

Sub:	**Machine Learning**				Sub Code:	**21AI63**	Branch:	**AInDS**	
Date:		Duration:	**90 minutes**	Max Marks:	**50**	Sem	**VI**		**OBE**

		Answer any FIVE Questions	**MARKS**	**CO**	**RBT**
1	a	Differentiate between Linear Regression and Logistic Regression	5	CO3	L1
	b	Compare Classification with regression with an example	5	CO2	L1
2		Explain End to End Machine Learning Project steps	10	CO2	L2
3		Suppose 10000 patients get tested for flu; out of them, 9000 are actually healthy and 1000 are actually sick. For the sick people, a test was positive for 620 and negative for 380. For the healthy people, the same test was positive for 180 and negative for 8820. Construct a confusion matrix for the data and compute the precision and recall for the data.	10	CO2	L3
4		Given the set of values X = (3, 9, 11, 5, 2)T and Y = (1, 8, 11, 4, 3)T . Evaluate the regression coefficients.	10	CO3	L3
5		Explain following 1) Gradient Descent 2) Polynomial Regression	10	CO2	L1
6		Explain how Support Vector Machine can be used for classification of linearly separable data and nonlinear data	10	CO3	L1

CI **CCI** **HOD**

Q 1a) Differentiate between Linear Regression and Logistic Regression [5 difference 5 marks]

Feature	**Linear Regression**	**Logistic Regression**
Purpose	Predicts a continuous output	Predicts a categorical output
Output Type	Continuous (real numbers)	Categorical (binary or multi-class)
Assumptions	Linearity, independence, homoscedasticity, normality of errors	Linearity in the logit, independence
Algorithm	Ordinary Least Squares (OLS)	Maximum Likelihood Estimation (MLE)
Decision Boundary	Not applicable	Defined by a threshold (e.g., 0.5 for binary)
Extensions	Polynomial Regression, Ridge, Lasso	Multinomial Logistic Regression, Ordinal Logistic Regression

Q 1b) Compare Classification with regression with an example [4 difference 4M and Example 1M]

Classification

Definition: Classification involves predicting a categorical label for each input instance.

Example: Suppose we want to predict whether an email is spam or not. The input features might include the frequency of certain words, the length of the email, and the presence of links. The output variable is categorical, with possible labels "spam" or "not spam."

Steps:

1. **Data Collection**: Collect a dataset of emails labeled as "spam" or "not spam."
2. **Feature Extraction**: Extract relevant features from each email.
3. **Model Training**: Train a classification algorithm (e.g., logistic regression, decision tree, or neural network) on the labeled data.
4. **Prediction**: Use the trained model to classify new emails as "spam" or "not spam."

Regression

Definition: Regression involves predicting a continuous numerical value for each input instance.

Example: Suppose we want to predict the price of a house based on features like the size of the house, the number of bedrooms, and the location. The output variable is continuous, representing the price of the house.

Steps:

1. **Data Collection**: Collect a dataset of houses with their prices and associated features.
2. **Feature Extraction**: Extract relevant features from each house.
3. **Model Training**: Train a regression algorithm (e.g., linear regression, decision tree, or neural network) on the labeled data.
4. **Prediction**: Use the trained model to predict the price of new houses based on their features.

Key Differences

1. **Nature of Output**:
   - **Classification**: Output is categorical.
   - **Regression**: Output is continuous.
2. **Evaluation Metrics**:
   - **Classification**: Common metrics include accuracy, precision, recall, F1 score, and ROC-AUC.
   - **Regression**: Common metrics include mean absolute error (MAE), mean squared error (MSE), and R-squared.
3. **Examples of Algorithms**:
   - **Classification**: Logistic regression, decision trees, support vector machines, neural networks.
   - **Regression**: Linear regression, decision trees, support vector regression, neural networks.

Example Comparison

Classification Example: Predicting if a student will pass or fail a course based on study hours, attendance, and past grades.

- **Input Features**: Study hours, attendance, past grades.
- **Output**: "Pass" or "Fail."

Regression Example: Predicting a student's final exam score based on study hours, attendance, and past grades.

- **Input Features**: Study hours, attendance, past grades.
- **Output**: Final exam score (a continuous value, e.g., 85%).

Q 2) Explain End to End Machine Learning Project steps [Each Step 2M]

1. Look at the big picture.
2. Get the data.
3. Discover and visualize the data to gain insights.
4. Prepare the data for Machine Learning algorithms.

5. Select a model and train it.
6. Fine-tune your model.
7. Present your solution.
8. Launch, monitor, and maintain your system.

Get the Data

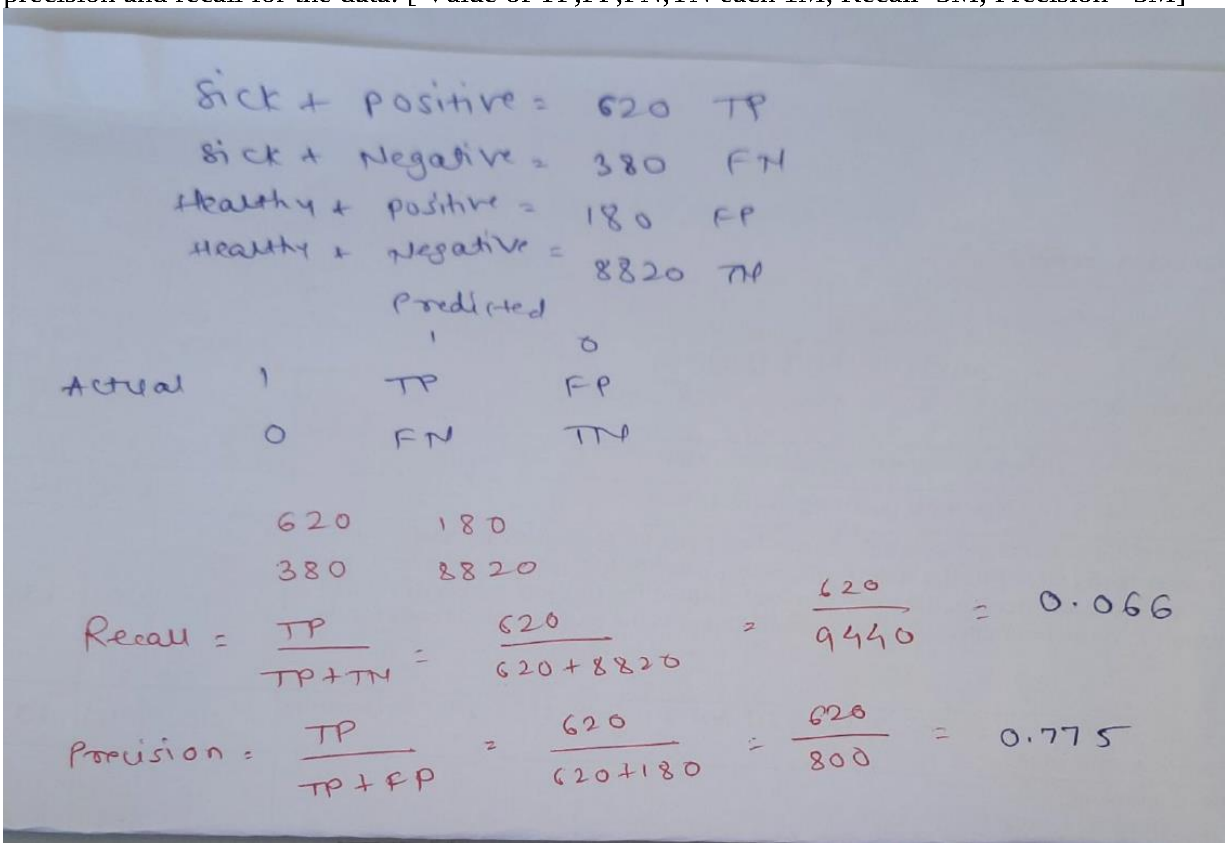
Create the Workspace
Download the Data
Take a Quick Look at the Data Structure
Use head(), info(), describe() Functions
Use histogram to visualize numerical attributes
Create a Test Set

Data Cleaning

Missing value treatment
Handling Text and Categorical Attributes
Custom Transformers
Feature Scaling

- ❑ Min-max
- ❑ Standard

Q 3) Suppose 10000 patients get tested for flu; out of them, 9000 are actually healthy and 1000 are actually sick. For the sick people, a test was positive for 620 and negative for 380. For the healthy people, the same test was positive for 180 and negative for 8820. Construct a confusion matrix for the data and compute the precision and recall for the data. [Value of TP,FP,FN,TN each 1M, Recall=3M, Precision= 3M]



Sick + positive = 620 TP
Sick + Negative = 380 FN
Healthy + positive = 180 FP
Healthy + Negative = 8820 TN

		Predicted 1	0
Actual	1	TP	FP
	0	FN	TN

620	180
380	8820

$$\text{Recall} = \frac{TP}{TP+TN} = \frac{620}{620+8820} = \frac{620}{9440} = 0.066$$

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{620}{620+180} = \frac{620}{800} = 0.775$$

Q 4) Given the set of values X = (3, 9, 11, 5, 2)T and Y = (1, 8, 11, 4, 3)T . Evaluate the regression coefficients. [value of a=4M value of b=4M final equation = 2M]

Mean of X= [3+9+11+5+2]/5 = 6

Mean of Y= [1+8+11+4+3]/5 = 5.4

$$
\begin{aligned}
b &= \frac{\sum_{\mathrm{i=1}}^{\mathrm{N}} (\mathrm{X_i} - \mathrm{mean_X}) \cdot (\mathrm{Y\prime_i} - \mathrm{mean_Y})}{\sum_{\mathrm{i=1}}^{\mathrm{N}} (\mathrm{X_i} - \mathrm{mean_X})^2} \\
&= \frac{\sum_{i=1}^{5} (X_i - 6) \cdot (Y\prime_i - 5.4)}{\sum_{i=1}^{N} (X_i - 6)^2)} \\
&= \mathbf{1.0} \\
a &= \mathrm{mean_{Y\prime_i}} - \mathrm{b} \cdot \mathrm{mean_X} \\
&= 5.4 - 1.0 * 6 \\
&= \mathbf{-0.6}
\end{aligned}
$$

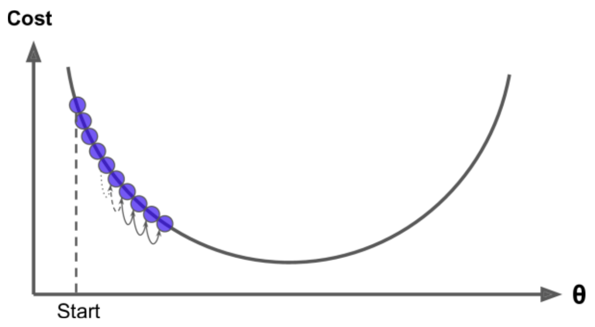
Q 5) Explain following [5 marks each]

1) Gradient Descent
2) Polynomial Regression

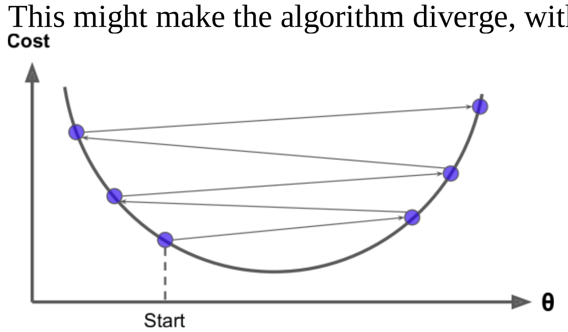
Gradient Descent:

Gradient Descent is a very generic optimization algorithm capable of finding optimal solutions to a wide range of problems.
The general idea of Gradient Descent is to tweak parameters iteratively in order to minimize a cost function.
It measures the local gradient of the error function with regards to the parameter vector **θ**, and it goes in the direction of descending gradient.
Once the gradient is zero, you have reached a minimum

- An important parameter in Gradient Descent is the size of the steps, determined by the *learning rate* hyperparameter.
- If the learning rate is too small, then the algorithm will have to go through many iterations to converge, which will take a long time

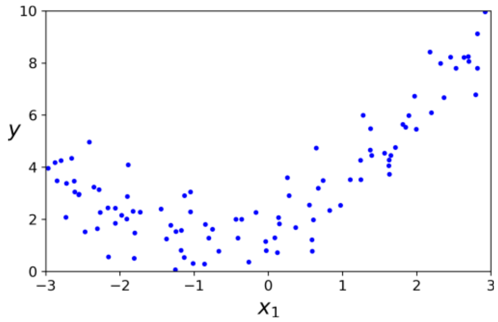


- On the other hand, if the learning rate is too high, you might jump across the valley and end up on the other side, possibly even higher up than you were before.
- This might make the algorithm diverge, with larger and larger values, failing to find a good solution

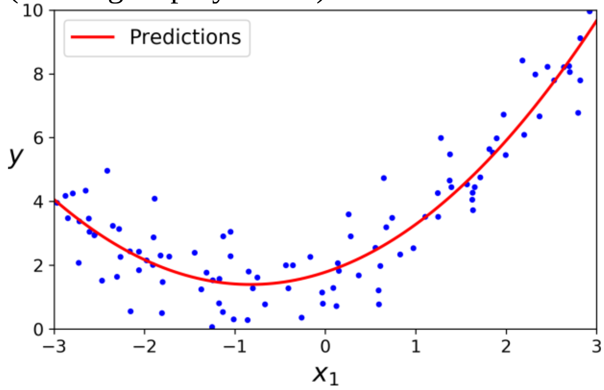


Polynomial Regression

- What if your data is actually more complex than a simple straight line?
- Surprisingly, you can actually use a linear model to fit nonlinear data.
- A simple way to do this is to add powers of each feature as new features, then train a linear model on this extended set of features.
- This technique is called *Polynomial Regression*.



- Clearly, a straight line will never fit this data properly.
- So let's use Scikit-Learn's PolynomialFeatures class to transform our training data, adding the square (2nd-degree polynomial) of each feature in the training set as new features

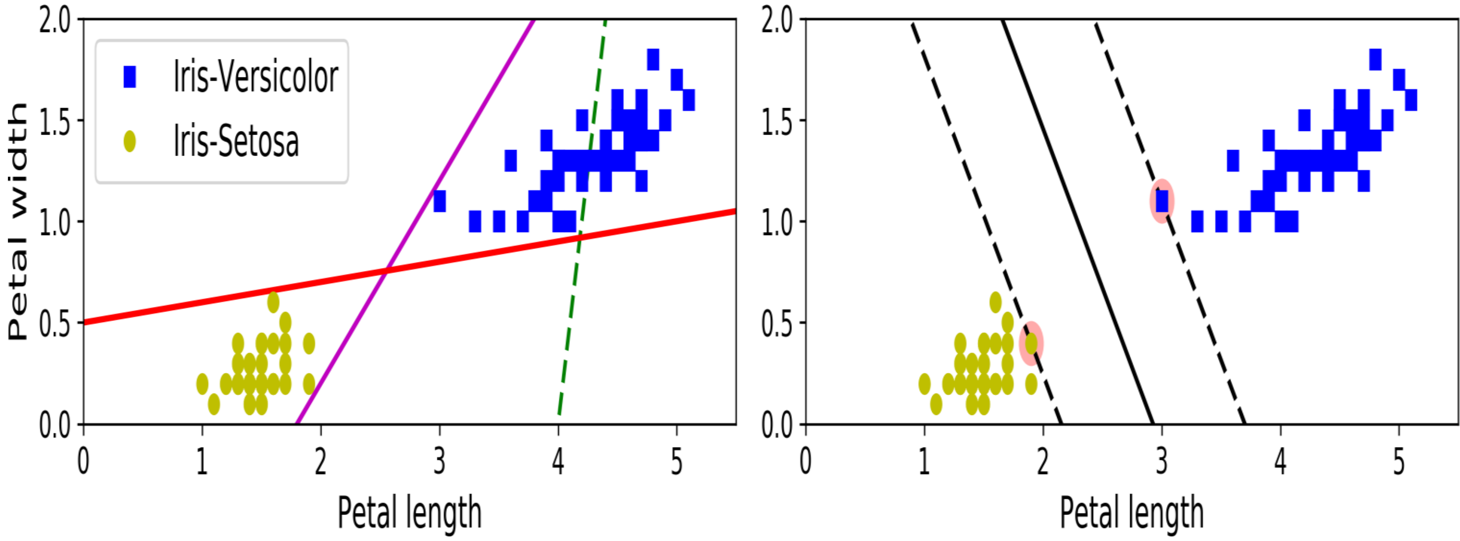


- The model estimates $y = 0.56x_1^2 + 0.93x_1 + 1.78$
- when in fact the original function was
- $y = 0.5x_1^2 + 1.0x_1 + 2.0$ + Gaussian noise.
- Note that when there are multiple features, Polynomial Regression is capable of finding relationships between features

  For example, if there were two features *a* and *b*, PolynomialFeatures with degree=3 would not only add the features a^2, a^3, b^2, and b^3, but also the combinations ab, a^2b, and ab^2

Q 6) Explain how Support Vector Machine can be used for classification of linearly separable data and nonlinear data

- A *Support Vector Machine* (SVM) is a very powerful and versatile Machine Learning model, capable of performing linear or nonlinear classification, regression, and even outlier detection.
- SVMs are particularly well suited for classification of complex but small- or medium-sized datasets.
- The two classes can clearly be separated easily with a straight line (they are *linearly separable*).
- The left plot shows the decision boundaries of three possible linear classifiers.
- The model whose decision boundary is represented by the dashed line is so bad that it does not even separate the classes properly.



Nonlinear SVM Classification

- Although linear SVM classifiers are efficient and work surprisingly well in many cases, many datasets are not even close to being linearly separable.
- One approach to handling nonlinear datasets is to add more features, such as polynomial features
- Consider the left plot in Figure 5-5: it represents a simple dataset with just one feature x_1.
- This dataset is not linearly separable, as you can see.
- But if you add a second feature $x_2 = (x_1)^2$, the resulting 2D dataset is perfectly linearly separable