MACHINE LEARNING

Machine Learning is a fascinating field with diverse concepts.

Here's a structured guide to get you started:

Basics of Machine Learning:

Steps:
Data Collection
EDA
FE
FS
Model Training - Pickle file
Web app - Flask
Github
Deployment - AWS Cloud

1. Introduction to ML

What is Machine Learning?

Types of Machine Learning:
Supervised, Unsupervised, and Reinforcement Learning

Supervised - Classification and regression (Classification= binary or multiclass, regression = Continuous Values)
Unsupervised - Cluster and similar groups
Semisupervised - a combination of the above(Netflix)
Reinforcement Learning - just like a baby learns based on the environments

Splitting the dataset examples:

Training - Books
Validation - Additional Books
Test - Exams

Model performance - More Accuracy / High performance

Overfitting - Low Bias/High Variance (Train Data = High Accuracy, TestData = Low Accuracy)

Underfitting - High Bias/LowVariance

Bias & variance

Bias is training accuracy wrt training data:
high accuracy = low bias
low accuracy = high bias

Variance is training accuracy wrt test data:
high accuracy = low variance
low accuracy = high variance

2. Data Preparation

Data Collection

Data Cleaning

EDA

Feature Engineering:

1. Handling Missing Values

- Types of data missing mechanism
MCAR - No specific reason for data missing
MAR - Some relationship with missing data
MNAR - With some reasons

- Imputation Techniques - Ways to handle missing values

2. Handling Imbalanced Dataset
-Upsampling - SMOTE(It joins two points and add data between them)
-Downsampling

lib = from sklearn.utils import resample

3. Data Interpolation(To create artificial datasets, to add more data to given data)
- Linear Interpolation
- Cubic Interpolation
- Polynomial Interpolation

4. Feature Extraction

1. Feature Scaling
- Standardization (with z score applying standard normal distribution
[x - mean/ standard deviation]) used in ML
- Normalization (using min-max scaler between 0-1
[x - Xmin / Xmax - Xmin]) used in DL
- Unit Vector - to get the magnitude of 1. Divide the other sides by the diagonal. and then calculate the diagonal again (normalise library)

2. Feature Selection (To select more important features)
- Filter Method (Correlation)
- Embedded method
- Wrapper method

3. PCA (Converting to higher dimension to lower dimension)

Difference between fit and transform. Fit calculates parameters wrt to formulas like mean and standard deviation of the data and transforms apply transformation to it.

5. DAta Encoding

- Nominal(OHE) -- when the features has many category don't use it.

- label and ordinal -- label and ranks are assigned, where ranks are requirred

- targetted guided ordinal encoding -- used when their relationship between target variable and categorical variable with large number of unique categories

6. Covariance -

- Spread of data-
- relationship between x and y

- + X + Y
- X - Y
results in + covariance

- + X - Y
- X -+Y
results in - covariance

7. To overcome the above problem we use :
- Pearson Correlation Coefficient [-1 to +1]

- Spearman Rank Correlation [-1 to +1]

Note - If the features are highly correalated, then drop the featuress based on some threshhold value....Given by domain expert. It is the concept of multicollinearity between independent features.

8. EDA(Exploratory Data Analysis)- To understand specific data set

1. Load Dataset
2. check info (summary)
3. Check descriptive statistcs
4. check/ count rows and column i.e shape

(data check to perform

1. Missing values
2. duplicates
3. data type
4. unique value of each column (nunique)
5. statistics
6. Different categories in different columns)

8. Check column names
9. Check unique values
10. check missing values (isnull.sum)
11. check for duplicated records
12. remove duplicate (drop duplicates)

Pandas-Profiling

13. check correlation
14. visualize coorelation with heat map using seaborn and change figure size using matplotlip.pyplot
15. check value counts and plot it to check balanced or unbalanced dataset.plot
16. Visulize distributtion of all the parameters(sns)
17. Check pair plot(sns)
18. check categorical plot(take categories and plot inside it using kind)sns

3. Machine Learning Models/ Algorithm

----------Linear Regression

---

1. Linear Regression

- It is used to solve regression problem.
- Line Equation, To get best fit line
- Error
- Cost function - MSE(Mean squared error)
- Aim -- To minimize MSE by optimizing Intercept and slope value again and again
- Gradient Decent curve drawn to get global minima
- Use conversion algorithm to reach to global minima
- Alpha i.e learning rate decides the conversion speed
- Derivative is calculated to get the slope of a point (theta and j(theta))
- (Right side direction = Upward +slope)
- (Right side direction = Downward -slope)

2. Multiple Linear Regression
 - Line Equation, To get best fit plane
 - gradient decent shape will be 3d Parabola

3. Polynomial Regression

Simple Polynomial Regression:

- It is used when the data has nonlinear realationship
- it will use ploy nomial degree.
- it is related with simple linear regression

Multiple Polynomial Regression:

- It is used when the data has nonlinear realationship
- it will use ploy nomial degree.
- it is related with multiple linear regression

Increase the degree to get minimum error

4. Performance Metrics / Accuracy

1. R square
- 0.75 denotes 75% accuracy (Max = 1)

-If their is the correlation between independent and independent features, it will increase. It will also increase by increasing number of features

2. Adjusted R square
- It will be less than r square
- It is feature independent.

5. COST FUNCTIONS

1. Mean Squared Error(MSE)

- It is just like a quadratic equation
- It is differentiable (At any point we can calculate slope)
- It has one local and one global minima
- It is not robust to outliers (because the errors are squared)
- It is not in the same unit ((becoz the errors are squared))

2. Mean Absolute Error(MAE)
- It is robust to outliers
- It will be in the same unit
- Conversion usually takes time
- Optimization is complex

3. Root Mean Squared Error(RMSE)
- Same unit
- Differtiable
- Not robust to outliers

= Steps to do before starting machine learning
1. Divide the features based on independent and independent features
2. Train Test Split
3. Standardize the features (to calculate gradient decent easily, it makes optimizing easy)
4. Training the data

= Assumptions when we can say, our model is performing better/good.

1. If y_test and y_pred_test scatter plot is showing linear relationship, it means it is good prediction.
2. If the graph(histogram) of residuals(y-y^) is coming as normal distribution, it means it is good prediction.
3. 2. If the graph of Y_pred and residuals should be uniform distribution, it means it is good prediction.

Ridge Regression (L2 Regularization)
- For reducing overfitting
- When lambda will increase, slope of global minima will decrease
- but it will never become 0

Lasso Regression (L1 Regularization)
- For feature selection
- Eliminating Low correlated feature
- When lambda will increase, slope of global minima will decrease
- But this can be 0

Elastic-Net Regression
- It is combination of both Ridge and Lasso, used fo reducing overfitting and For feature selection

Lambda 1 and lambda 2, will be selected with hyperparameter tuning.

Cross-Validation = Train data is further divided into train and validation dataset.
validation dataset help to perform hyperparameter tuning .

Ex = Data-Points = 1000, CV = 5

VD = Validation Dataset
TD = Training Dataset

when cv = 1
data will be divided as 200 / 800
VD TD

when cv = 2
data will be divided as 200 / 200 / 600
TD VD TD

when cv = 3
data will be divided as 400/ 200 / 400
TD VD TD

when cv = 4
data will be divided as 600/ 200 / 200
TD VD TD

when cv = 5
data will be divided as 800/ 200
TD VD

Check the accuracy with each training dataset, and then final accuracy which is average of all accuracy.

2. Logistic Regression:

- It is used to solve the classification problem
- It uses Squashing technique by using Sigmoid activation function (0-1)
- Sigma is responsible for squashing
- But we do not use sigmmoid activation function beacuse it create non-convex function
- We use a log loss function (Cost function)
- Final aim: To minimize cost function

1. log loss function + l2 regularization (to reduce overfitting)
2. log loss function + l1 regularization (for feature selection)
3. log loss function + elastic net (l1 + l2)

To check how well the model is performing we see:

----------confusion_matrix,accuracy_score,classification_report----------
---

- Confusion Matix

TP and TN should be high (they are most accurate data)
FP and FN should be low

- Accuracy

To calculate model accuracy we do TP + TN / TP + FP + FN + TN

If the dataset is imbalanced we use Precision, Recall and F-beta score

- Precision

TP / TP + FP
-Out of all the actual values, how many are correctly predicted with actual values.
-It is used where FP is important (to Reduce FP)

- USE CASE - 1:
Email - Spam and NOt Spam(1, 0) . MAil = 0, but model = 1 (wrong prediction + blunder)

- Recall
TP / TP + FN
-Out of all the predicted values, how many are correctly predicted with actual values.
-It is used where FN is important (to Reduce FN)
- - USE CASE - 2:
Diabetes- Diabetes and NO Diabetes(1, 0) . Diabetes = 1, but model = 0 (wrong prediction + blunder) because person will skip the check-up

- F-Beta score
(1 + beta^2) Precision * Recall / Precision + Recall
1. If FP + FN are both important (Beta = 1) equation also called - Harmonic Mean(F-1 score)
2. If FP is more important than FN(Beta = 0.5)(F-0.5 score)
3. If FN is more important than FP(Beta = 2)(F-2 score)

= Cross-validation help to do hyperparameter tuning
= Types Of cross Validation

1. Leave one out CV (LOOCV):
- Time Complexity is huge for training big dataset
- Model tends to overfit.

To overcome the above disadvantage we use this----
2. Leave P out CV (LPOCV)(P = 10,20,100.....):

3. K-fold Cross Validation:
K = Number of experiments to be performed. dataset/K = CV dataset

4. Stratified K-fold Cross Validation:
- It is used when the dataset has imbalanced dataset
- In CV dataset 0s, 1s will be equal

Hyperparameter Tuning:
1. Grid Search CV
- In this we take all the possible combination
- Because of all possible combination, Time Complexity increases with huge dataset

2. Randomized search CV
-Here we will select random parameters, n_iter times.(n_iter = 10, 20, 30, 80....)
- It is faster than a grid search CV.

- Time Complexity decreases

To solve the multiclass classification problem
We use this technique:
1. OVR - One Versus Rest
2. Multinomial

---Decision Trees

2.

1. Decision Trees Classifier

- Purity Split Check - Pure split or Impure Split
- Pure Split - when no further split is required, also called a leaf node.
(complete yes, complete no)
1. We check pure split by 2 techniques i.e Entropy, and Gini-Impurity
- Impure Split - when further split is required

2. What features do you need to select to start the split - Information Gain

3. When entropy gives the output 1 it means it is a completely impure split. And further calculation is required. When entropy gives the output 0 it means it is a completely pure split.

4. When gini-impurity gives the output 0.5 it means it is a completely impure split. And further calculation is required. When gini-impurity gives the output 0 it means it is a completely pure split.

5. Information Gain - is used to select the feature to start the split. We select the feature with a higher gain.

6. When the dataset is small(10,000) -> Entropy
When the dataset is huge -> Gini-impurity

7. Post-Pruning and Pre-Pruning

When we further split the decision tree to its end, there are chances of overfitting.

Pruning = Cutting

1. Post-Pruning = First Construct the full tree to its max_depth, then prune it.
It is suitable for smaller datasets.

2. Pre-Pruning = We perform hyperparameter tuning to select the best parameters.

- No need to perform feature Scaling.

2. Decision Trees Regressor

1. Final aim to calculate variance Reduction
2. Variance or error

- Select that split where variance reduction is high.

Random Forest

----------Support Vector Machines (SVM)

---

1. Support Vector Classifier(SVC)
- It creates marginal lines above and below the best fit line.
- Wherever the distance between marginal lines is maximum, we will take that.
- The points touched by marginal lines are called Support Vector.
- Hard Margin - When none of the data is misclassified. (Impossible chances)
- Soft Margin - Some of the data points are misclassified (there will be some error, which is ok.)

Cost-function for hard-margin

- Cost-function = maximize = 2 / ||W|| => Distance between marginal plane.
- Also, Cost-function = minimize = ||W|| / 2
- For all the correctly identified points, the cost function will be >= 1

Cost-function for Soft margin

f(w)= // w // 22+C(∑i=1Nξ)k

C(∑i=1Nξ)k = Hinge Losss

C = How many points we can consider, threshold value.(5,10,15)
C = 1 / lambda
ξ = Summation of the distance of incorrect data points to the marginal plane.

2. Support Vector Regressor(SVR)

Upper Marginal Plane - Wtx + b + ε

Lower Marginal Plane - Wtx + b - ε

3. Support Vector Kernal

- we convert 2d data points to 3d data points(Basically lower to higher dimensions), and then we can create marginal plane.
- We convert it by transforming the points by mathematical formula

= > Three types of Support Vector Kernal techniques

1. Polynomial kernel
2. RBF kernel
3. Sigmoid kernel

----------Naive Bayes

---

1. Naive Bayes
- It is especially used to solve the classification problem.
- Probability
- Bayes Theorem

- Variants of Naive Bayes :
1. Bernoulli Naive Bayes - 0,1 dataset and also used in NLP because it has Sparse Matrix i.e maximum 0's and 1's.
2. Multinominal Naive Bayes - Used in NLP ex. spam and not-spam(ham)
3. Gaussian Naive Bayes - when the distribution is in bell shape i.e Gaussion Distribution.

----------Ensemble technique and its types----------

- It is used to improve the accuracy of the model, model performs well.
- Ensemble = Combining multiple models
- Two Types :

1. Bagging
- Random Forest Classifier
- Random Forest Regressor

They both uses Base learners

- Random forest classifier and regressor - only decision tree is used by default, and also row sampling and feature sampling is used for the base learners.

- Custom Bagging Technique:
- the training dataset is divided into samples to create base learners(models), this technique is called bootstrap aggregation.

1. In the classification problem we take the majority voting classifier, maximum occurring output.
2. In the regression problem we take the average, average of all the models.

- Out of bag score - It is the remaining unutilized data, from the dataset. Which can be important. It is called out-of-bag data.
- OOB_score = False (We will miss the data)
- OOB_score = True (The data will be used as validating data to check the performance and accuracy of the model.)

2. Boosting
- Adaboost - we assign weights to the weak learners
- Gradient boost
- xgboost

1. Adaboost algorithm

- - we assign weights to the weak learners

weak learners = decision tree stump

-They uses weak learners.
-Stump = All the sequential decision tree, depth 1
-In boosting we have sequence of decision trees, all the wrong prediction are transferred to the next decision tree. Each sequence is weak learners.

Steps:
1. Create decision tree stumps an select best models
2. Assigning sample weights, it will help to calculate the sum of total errors
3. Performance of the stump
4. Updates the wrong and right data weights. Increase the weights of wrongly classified datapoints, and decrease the weights of correcly classified data points.
5. Normalize weights and assign bins.
6. Passing wrong predicted points to other stump.
7. High perfomance score will get selected.

2. Gradient boosting algorithm
- both classification and regression problems can be solved.

Steps:
1. Create a base model
2. compute the residual and error
3. Construct a decision tree - using features and predicted output
4. Repeat

3. xgboost - Xtreme gradient boosting algorithm
- both classification and regression problems can be solved.

Steps:
1. Create a base model
2. compute the residual and error
3. Construct a decision tree - using features and predicted output
4. Calculate Similarity Weights
5. Calculate Gain
6. Repeat

---------- k-Nearest Neighbor
—----------(k-NN)----------
---

1. We can solve both classification and regression problem.
2. K > 0 to ……
3. Initialize K value
4. Find the k nearest neighbour from the test data
5. Check how may points belong to which category.
6. To optimize the KNN, we use two techniques that are KD tree, BAll TRee. They both are binary tree. We use this to decrease time complexity.
7. KD tree - Dividing into blocks
8. BAll TRee - Dividing into groups

---------- Principle Component Analysisr
—----------(PCA)----------

PCA - Dimensionality Reduction

It is data transformation technique.

- To different ways to remove curse of dimensionality

1. Feature Selection - Imp Features
    - Covariance = +ve - Positive Relationship, -ve Negative Relationship
    - Peason Correlation = (Negative Relationship) -1 to +1 (Positive Relationship)
    - We can drop the non-related features
    - Here there is data loss

2. Dimensionality Reduction - Feature Extraction
    - Less data loss
    - Maximum Variance is captured.
    - Pc1 > pc2 > pc3…..pcN
    - In case of 3D - 1D

- Var(Pc1) > Var(Pc2) > Var(Pc3)
- Eigen vector -> Eigen Value — Magnitudeof eigen vectors
- Eigen vector - We will choose that, which Captures the maximum variance.
- A . V = lambda * V

Steps ;

1. Standardize the data
2. Matrix of variance and Covariance
3. Lambda Value to calculate PC line
4. Select best line, which Captures the maximum variance.

---------- Unsupervised Machine Learning Algorithm—----------

---

Clustering - Group your data in similar clusters, having similar values.

Algorithms :

1. K- Means Algorithm
2. Hierrical Clustering
3. DBSCAN Clustering

    1. Silhouette Scoring - to validate the correct data.

1. K- Means Algorithm

    Steps:

    1. We initialize some centriods (K-value)(k = 2)(Cluster 1, Cluster 2)
    2. We need to find the points, nearest to the centroids.
    3. Move the centrioids - Average
    4. Repeat from 2nd Step

WCSS - Within centroids some of squares
Elbow method - When the WCSS is decreasing abdruptly, when it gets stablize select

the K-Value.

- K- Means ++ Initializing technique (The distance between each K-point is maximum)

1. Hierarchical Clustering - there are no centroids

1. Agglomeritive - It means combining
   Bottom to top

Steps:

2. For each datappoint, initially will consider a separate class
3. Find the nearest point, and create a new cluster.
4. Repeat, until single cluster

2. Divisive - It means dividing
   Top to bottom

-Dendrogram
-Setting Threshhold Value to identify number of clusters.
- Longest Vertical such that no horizontal line passes to it for setting threshold values.

K means vs Hirarchical Clustering

- Dataset Size:

  Huge - Kmeans
  Small - Hierarchical

- K mean - Numerical Data
- Hierarchical Clustering - Numerical and categorical (based on cosine similarity) (Variety of data)

3. DBSCAN Clustering (Work with non linear data also)

1. Red point - Core Point

- Minimum Point
- Epsilon - Radius
- Minimum point within Radius

2. Yellow Point - Border Point
    - No of datapoint will be less than min points

3. Blue Point - Outlier (Noise)
    - No datapoint within radius

:- Silhouette Scoring - to validate the correct data.

Range between -1 to 1

1. More near to 1 better clustering model we have created.a(i) < b(i)
2. More near to - 1 worst clustering model we have created (a(i) > b(i))

- Anomaly Detection(To detect outlier) — Where this outlier is very important

1. Isolation Forest [Decision Trees]
    - Data point is isolated as a leaf node.
    - Anomaly score coming to 1 = Outlier
    - Anomaly score coming to 0 = Normal Data Point

2. Dbscan Clustering Anomaly Detection

3. Local Outlier factor Anomaly Detection
    - Local Outlier - When density is less
    - Global Outlier - When density is more
    - LOF score
    - KNN is used internally

----------Time Series----------
---

1. Time Series (Column representing time)
2. Non - Time Series (No time Column)

- Time Stamp = Hours, min, days, month , year

1. Interpolation - To find out the value in the range itself

- Here based on historical data, we will try to predict the data, within the same range.

2. Extrapolation - To find out the value outside of range

- Here based on previous history, we will try to predict future data, we forecast the data….

Time Series data to be made stationary, then create a model:

1. Arima
2. Sarima
3. Sarimx

For **checking** Stationary or Non stationary

1. Visualization
2. Stats based test
- ADF - Augumented Dickey Fullar Test
    1. Static vale
    2. P-Value - By seeing this we can say Stationary or Non stationary
    3. Critical Value

For doing DATA **Decomposition**:

1. Additive
2. Multiplicative

Check **Outlier**

To **convert** non-st to st

1. Differencing - until we get stationary data
2. Log
3. Root
4. Adjustment of seasonal data

ACF , PACF , Auto Regression

ACF - Auto Correlation funtion
PACF - Partial Auto Correlation Function
Auto Regression

1. ACF - Measure correlation between time series and its lag value. Current to previous.

2. PACF - Measure correlation between time series and its lag value. Current to previous leaving columns between.

3. ARIMA - Auto Regression Integration Moving Average
    - Models:
      ARIMA
      SARIMA, SARIMX
      Dl, Rnn, Attention, Transformer

ARIMA - Auto Regression Integration Moving Average

(p, d, q)

SARIMA - Seasonal Auto Regression Integration Moving Average

(p, d, q) (P, D, Q)s

SARIMX - Seasonal Auto Regression Integration Moving Average Exgeouns(outlier)

(p, d, q) (P, D, Q)s x

—----------

----------End-To-End Project Implimentations—----------

—----------

1. Data ingestion (Data collection)
2. EDA (analysisi of data)
3. FE (Preprossing or transformation)
4. Model Building
5. Model Evaluation

: Projects

1. **Diamond Price Predictions**
2. **Wafer fault detection**

DEVOPS - Continuous Integration +Continuous Deployment
MLOPS - Continuous Integration + Continuous Training +Continuous Deployment

App - Image - Container - Docker (for creating virtual machines)

—----------
----------Deployement----------
—----------
Reference = WaferFaultDetection

1. Post the code to github, and create .github/workflow/main.yml file
2. For making a connection you required these credential

   Login to aws account

3. OPen the IAM service
4. CReate a user
5. Generate
6. AWS_ACCESS_KEY_ID
7. AWS_SECRET_ACCESS_KEY
8. Go to ECR
9. Create a repo
10. AWS_DEFAULT_REGION
11. AWS_ECR_REPO_URI
12. Go to github/settings/secret key create secret key for the above variable.

13. Push the code from vscode
14. Check for the image, which has been created
15. Now go to AWS runner
16. Create Service
17. Select Automatic
18. Create new service role
19. Give service name
20. Then, Next
21. Select CPU, and RAM, PORT_NUMBER
22. Then Next
23. CReate and Deploy

Neural Networks

4. Model Evaluation

Metrics: Accuracy, Precision, Recall, F1 Score

Cross-Validation

Overfitting and Underfitting

5. Feature Selection

- Importance of Feature Selection

Methods: Filter, Wrapper, Embedded

Intermediate Concepts:

6. Ensemble Learning

- Bagging and Boosting

- Adaboost

Gradient Boosting

7. Clustering

K-Means

Hierarchical Clustering

DBSCAN

Dimensionality Reduction (PCA)

8. Natural Language Processing (NLP)

Text Preprocessing

Tokenization

Word Embeddings (Word2Vec, Glove)

- Text Classification

9. Deep Learning

- Neural Networks

Convolutional Neural Networks (CNN)

Recurrent Neural Networks (RNN)

Transfer Learning

- Autoencoders

10. Time Series Analysis

- Time Series Data

- ARIMA Models

- LSTM for Time Series

Advanced Topics:

11. Reinforcement Learning

Basics of RL

Markov Decision Process (MDP)

- Q-Learning

- Deep Q-Networks (DQN)

12. Generative Adversarial Networks (GANs)

- Introduction to GANS

Generator and Discriminator

- Applications of GANS

13. Deploying Machine Learning Models Model Deployment Strategies

Containers (Docker)

Cloud Services (AWS, Azure)