MACHINE LEARNING FUNDAMENTALS

- Machine Learning is programming computers to optimise themselves using example data or past data.
- It can automatically detect uncovered patterns in the data and use them to predict future data or outcomes.
- It falls in the category of Artificial Intelligence and uses Statistical Models to predict and infer.
- Machine Learning is analogous to Human Learning. Humans observe things around them and process them that finally becomes a skill and helps them preform better. Similarly, we give past data to the computer and program in such a way that the computer learns from the past data and enhances its own performance.

- Example — A real estate agent wants to price the value of a property based on many features.

Output Variable (Y) : Price of property
Input Variable (X) : Area covered(X1), Number of Bedrooms(X2), Number of Bathrooms(X3), Proximity to Market(X4), and so on

$$Y = f (X1, X2, X3, X4, ….)$$

Whenever the agent will put the values of input variable in the function, the price of the property will be given as output.

Machine Learning Purposes

A. Prediction

- When the output is only what matters.
- Example, agent only wants to put a price on the property.
- Accuracy is very important.
- Non-linear models are preferred.

B. Inference

- When the relation between output and input variables is what matters.
- Example, property builder wants to understand how the price of a property will be affected.
- Interpretability is very important.
- Linear model are preferred.

Parametric and Non-Parametric Algorithm

A. Parametric

- We assume a functional form between output and features.
- The job is to find the best fit factors [alpha1, alpha2, alpha3, …..].
- Simple, fast and need less data for training but poor fit.
- Examples are Linear Regression, Logistic Regression, LDA, Perceptron etc.
- Better for inference problems because better at interpretability.

B. Non-Parametric

- This approach does not make any assumptions.
- Non-parametric approach needs a lot of data to learn from and thus has high accuracy.
- This makes it preferable for predictive problems.
- Examples are Decision Trees, Naive Bayes, SVM, Neural Networks.
- Slow and leads to overfitting.

Types of Learning

A. Supervised Learning

- We have a dataset having predictor variables and an output variable and goal is to find a mapping function between them.
- The task is to find the best mapping function,. So, if we feed new input variables values, we can predict the output value accurately.
- Classification applications are image-classification, sentiment-analysis, email-spam-detection etc.
- Regression applications are product-pricing, weather-forecast, advertising-budget-allocation etc.

B. Unsupervised Learning

- We have a dataset having input variables but no output variable.
- The task is to find underlying structure and patterns in the data, learn more about it and make business plans accordingly.

C. Reinforcement Learning

- Feedback based machine learning in which an agent [computer program] learns to behave in an environment by performing the actions and seeing the result of each action.
- For each good action, the agent gets a positive feedback and for each bad action, the agent gets a negative feedback.
- The difference between reinforcement and supervised learning is that reinforcement learning doesn't need any labelled data to train on. Since there is no labelled data present, the agent is bound to learn from experience.
- This type of learning is completely based on hit and trial method.

D. Semi-Supervised Learning

- Semi-supervised learning lies between supervised and unsupervised learning.
- The dataset used for semi-supervised learning has both labelled and unlabelled dataset. Generally, a minor segment of dataset is labelled and the major segment is unlabelled.
- There are several approached for semi-supervised learning.

Steps to Approach a Machine Learning Problem

STEP 1 Problem Formulation

- Convert business problem into a statistical problem.
- Define the dependant and independent variables.
- Decide whether your problem is a prediction problem or inference problem.

STEP 2 Data Collection

- Collection of data from primary and secondary research and creating a dataset that can be fed to machine learning models.

STEP 3 Data Preparation

- Data reduction
- Data cleaning
- Data integration
- Data transformation

STEP 4 Train-Test Split

- Split the data into training and testing dataset.
- Training dataset is used to train a machine learning model and testing dataset is used to testify the accuracy and interpretability of the model.
- Training dataset uses input and output variable values to identify a mapping function between them.
- The training dataset generally uses 70-80 percent of the data.
- Larger the training dataset, better the accuracy of the model.

STEP 5 Model Creation

- Train our data on a model and find the hyperparameters based on our problem needs.

STEP 6 Model Evaluation and Validation

- We can get 2 types of error - In Sample Error and Out of Sample Error.
- We get In Sample Error when we use the trained model on training dataset only.
- We get Out of Sample Error when we use the trained model on a new dataset.

STEP 7 Prediction

- Set up a pipeline to use the model in real life problems.
- Try to automate.
- Improve the model with time.

Interpolation and Extrapolation

1. Interpolation — Measuring the performance of model on previously seen data [training]
2. Extrapolation — Measuring the performance of model on previously unseen data [testing]

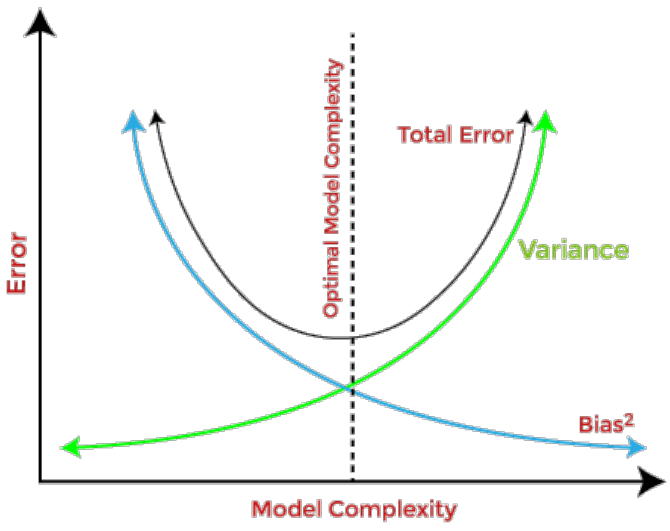
Bias-Variance Tradeoff

The expected test error is affected by error due to variance, bias and error due to randomness that the machine learning model cannot understand. The error due to randomness is irreducible. It is only possible to manage the error due caused by variance and bias.

1. Variance
    1. A model is trained on a training dataset and finds an optimal function that best fits the data. The amount by which the function will change if the training dataset is changed is known as variance.
    2. Example, the sample regression line is not same as the population regression line because the sample and population datasets are different.
    3. Increasing variance in a model leads to overfitting because now the model is too flexible and changeable and will map all datapoints rather than only find the underling pattern in the data.
    4. Low variance is seen in less flexible models whereas high variance is seen in more flexible models. If a few observations in the dataset are changed, the linear regression function will mildly see a difference. On the other hand, KNN model will try to follow data too closely and lead to a big change.
    5. Low variance models include linear regression, LDA and logistic regression.
    6. High variance models include decision trees, KNN, support vector machines.

2. Bias
    1. While making predictions, there comes a difference between the predicted values and the actual values and this difference is known as bias. It can be defined as the inability of the model to capture the true relationship between data points. Each algorithm begins with some amount of bias and then works on reducing it.
    2. Increasing bias leads to underfitting because the difference in actual and predicted values are higher which means that the model was unable to find the underlying pattern in the data. On the other hand, decreasing the bias leads to overfitting.
    3. High bias is seen for less flexible models whereas low bias for more flexible models. A linear model would never be able to capture the relationship between data points in a complicated dataset
    4. High bias models include linear regression, LDA and logistic regression
    5. Low bias models include decision trees, KNN, support vector machines
    6. The problem of high bias can be solved by using a machine learning model with high flexibility

3. Bias-Variance Tradeoff
    1. When we change the flexibility of the model, bias and variance change inversely. To solve this problem, we find the minimum sum of error due to variance and bias.
    2. High bias and low variance leads to underfitting
    3. Low bias and high variance leads to overfitting



Overfitting and Underfitting

1. Overfitting

1. Overfitting occurs when a machine learning model tries to cover all the data points in a dataset rather than only try to find the underlying pattern of the dataset.
2. It starts catching noise where noise is the irrelevant data present in the dataset that reduces the performance of the model.
3. The possibility of overfitting increases as we increase the training of a model. The more we train, there will be more chances of overfitting.
4. An overfitted machine learning model has high variance and low bias
5. Ways to control overfitting
    1. Cross validation
    2. Ensemble learning
    3. Dropping features
    4. Training with more data
    5. Regularisation methods like ridge and lasso

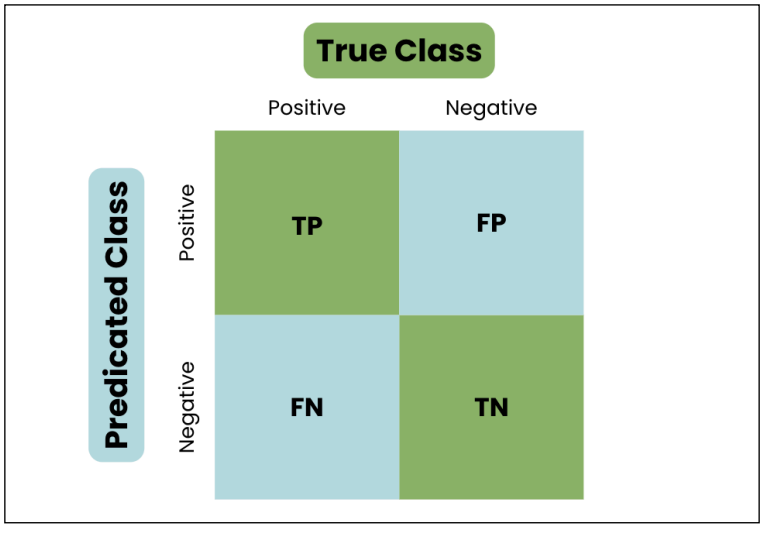
2. Underfitting

1. Underfitting occurs when a machine learning model is unable to capture the underlying trend of a dataset.
2. The model is unable to learn enough from the dataset and thus the predictions are inaccurate and unreliable
3. Underfitted model has high bias and low variance
4. Underfitting also occurs when we train a complicated dataset on a relatively simpler model
5. Ways to prevent underfitting
    1. Adding features and data
    2. Choosing relatively simpler models
    3. Increasing the training time of the model

- An ideal fit of a machine learning model lies between overfitting and underfitting.
- When we start training a model with a dataset, the accuracy is low in the starting [underfitted] and increases gradually. But there comes a time when the model is trained enough. After that, the model begins to capture noise and it leads to decrease in accuracy [overfitted]

Confusion Matrix

1. A table used to evaluate classification results of a machine learning model like logistic regression.
2. Components of confusion matrix [positive/negative refers to predicted output and true/false refers to whether that prediction is true or not]
    1. True positive - when predicted output is positive and correctly predicted
    2. True negative - when predicted output is negative and correctly predicted
    3. False positive [type 1 error] - when predicted output is positive but incorrectly predicted [actual output is negative]
    4. False negative [type 2 error] -when predicted output is negative but incorrectly predicted [actual output is positive]



3. Examples of errors
    1. Type 1 [FP] - doctor tells a man is pregnant when he cannot be.
    2. Type 2 [FN] - woman is pregnant but the doctor is telling the opposite.
4. Performance metrics like accuracy, precision, recall and F1 score are calculated from confusion matrix.

Hypothesis Testing

1. Hypothesis testing is a method in machine learning to check if a statement is True. This is done by assuming a null hypothesis [Ho] and an alternate hypothesis [Ha]
2. The null hypothesis states an assumption that there is no relationship. On the other hand, an alternate hypothesis contradicts it.
3. Key Terms
    1. Test statistic
        1. Any statistic calculated to decide whether to reject a null hypothesis or not.
        2. Tells us if their exists a relation or not.
        3. Examples are t-value or z-value.
        4. A critical value is set as a threshold for rejecting the null hypothesis.
    2. P-value
        1. A probability of confidence on the test statistic to reject the null hypothesis.
        2. If the p-value is below level of significance, then we can reject the null hypothesis.
        3. A level of significance is a threshold below which null hypothesis can be rejected. Level of significance is generally 0.05 or 5% which means that there is only a 5 percent chance that rejecting a null hypothesis is a mistake.

Curse of Dimensionality

1. The curse of dimensionality is a term used in machine learning to describe the problems that arise when working with data that has a high number of features, also known as dimensions.
2. The number of samples in a data should always be a lot more than the number of features. A basic reference generally considered is that as the number of features increase, the number of samples should increase at an exponential rate.
3. Challenges
    1. Data Sparsity — at higher dimensions, the data points are spread out so much making it hard to find trends and relationships.
    2. Computational Complexity — as the number of features increase, the number of computations needed to find meaningful trends and relationships increase taking up a lot more time and memory space.
    3. Overfitting — as the number of features increase, the number of samples needed to train a model also increases. Otherwise the model will not have enough samples to generalise.
    4. Distance Distortion — at higher dimensions, the data points are spread out so much that the distance metrics become less meaningful and less relevant. Hence, not good for models based on distance metrics.
    5. Complicated Algorithms — models to fight the curse of dimensionality are very hard to tune and complicated.
4. Solutions
    1. Dimensionality reduction
    2. Feature selection
    3. Regularisation
    4. etc