***Machine Learning Lecrure 1***

Definition of Machine Learning:

* Machine learning is defined as the science of programming computers so they can learn from data. It means that computers improve their performance on a task with experience without being explicitly programmed for the task

Why Machine Learning:

* Simplification: Instead of manually coding complex rule sets an ML system learns from examples.
* Adaptability: ML systems can adapt to fluctuating environments and continuously improve as new data arrives
* Problem Solving: They are particularly useful for complex problems where traditional methods fall short
* Insight Extraction: ML helps in uncovering patterns and insights from large complex datasets

Types of ML Systems:

* Based on Supervision
  + Supervised Learning: Uses labeled training data
  + Unsupervised Learning: Works with unlabeled data to find hidden patters
  + Reinforcement Learning: Involves an agent that learns by interacting with the environment receiving rewards or penalties
  + Also includes semi-supervised and self-supervised methods
* Based on Learning Process
  + Batch(Offline Learning): The model is trained on the full dataset at once before deployment
  + Online Learning: The model updates incrementally with incoming data, suitable for dynamic environments
* Based on Generalization Approach:
  + Instance-Based Learning: Makes predictions based on Similarity measures between new instances and stored examples
  + Model-Based Learning: Involves building a predictive model that is used to forecast outcomes

Core Technology:

* Training Set/Instance: The data examples used to train the model
* Model: The internal representation built during training that makes predictions
* Feature/Attribute: A measurable property of an instance used as an input
* Training Error vs Generalization Error: Metrics to evaluate performance on training data versus unsees (test) data
* Overfitting: When a model learns the training data too well (included noise) and fails to generalize
* Underfitting: When a model is too simple to capture the underlying structure in the data
* Regularization: Techniques that reduce model complexity (by limiting degrees of freedom) to combat overfitting
* Hyperparameters: Settings (like regularization strength) that control the learning process
* Holdout Validation: Splitting data into training, validation and test sets to fairly access model performance
* No Free Lunch Theorem: States that no single model is best for all problems; model performance depends on the data and assumptions made

***Quiz Questions and Answers***

1. Q: What is Machine Learning?  
   A: Machine Learning is the science of programming computers so they can learn from data and improve their performance on a task through experience without being explicitly programmed for each specific scenario
2. Q: What are the primary types of Machine Learning based on human supervision  
   A: The 3 Mentioned above
3. Q:How does supervised learning differ from unsupervised learning?  
   A: Supervised learning uses training data that includes labels or target values to guide the learning process, making it suitable for tasks like spam filtering and price prediction. Unsupervised learning, however works with unlabeled data to identify patters or grouping such as clustering similar visitors or detecting anomalies
4. Q: Provide an example of a supervised learning application discussed in the slides  
   A: The spam filter is a prime example where emails are classified as spam or not spam or labeled training data
5. Q: What is reinforcement learning and how does it work?  
   A: Reinforcement Learning is a type of ML where an agent interacts with an environment taking actions and receiving rewards or penalties. The goal is for the agent to learn an optimal policy that maximizes cumulative rewards over time.
6. Q: What is the difference between batch (Offline) learning an online learning?  
   A: Batch learning involves training a model on the entire dataset at once before deployment while online learning updates the model incrementally as new data becomes available, making it deal for situations where data flows continuously
7. Q: How do instance-based and model-based learning differ?  
   A: Instance-based learning makes predictions by comparing new data with stored examples using a similarity measure. In contrast model based learning builds a predictive model like a linear model based on the training data and then uses this model to make predictions on new instances
8. Q: What is overfitting and why is it a problem?  
   A: Overfitting occurs when a model learns the training data too precisely including the noise , and thus performs poorly on new unseen data. It is problematic because it limits the models ability to generalize to real world scenarios
9. Q: What role does regularization play in addressing overfitting?  
   A: Regularization simplifies a model by reducing its degrees of freedom thereby preventing the model from fitting the noise in the training data too closely. Hyperparameters control the level of regularization applied.
10. Q: What does the No Free Lunch Theorem imply about choosing an ML model?  
    A: It implies that no single ML model is universally best for all problems. The effectiveness of a model depends on the specific data and assumptions made, meaning that different problems might require different approaches
11. Q: Why is it important to split data into training, validation and test sets?  
    A: Splitting the data allows you to train the model on one portion (training set), fine tune and select models using another (validation set) and finally assess how well the model generalizes on unseen data (test data).This process helps avoid overfitting and ensures a fair evaluation of performance
12. Q: What is feature engineering and why is it critical in Machine Learning?  
    A: Feature engineering involves selecting, transforming or creating new features from raw data to improve the models performance. It is crucial because the choice and quality of features directly influence the models ability to learn relevant patterns.

***Machine Learning Lecture 2***

Overview and History:

* Reinforcement learning (RL) is machine learning paradigm where an agent learns by interacting with its environment
* It has roots in the 1950s with applications in game playing and recent breakthroughs like DeepMind’s Atari-playing system and AlphaGo have showcased its potential

The RL Agent and Policy:

* An RL agent is a software program that perceives its environment through sensors, takes actions according to a policy and learns from rewards (or penalites)
* The policy is the decision-making algorithm the agent uses, which can be deterministic or stochastic
* Policy search involves finding the best parameters for the policy using methods like brute-force search genetic algorithms or gradient based optimization

Exploration vs Exploitation

* A key challenge in RL is the balance between explotation (choosing actions known to yield high rewards) and exploration (trying new cations to discover potentially better rewards)
* Techniques such as the ε-greedy strategy help manage this trade-off by sometimes choosing a random action

The Credit Assignment Problem:

* In RL, rewards can be delayed making it difficult to know which actions contributed to a final outcome
* This is addressed by computing the action return (the discounted sum of future rewards using a discount factory ) and the action advantage which normalizes how much better or worse an action is compared to the average

Markov Decision Processes (MDP’s):

* MDPs extend Markov chains by incorporating actions and rewards. In an MDP the agent selects an action in each state and the system transition to a new state with an associated reward
* Optimal state Values (V\*(s)) represent the maximum expected discounted reward starting from state ‘s and methods like value iteration are used to compute these values

Q Learning and Temporal Difference Learning:

* Q-Learning is an off-policy Temporal Difference learning algorithm that estimates the Q-value (expected cumulative reward) for each state-action pair, even when transition probabilities and rewards are unknown
* It updates Q-values incrementally while the agent explores the environment gradually refining its policy
* The algorithm uses a learning rate (α) that is typically reduced over time for convergence

Neural Network Policies:

* Modern RL can employ neural networks to approximate policies where the network takes observations as inputs and outputs a probability distribution over actions
* This approach is particularly useful when dealing with high- dimensional or complex environment’s

***Quiz Questions and Answers***

1. Q: What is Reinforcement Learning?  
   A: Reinforcement Learning is a machine learning paradigm in which an agent learns to make decision by interacting with its environment receiving rewards or penalties and updating its policy to maximize cumulative rewards
2. Q: What is historical breakthrough’s have contributed to the popularity of RL?  
   A: Early RL was applied to game playing in the 1950’s. More recently DeepMind’s system in 2013 that learned to play Atari games from scratch and AlphaGo which beat world champions in Go (2016-2017) have been significant breakthroughs
3. Q: Define an RL agent and explain its role  
   A: As an RL agent is a software program that interacts with its environment by perceiving data through sensors selecting actions based on a policy and receiving rewards or penalties to learn optimal behavior over time
4. Q: What is a policy in the context of Reinforcement Learning?  
   A: A policy is the strategy or algorithm that the agent uses to decide which action to take in a given state. It can be deterministic or stochastic and is often refined via policy search methods
5. Q: How does the exploration vs exploitation dilemma affect an RL agent?  
   A: The dilemma involves balancing the need to exploit actions that yield known rewards with the need to explore new actions that might result in even better rewards. Techniques like the ε-greedy strategy help maintain the balance
6. Q:What is the credit assignment problem and how is it addressed in RL?  
   A: The credit assignment problem refers to the difficulty or determining which actions are responsible for received rewards especially when rewards are delayed. It is addressed by computing the action return the discounted sum of future rewards and by evaluating the action advantage
7. Q: What is Markov Decision Process (MDP) and what are its key components?  
   A: An MDP is a framework for modeling decision-making in situations where outcomes are partly random and partly controlled by an agent. Its key component’s include states actions transition probabilities and rewards. The goal is to maximize the cumulative discounted reward
8. Q: What is the purpose of the value iteration algorithm in MDP’s  
   A: The value iteration algorithm is used to compute the optimal state values by iteratively updating them based on expected discounted rewards, eventually converging to the optimal policy for the MDP.
9. Q: Explain Q-Learning and its significance in Reinforcement Learning?  
   A: Q-Learning is an off-policy Temporal Difference Learning algorithm that estimates the Q values for state action pairs. It allows an agent to learn an optimal policy through trial and error updating its estimates as it interacts with the environment even when the transition dynamics are unknown.
10. Q: What does it mean that Q-Learning is an off-policy algorithm?  
    A: Being off-policy means that the Q Learning algorithm learns the optimal policy independently of the agent’s current exploration strategy. This allows the agent to explore the environment using one strategy while learning the value of the best possible actions.
11. Q: How are neural network used in modern RL?  
    A: Neural Network policies take observations as input and output a probability distribution over possible actions enabling the agent to choose actions probabilistically. This method is effective in handling high-dimensional or complex environments where traditional approaches may struggle

***Machine Learning Lecture 3***

Overview of an End-to-End ML Project

* An ML project is a comprehensive process from understanding the problem to deploying and monitoring the model. The process includes planning , data acquisition, exploration, preparation, model training, fine tuning and finally launching the system

Look the Big Picture

* Problem Framing: Define the business objective, understand current solutions and decide how the ML model’s output will be used.
* Task identification: Determine the type of ML task (regression, classification, clustering, etc…) and the level of supervision (supervised, unsupervised, online vs batch).
* Performance Measures: Choose suitable metrics (e.g, RMSE and MAE for regression, accuracy or confusion matrices for classification, silhouette score or clustering).

Get the Data

* Data Acquisition: Gather data from various sources (CSV files, databases, compressed files).
* Initial Data Inspection: Examine the structure of the dataset, noting capped values, different scales and potential issues like heavy tails
* Test Set Creation: Prevent overfitting by splitting data into training and test sets. Consider random sampling versus stratified sampling (e.g. Keeping income distributions consistent) and use a fixed random seed (commonly 42) for reproducibility.

Explore and Visualize the Data

* Visualization: Use geographical plots, scatter matrices and correlation matrices to understand relationships between features and the target.
* Correlation Analysis: Compute Pearson’s r to identify linear relationships and experiment with feature combinations to uncover new insights.

Prepare the Data from ML Algorithms

* Data cleaning: Address missing values (e.g: using imputation with median or mean) and remove or correct errors
* Categorical Transformation: Convert categorical data using techniques like vocabulary mapping or one-hot encoding
* Numeric Transformation: Normalize numeric features via min-max scaling or z-score standardization; use transformation (e.g., log transformation or bucketisation) to handle skewed distributions.
* Pipelines and Custom Transformers: Build reproducible data transformation pipelines using tools like Scikit-Learns Pipeline and consider writing custom transformers for repeated use.

Select and Train a Model

* Initial Model Selection: Start with a baseline model (such as linear regression ) to gauge performance on the training set.
* Cross-Validation: Use techniques like k-fold cross-validation to access model performance reliably and to understand error variability.
* Underfitting vs Overfitting: Monitor model performance to ensure the model is complex enough to learn patterns without overfitting to the training data.

Fine Tune your Model

* Hyperparameter Tuning: Use grid search or randomized search (e.g., with GridSearchCV) to experiment with different hyperparameter combinations and analyze model errors to refine the model.
* Test Set Evaluation: After fine-tuning, evaluate the best model on the separate test set to measure its true generalization performance.

Launch, Monitor and Maintain

* Deployment: Save your model (e.g., with joblib) and deploy it as a web service, embedded in a website or on a cloud platform.
* Monitoring: Continuously track the models performance in production to detect issues like model rot.
* Maintenance: Regularly update datasets retrain the model, and adjust the system as new data arrives to ensure sustained performance

***Quiz Questions and Answers***

1. Q: What are the seven main stages of an end to end ML project?  
   A: The seven stages are:  
   1) Look at the Big Picture  
   2) Get the data  
   3) Explore and Visualize the Data  
   4) Prepare the Data for ML Algorithms   
   5) Select and Train a model   
   6) Fine-Tune your Model  
   7) Launch, Monitor and Maintain
2. Q: How do you frame the problem in the initial stage of an ML project?  
   A: Framing the problem involves defining the business objective understanding what the client aims to achieve, analyzing current solutions and determining the type of ML task (e.g., regression, classification) along with the appropriate performance measures.
3. Q: Why is creating a separate test set important in an ML project?  
   A: A separate test set ensures that you have unbiased data to evaluate the model’s generalization performance. It prevents data snooping and overfitting by keeping a portion of data unseen during training and tuning.
4. Q: What are some common methods for handling missing data during the data preparation stage?  
   A: Common methods include ignoring instances with missing values, dropping the feature altogether or using imputation techniques such as replacing missing values with the median or mean of the feature using tools like Scikit-Learns’s Simplelmputer
5. Q: Explain the difference between random sampling and stratified sampling for creating test sets.  
   A: Random sampling splits the dataset without considering any underlying distributions which may lead to sampling bias if the dataset is small or imbalanced. Stratified sampling divides the data into homogeneous subgroups (strata) and samples from each subgroup proportionally ensuring that important attributes (like median income) maintain their distribution in both training and test sets.
6. Q: What is the purpose of cross-validation in model training?  
   A: Cross-validation, such as k-fold cross-validation, helps estimate the model’s performance more reliably by splitting the training data into multiple folds. It provides not only the mean error but also the variability (standard deviation) of the error, reducing the risk of overfitting to a single validation set.
7. Q: How does grid search contribute to fine-tuning a model?  
   A: Grid search automates the process of hyperparameter tuning by exhaustively searching through a specified set of hyperparameter values. It uses cross-validation to evaluate the performance of each combination, enabling you to select combination that yield the best performance.
8. Q: What are the common normalization techniques used for numeric data transformation?  
   A: The common normalization techniques are:  
    Min-Max Scaling: Rescales the Features to a specific range (e.g., 0 to 1 or -1 to 1)  
    Z-Score Standardization: Transforms the data so that it has a mean of 0 and a standard deviation of 1. Additionally, log transformation or bucketisation may be used for handling skewed distributions.
9. Q: What techniques are available to transform categorical attributes for ML models?  
   A: Categorical attributes can be transformed using:  
    Vocabulary Mapping: Assigning a unique number to each category, which works best for ordered categories.  
    One-Hot Encoding: Converting each category into binary vectors (1 for the presence and 0 for the absence of a category), especially useful for unordered categories.
10. Q: Once a model is deployed, what strategies are used to ensure it continues performing well over time?  
    A: After deployment its crucial to monitor the models live performance by tracking key metrics and watching for model rot. Maintenance strategies include regular retraining with fresh data, automating retraining pipelines and setting up alerts for anomalies or shifts in data distributions.