Unit 2

20 August 2025 12:34

1. Types of Machine Learning Model:

1) **Supervised**
    - Classification (Categorical data)
        - Logistic Regression
            - Binary Classification
            - Multi Classification
    - Regression (Continuous data)
        - Linear
        - Non-Linear

2) **Unsupervised**
    - Clustering Models
    - Dimensionality Reduction/Feature Learning
    - Association/Pattern Mining
    - Anomaly/Outlier Detection

3) **Reinforcement**
    - Value Based Methods
    - Policy Based Methods
    - Model Based Methods

1.1 Process of Modelling
1. **Problem Definition:** Identify the problem and define objectives.
2. **Data Collection:** Gather relevant datasets.
3. **Data Preprocessing:** Clean data, handle missing values, normalize/scale features.
4. **Model Selection:** Choose an appropriate algorithm for the problem.
5. **Model Training:** Fit the model on training data.
6. **Model Validation:** Evaluate the model on unseen/validation data.
7. **Model Deployment:** Implement the model in production if satisfactory.

1.2 Training a Model
Definition: Learning the relationship between input features and target output from data.

- **Goal:** Minimize error (cost function) to improve prediction accuracy.
- **Techniques:**
    - Gradient Descent
    - Stochastic Gradient Descent (SGD)
    - Mini-batch Gradient Descent

1.3 Validating a Model
Purpose: Test model generalization on unseen data.

- **Techniques:**
    - Hold-out validation (split data into training and test sets)
    - K-Fold Cross-Validation
    - Leave-One-Out Cross-Validation (LOOCV)

- **Evaluation Metrics:**
    - Regression: MSE, RMSE, MAE, R^2
    - Classification: Accuracy, Precision, Recall, F1-score

1.4 Model Representation
- **Mathematical Representation:** Equation-based representation (e.g., $y = b_0 + b_1x$).
- **Graphical Representation:** Plots and diagrams showing model behavior (e.g., decision boundaries, regression line).
- **Programmatic Representation:** Implementation using ML frameworks (Python, R, TensorFlow, etc.).

2. Feature Engineering

Definition: Process of creating, transforming, and selecting relevant features from raw data to improve model performance. Better features result in more accurate models which have reduced training time.

2.1 Process of Feature Engineering

1. **Feature Creation:** Generate new features from existing data.
2. **Feature Transformation:** Modify features to improve model learning (e.g., scaling, normalization).
3. **Feature Selection:** Choose the most important features, reduce dimensionality, remove irrelevant/redundant features.
4. **Feature Evaluation:** Assess feature impact using statistical tests or model performance.

2.2 Feature Engineering Techniques

- **Handling Missing Values:** Imputation (mean, median, mode), or dropping.
- **Encoding Categorical Variables:** One-hot encoding, Label Encoding, Target Encoding.
- **Scaling & Normalization:** Min-Max Scaling, Standardization (Z-score).
- **Binning:** Converting continuous variables into categorical bins.
- **Polynomial Features:** Create interaction terms for non-linear relationships.

2.3 Feature Transformation

- Logarithmic transformation
- Square root or power transformation
- Standardization/Normalization
- Box-Cox transformation

2.4 Feature Selection

- **Filter Methods:** Statistical tests (Chi-square, ANOVA, correlation).
- **Wrapper Methods:** Recursive Feature Elimination (RFE), Forward/Backward Selection.
- **Embedded Methods:** Feature selection during model training (e.g., LASSO, Decision Tree importance).

2.5 Introduction to Feature Engineering Tools

- **Python Libraries:**
  - Pandas: Data manipulation
  - Scikit-learn: Preprocessing & selection functions
  - Feature-engine: Advanced feature engineering
  - Category Encoders: Encoding categorical variables
- **Other Tools:** Excel, Tableau (basic transformations), SQL (feature extraction)

Key Points:

Statistics

- **Variance:** how much a variable deviates from its mean in squared unit. Variance = $\sum(x\text{-}u)^2/n$
- **Standard Deviation:** deviation in original unit. SD = Sqrt(Variance(x))
- **Covariance:** tells the direction of the relationship but not strength as the magnitude depends on the value of the variables and not really their relationship.
- **Correlation:** standardized form of covariance which tells about both the direction and the strength of relationship between variables. Covariance(x,y) = $\sum(x\text{-}u)(y\text{-}v)/n$
- **Pearson's Correlation (r):** correlation metric for linear relationships. It assumes normal distribution of the data points. r = [Covariance(x,y)]/($\sigma x \ \ \sigma y$)

Loss Functions

- **Regression**
  - **MSE:** MSE is used to heavily penalized outliers. It is used in models working on a dataset where outliers can be interpreted believable which results in risk. Ex: Property Pricing.
  - **RMSE:** RMSE is just a more presentable and interpretable version of MSE which converts result unit into standard unit

instead of squared units.
    - **MAE:** MAE are used to be robust against outliers and treat all data points equally. Used in models working on datasets where outlier prediction can be distinguished easily. Ex: Taxi Fare
    - **R^2:** Shows the variance in predicted data on the regression line compared to the actual data.
- **Classification**
    - **Accuracy:** The proportion of correctly classified cases (both pos and neg) among the total number of cases. In other words, how often the model is correct overall.
        - $$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$
    - **Precision:** The proportion of correctly predicted positive cases among all predicted positive. In other words, of all predicted positive by the model how many were actually positive.
        - $$Precision = \frac{TP}{TP + FP}$$
    - **Recall:** The proportion of correctly predicted positive cases among all positive. In other words, of all actual positives how many did the model detect.
        - $$Recall = \frac{TP}{TP + FN}$$
    - **F1 score:** The harmonic mean of Precision & Recall to provide a single matrix that balances both.
        - $$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Optimization Algorithms
- **OLS:** It is used to minimize the MSE using direct calculus formula instead of iterative trial and adjust.
- **Iterative**
    - **Full Batch Gradient Descent:** Go through all the training sample and calculate the cumulative error, then go back and adjust weight and bias using learning rate and derivatives. Good for small training set.
    - **Stochastic Gradient Descent:** Calculates error on one randomly picked sample and then adjust weight and bias based on that, forwards the adjusted weight and bias to next random sample and repeats the process. Good for very big training set where too much computation is not possible.
    - **Mini-batch Gradient Descent:** Combines Batch Gradient and SGD, picks one small batch of samples randomly and adjust weight and bias, forwards the same weight-bias to next randomly picked mini batch and repeats the process.
    - **Epoch:** One cycle of calculating error on the training sample(s) and adjusting weight-bias accordingly.
    - **Forward Pass:** Passing the adjust weight-bias to the next training sample or batch of training samples.

Feature Engineering Techniques
- **Imputation**
    - **Mean:** works well with symmetric data such as height.
    - **Median:** better for skewed data such as salary.
    - **Mode:** used for categorical data such as favourite colour.
- **Encoding**
    - **One-hot Encoding:** converting categorical values into multiple binary columns with single 1 and rest 0s.
    - **Label Encoding:** the categorical values are assigned arbitrary numbers but this could lead to unintended ordinal relationship.
    - **Target Encoding:** replaces categories with the mean of the target variable for that category.
- **Scaling**
    - **Normalization (min-max):** rescaling data into a fixed range (0-1). Used when features are on different scales (Salary vs Age). Used when features have very different scales.
        - $$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$
    - **Standardization (z-score):** how far are each data from the average in their category in standard deviation unit. Used when distribution is roughly gaussian.
        - $$z = \frac{x - \mu}{\sigma}$$
- **Conversion**
    - **Binning:** grouping continuous values into bins to simplify the model. Age(10, 12, 25, 70) -> Bin(Child, Child, Adult, Old). Encoding follows Binning in linear models but optional for tree-based models.

Feature Transformation
- **Logarithmic** compressed wide gaps in data, reducing skewness and making patterns easier to model.

- **Square root:** to reduce skewness in data same as log but weaker.
- **Power:** to highlight the differences in large values.
- **Standardization/Normalization:** min-max and z-score.
- **Box-Cox:** Instead of arbitrarily choosing log or sqrt, Box-Cox finds the best power (λ) to stabilize variance and approximate normality. Only for positive data and when model assumed normal residuals (like Linear Regression).

Feature Selection

- **Chi-square:** Measures the dependency between categorical features and the target variable. Cat In -> Cat Out. Example: Input (Being Married) -> Output (Buying Product).
  - **Example**:
    - Predicting if a customer will buy (Yes/No) based on marital status (Single/Married).
    - Chi-square checks if "being married" is related to "buying".

- **ANOVA:** Analysis Of Variance, checks if means of different groups are significantly different. Cat In -> Con Out. Example: Input (School Board) -> Output (Marks).
  - **Example**:
    - Predict exam scores (numerical target) based on type of school (Govt/Private/International).
    - ANOVA tests if average scores differ across school types.

- **Correlation:** Measures linear relationship between two continuous variables. Con In -> Con Out. Example: Input (Study Hours) -> Output (Marks). Ranges from +1 to -1.

- **Recursive Feature Elimination (RFE):** Iteratively tests model and removes the least important feature until best result. Used with models that provide feature importance.
  - **Example**:
    - 100 medical test features to predict disease.
    - RFE trains model, drops weakest features, retrains, and continues until top 10 remain.

- **Forward/Backward Selection:** Start with no features, add one at a time (strongest feature) until no further improvement. Start with all features, remove one at a time (weakest feature).
  - **Example**:
    - Predict employee attrition.
    - Start with "Salary" (strongest predictor), add "Experience", then "Department", etc.

- **LASSO:** Least Absolute Shrinkage and Selection Operator. Adds a penalty (λ) proportional to the absolute value of the coefficients. This forces some coefficients to become zero, deselecting them from feature list.
  - **Example**:
    - Predict house prices with 50 features.
    - LASSO automatically removes irrelevant ones like "colour of mailbox" by shrinking coefficient to 0.

- **Decision Tree:** Tree-based models split data on features that maximize information gain. If a feature often appears in top splits, it's very important.
  - **Example**:
    - Predict loan approval.
    - Tree may first split on "Credit Score", then on "Income" → these are important features.

Method	Input Type	Target Type	Use Case
Chi-Square	Categorical	Categorical	Classification
ANOVA	Categorical	Continuous	Regression
Correlation	Continuous	Continuous	Regression
RFE	Any	Any	General-purpose
Forward/Backward	Any	Any	Small datasets
LASSO	Continuous	Continuous	High-dim regression
Decision Trees	Any	Any	Flexible, general

Flow Chart

Select ML Model (Supervised/Unsupervised) **-->** Imputation **-->** Scaling/Normalization **-->** Transformation (log/sqrt/pwr) **-->** Binning(if required) **-->** Encoding (one-hot/label/target) **-->** Feature Selection (CH-square/ANOVA/LASSO) **-->** Loss Function Selection (MAE/MSE/RMSE) **-->** Optimization Algorithm (OLS/Gradient Descent) **-->** Evaluation Metrics (R^2/Accuracy/Recall) **-->** Hyperparameter Tuning