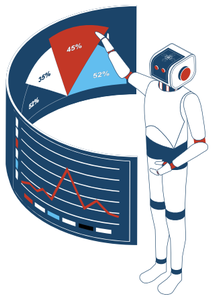
Codveda

Machine Learning



Content

1. About Us
2. Instructions
3. Submission
4. Task List

About Us

Welcome to Codveda Technology, where innovation meets excellence. Founded with a vision to empower businesses through cutting-edge IT solutions, we specialize in delivering tailored services that drive success in the digital era.

At Codveda, we offer a diverse range of services, including web development, app development, digital marketing, SEO optimization, AI/ML automation, and data analysis.

Our team of skilled professionals is committed to helping businesses unlock their full potential by providing innovative, scalable, and reliable solutions.

INSTRUCTIONS

- Update your LinkedIn profile with your achievements, including the offer letter and completion certificate. Mention and tag @Codveda in your posts.
- Use hashtags like #CodvedaJourney, #CodvedaExperience, and #FutureWithCodveda to showcase your progress and experiences.
- Share your project completion updates on LinkedIn, accompanied by a video explanation and the GitHub project repository link.
- You will be provided with three tasks. Select and complete any two tasks per level within your domain to fulfill the internship requirements.
- Submit your completed tasks via the Codveda submission form. Ensure all tasks are submitted within the allocated 1 month.

Level 1 (Basic)

Task 1: Data Preprocessing for Machine Learning

- Description: Preprocess a raw dataset to make it ready for machine learning.

Objectives:

- Handle missing data (e.g., filling with mean/median, dropping).
- Encode categorical variables (e.g., using one-hot encoding or label encoding).
- Normalize or standardize numerical features.
- Split the dataset into training and testing sets.
- Tools: Python, pandas, scikit-learn.

Level 1 (Basic)

Task 2: Build a Simple Linear Regression Model

- Description: Build a linear regression model to predict a continuous variable (e.g., house prices).

Objectives:

- Load a dataset and preprocess it.
- Train a linear regression model using scikit-learn.
- Interpret the model coefficients.
- Evaluate the model using R-squared and mean squared error (MSE).
- Tools: Python, pandas, scikit-learn.

Level 1 (Basic)

Task 3: Implement K-Nearest Neighbors (KNN) Classifier

- Description: Build a KNN classifier to classify data points into categories.

Objectives:

- Train a KNN model on a labeled dataset.
- Evaluate the performance using accuracy, confusion matrix, and precision/recall.
- Use different values of K and compare the results.
- Tools: Python, scikit-learn, pandas.

Level 2 (Intermediate)

Task 1: Logistic Regression for Binary Classification

- Description: Implement a logistic regression model to predict binary outcomes (e.g., whether a customer will churn).

Objectives:

- Load and preprocess the dataset.
- Train a logistic regression model using scikit-learn.
- Interpret model coefficients and the odds ratio.
- Evaluate the model using metrics such as accuracy, precision, recall, and the ROC curve.
- Tools: Python, pandas, scikit-learn, matplotlib.

Level 2 (Intermediate)

Task 2: Decision Trees for Classification

- Description: Build a decision tree classifier to predict a categorical outcome (e.g., predict species of flowers).

Objectives:

- Train a decision tree on a labeled dataset (e.g., the Iris dataset).
- Visualize the tree structure.
- Prune the tree to prevent overfitting.
- Evaluate the model using classification metrics such as accuracy and F1-score.
- Tools: Python, scikit-learn, pandas, matplotlib.

Level 2 (Intermediate)

Task 3: K-Means Clustering

- Description: Implement K-Means clustering to group unlabeled data into clusters (e.g., customer segmentation).

Objectives:

- Load a dataset and preprocess it (scaling).
- Apply K-Means clustering and determine the optimal number of clusters using the elbow method.
- Visualize clusters using 2D scatter plots.
- Interpret the clustering results.
- Tools: Python, scikit-learn, matplotlib, seaborn.

Level 3 (Advanced)

Task 1: Build a Random Forest Classifier

- Description: Implement a Random Forest model for classification on a complex dataset.

Objectives:

- Train a Random Forest model and tune hyperparameters (e.g., number of trees, max depth).
- Evaluate the model using cross-validation and classification metrics (precision, recall, F1-score).
- Perform feature importance analysis to identify the most important features in the dataset.
- Tools: Python, scikit-learn, pandas, matplotlib.

Level 3 (Advanced)

Task 2: Support Vector Machine (SVM) for Classification

- Description: Implement a Support Vector Machine (SVM) model for binary classification.

Objectives:

- Train an SVM model on a labeled dataset.
- Use different kernels (linear, RBF) and compare performance.
- Visualize the decision boundary.
- Evaluate the model using accuracy, precision, recall, and AUC.
- Tools: Python, scikit-learn, pandas, matplotlib

Level 3 (Advanced)

Task 3: Neural Networks with TensorFlow/Keras

- Description: Build a simple feed-forward neural network using TensorFlow/Keras for classification tasks (e.g., MNIST digit classification).

Objectives:

- Load and preprocess the dataset.
- Design a neural network architecture (input layer, hidden layers, and output layer).
- Train the model using backpropagation.
- Evaluate the model using accuracy and visualize the training/validation loss.
- Tools: Python, TensorFlow/Keras, pandas, matplotlib

How to Contact Us?

For additional information, kindly get in touch with our team.

@codveda

support@codveda.com

www.codveda.com