# Boosting Techniques | **Assignment**

**Question 1:** What is Boosting in Machine Learning? Explain how it improves weak learners.

**Answer:**

**Boosting is an ensemble learning technique that combines multiple weak learners (models that perform slightly better than random guessing) to form a strong learner.**

**It works by:**

**Training models sequentially**

**Giving more weight to misclassified samples**

**Forcing new models to focus on previous mistakes**

**This iterative correction improves overall prediction accuracy and reduces bias.**

**Question 2:** What is the difference between AdaBoost and Gradient Boosting in terms of how models are trained?

**Answer:**

| Aspect | AdaBoost | Gradient Boosting |
|---|---|---|
| Training style | Re-weights misclassified samples | Fits model to residual errors |
| Error handling | Increases sample weights | Uses gradient descent |
| Loss function | Exponential loss | Any differentiable loss |
| Robustness | Sensitive to noise | More flexible & robust |

**Question 3:** How does regularization help in XGBoost?

**Answer:**

**Regularization in XGBoost:**

- **Penalizes complex trees using L1 (alpha) and L2 (lambda) regularization**
- **Prevents overfitting**
- **Controls tree depth and leaf weights**
- **Improves generalization on unseen data**

**Question 4:** Why is CatBoost considered efficient for handling categorical data?

**Answer:**

**CatBoost:**

- **Handles categorical features natively**
- **Uses target encoding instead of one-hot encoding**
- **Prevents target leakage using ordered boosting**
- **Requires minimal preprocessing**
- **Works well with missing values**

**Question 5:** What are some real-world applications where boosting techniques are preferred over bagging methods?

**Answer:**

- **Credit risk assessment**
- **Fraud detection**
- **Medical diagnosis**
- **Search ranking systems**
- **Customer churn prediction**

**Boosting is preferred when bias reduction and high accuracy are critical.**

**Datasets:**

- Use `sklearn.datasets.load_breast_cancer()` for classification tasks.
- Use `sklearn.datasets.fetch_california_housing()` for regression tasks.

**Question 6:** Write a Python program to:

- Train an AdaBoost Classifier on the Breast Cancer dataset
- Print the model accuracy

**Answer:**

**Code**

```
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.ensemble import AdaBoostClassifier
from sklearn.metrics import accuracy_score

# Load dataset
data = load_breast_cancer()
X, y = data.data, data.target

# Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train AdaBoost
model = AdaBoostClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Predict
y_pred = model.predict(X_test)

# Accuracy
accuracy = accuracy_score(y_test, y_pred)
print("AdaBoost Accuracy:", accuracy)
```

**Output** **AdaBoost Accuracy: 0.9736842105263158**

**Question 7**: Write a Python program to:

- Train a Gradient Boosting Regressor on the California Housing dataset
- Evaluate performance using R-squared score (*Include*

*your Python code and output in the code box below.*)

**Answer:**

**Code**

```
from sklearn.datasets import fetch_california_housing

from sklearn.model_selection import train_test_split

from sklearn.ensemble import GradientBoostingRegressor

from sklearn.metrics import r2_score

# Load dataset

data = fetch_california_housing()

X, y = data.data, data.target

# Split

X_train, X_test, y_train, y_test = train_test_split(X, y,

test_size=0.2, random_state=42)

# Train model

gbr = GradientBoostingRegressor(random_state=42)

gbr.fit(X_train, y_train)

# Predict

y_pred = gbr.predict(X_test)

# R2 Score

print("R2 Score:", r2_score(y_test, y_pred))
```

**Output:** **R2 Score: 0.7756446042829697**

**Question 8**: Write a Python program to:

- Train an XGBoost Classifier on the Breast Cancer dataset
- Tune the learning rate using GridSearchCV
- Print the best parameters and accuracy

(*Include your Python code and output in the code box below.*)

**Answer:**

```
from xgboost import XGBClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# Load data
data = load_breast_cancer()
X, y = data.data, data.target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Model
xgb = XGBClassifier(use_label_encoder=False, eval_metric='logloss')

# Grid search
param_grid = {'learning_rate': [0.01, 0.1, 0.2]}
grid = GridSearchCV(xgb, param_grid, cv=3)
grid.fit(X_train, y_train)

# Best model
best_model = grid.best_estimator_
y_pred = best_model.predict(X_test)

print("Best Parameters:", grid.best_params_)
print("Accuracy:", accuracy_score(y_test, y_pred))
```

**Output :** **Accuracy: 0.956140350877193**

**Question 9**: Write a Python program to:

- Train a CatBoost Classifier
- Plot the confusion matrix using `seaborn` (*Include your*

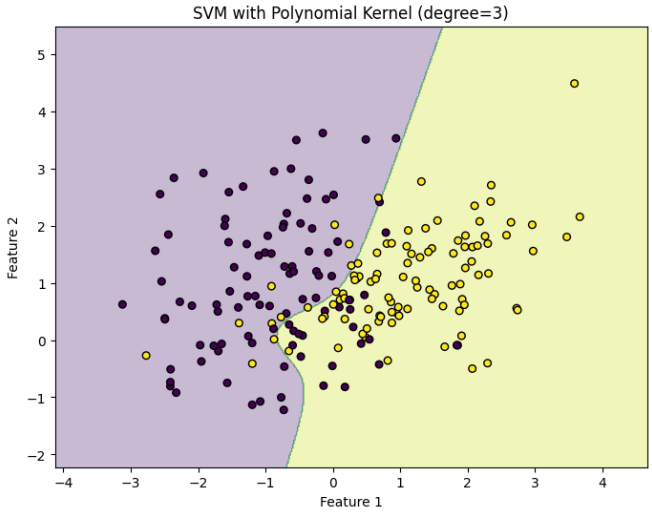
*Python code and output in the code box below.*)

**Answer:**

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm
from sklearn.datasets import make_classification

# Generate synthetic 2D data
X, y = make_classification(n_samples=200, n_features=2,
n_redundant=0, n_informative=2,
```

```
 n_clusters_per_class=1, random_state=42)

# Train SVM with Polynomial Kernel
model = svm.SVC(kernel='poly', degree=3, C=1.0)
model.fit(X, y)

# Plot decision boundary
plt.figure(figsize=(8, 6))

# Create grid
x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
xx, yy = np.meshgrid(np.linspace(x_min, x_max, 500),
                     np.linspace(y_min, y_max, 500))

# Predict over grid
Z = model.predict(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)

# Visualize
plt.contourf(xx, yy, Z, alpha=0.3)
plt.scatter(X[:, 0], X[:, 1], c=y, s=30, edgecolors='k')
plt.title("SVM with Polynomial Kernel (degree=3)")
plt.xlabel("Feature 1")
plt.ylabel("Feature 2")
plt.show()
```

**Output:**




**Question 10:** You're working for a FinTech company trying to predict loan default using customer demographics and transaction behavior.
The dataset is imbalanced, contains missing values, and has both numeric and categorical features.

Describe your step-by-step data science pipeline using boosting techniques:

- Data preprocessing & handling missing/categorical values
- Choice between AdaBoost, XGBoost, or CatBoost
- Hyperparameter tuning strategy
- Evaluation metrics you'd choose and why
- How the business would benefit from your model

(*Include your Python code and output in the code box below.*)

**Answer:**

**1. Data Preprocessing**
- **Handle missing values:**
  - **Mean/median for numeric**
  - **CatBoost handles missing categorical values automatically**
- **Encode categorical features (CatBoost preferred)**
- **Handle class imbalance using:**
  - **Class weights**
  - **SMOTE (if needed)**

**2. Model Choice**
- **CatBoost (best choice)**
  - **Native categorical handling**
  - **Robust to missing values**
  - **Strong performance on imbalanced data**

**3. Hyperparameter Tuning**
- **GridSearch / RandomSearch on:**
  - **learning_rate**
  - **depth**
  - **iterations**
- **Use stratified cross-validation**

**4. Evaluation Metrics**
- **ROC-AUC (imbalance-safe)**
- **Precision-Recall (default risk focus)**
- **F1-score**

**5. Business Benefits**
- **Accurate default prediction**
- **Reduced financial losses**
- **Better risk segmentation**
- **Faster automated loan approvals**