Regularization in Machine Learning - Q&A

Q: What is regularization in machine learning?

A: Regularization is a technique used to reduce overfitting by adding a penalty term to the loss function. This discourages the model from learning complex patterns that don't generalize well.

Q: Why do we use regularization?

A: To prevent overfitting, improve generalization, and ensure the model performs well on unseen data by penalizing large coefficients.

Q: What problems does regularization solve?

A: It solves overfitting, high variance, model complexity, and multicollinearity (in some cases like Lasso).

Q: What is overfitting and how does regularization help prevent it?

A: Overfitting happens when a model performs well on training data but poorly on test data. Regularization restricts the magnitude of model coefficients, reducing complexity.

Q: What is the difference between L1 and L2 regularization?

A: L1 (Lasso) adds the absolute value of coefficients to the loss. Can reduce some coefficients to zero. L2 (Ridge) adds the square of the coefficients. Shrinks coefficients but never zeroes them.

Q: What is Lasso Regression (L1)?

A: Lasso (Least Absolute Shrinkage and Selection Operator) is L1 regularization. It can eliminate irrelevant features by setting their coefficients to zero.

Q: What is Ridge Regression (L2)?

A: Ridge adds a penalty based on the squared values of the coefficients. It keeps all features but shrinks the coefficients.

Q: What is ElasticNet? How is it different from Lasso and Ridge?

A: ElasticNet combines L1 and L2. It balances between eliminating features (L1) and shrinking coefficients (L2). It's useful when there are many correlated features.

Q: When would you prefer Lasso over Ridge (or vice versa)?

A: Use Lasso when you suspect some features are irrelevant. Use Ridge when all features contribute and you want to shrink them, not eliminate.

Q: Why might Lasso eliminate some features (set their coefficients to zero)?

A: Because the L1 penalty promotes sparsity. It forces some coefficients to become exactly zero, removing less important features.

Q: How is the loss function modified in Ridge and Lasso regression?

A: Ridge: Loss = RSS + * sum(). Lasso: Loss = RSS + * sum(||).

Q: What is the role of the alpha parameter in regularization?

A: Alpha () controls how strongly the model penalizes large coefficients: High alpha = more regularization, Low alpha = less regularization.

Q: How does changing alpha affect bias and variance?

A: Increase alpha Higher bias, lower variance. Decrease alpha Lower bias, higher variance. It's a bias-variance trade-off.

Q: What happens when alpha = 0 in Lasso or Ridge?

A: Regularization has no effect model becomes ordinary least squares regression.

Q: What does regularization do to the coefficients of a model?

A: It shrinks them to reduce complexity. In Lasso: Some become zero. In Ridge: All shrink but remain non-zero.

Q: How do you implement Lasso or Ridge in scikit-learn?

A: from sklearn.linear_model import Lasso, Ridge

lasso = Lasso(alpha=0.1)

lasso.fit(X_train, y_train)

ridge = Ridge(alpha=1.0)

ridge.fit(X_train, y_train)

Q: How do you tune the alpha parameter?

A: Use cross-validation:

```
from sklearn.linear_model import LassoCV

lasso_cv = LassoCV(alphas=[0.01, 0.1, 1])

lasso_cv.fit(X_train, y_train)

print(lasso_cv.alpha_)
```

Q: What is cross-validation and how is it useful with regularization?

A: Cross-validation splits data into k folds, trains on k1 folds, tests on 1 fold. Helps find the best alpha and ensures the model generalizes well.

Q: What metrics would you use to evaluate a regularized model?

A: R score, Mean Absolute Error (MAE), Mean Squared Error (MSE), RMSE. Evaluate both training and validation scores.

Q: Can you use regularization with models other than linear regression?

A: Yes. It's used in Logistic Regression, Neural Networks (L2 = weight decay), SVM (via parameter C), Decision Trees (via pruning).