Classwork 1: Diagnosing Overfitting

Learning Objective

You will **diagnose overfitting and multicollinearity** in a housing price prediction model to **recommend whether regularization is needed** for production deployment.

Business Context

You're a data scientist at HomeLend Financial, a mortgage lending company. The modeling team built a multiple regression model to predict house prices for automated loan valuations. The model performed well during development, but when deployed to new markets, it's overvaluing properties by 15% on average.

Your task: Diagnose whether the model is overfitting and identify which predictors are causing instability.

Dataset: housing_prices.csv

- **750 houses** from various neighborhoods
- **Target:** sale_price (in dollars)
- **Predictors:**
  - `sqft_living`: Interior square footage
  - `sqft_lot`: Lot size square footage
  - `bedrooms`: Number of bedrooms
  - `bathrooms`: Number of bathrooms
  - `floors`: Number of floors
  - `waterfront`: Waterfront property (0/1)
  - `view`: Quality of view (0-4 scale)
  - `condition`: House condition (1-5 scale)
  - `grade`: Construction quality (1-13 scale)
  - `sqft_above`: Square footage above ground
  - `sqft_basement`: Square footage of basement
  - `yr_built`: Year built
  - `yr_renovated`: Year renovated (0 if never)
  - `zipcode`: Zip code (categorical)

- `lat`: Latitude coordinate
- `long`: Longitude coordinate

Time Required

25-30 minutes

Setup

1. Open RStudio
2. Create a new R script
3. Navigate to: `lecture_3_advanced_regression/classwork_1/`
4. Open `template.R`
5. Work through exercises 1-12 in order

Exercises Overview

1. **Load and explore data** (2 min) - Understand the business problem
2. **Create train/test split** (2 min) - Prepare for validation
3. **Fit full model** (2 min) - Build multiple regression
4. **Calculate training performance** (2 min) - Measure fit on training data
5. **Calculate test performance** (2 min) - Measure generalization
6. **Compare train vs test** (2 min) - Quantify overfitting
7. **Calculate VIF values** (3 min) - Identify multicollinearity
8. **Identify problematic predictors** (2 min) - Find high VIF features
9. **Calculate correlation matrix** (3 min) - Understand feature relationships
10. **Plot learning curves** (3 min) - Visualize overfitting
11. **Test coefficient stability** (3 min) - Check for instability
12. **Make recommendation** (2 min) - Business decision

Success Criteria

By the end, you will have:

- [ ] Quantified the train/test performance gap
- [ ] Identified predictors with multicollinearity (VIF > 5)
- [ ] Visualized the overfitting problem with learning curves
- [ ] Demonstrated coefficient instability

- [ ] Made a data-driven recommendation to stakeholders

Common Issues & Solutions

Issue: Package not installed (car, caret)

- **Solution:** Run `install.packages(c("car", "caret"))` first

Issue: VIF function error

- **Solution:** Make sure car package is loaded: `library(car)`

Issue: createDataPartition not found

- **Solution:** Load caret package: `library(caret)`

Issue: Correlation matrix too large to read

- **Solution:** Use `corrplot` package for visualization (optional)

Issue: Test RMSE lower than training RMSE

- **Solution:** This can happen by chance. Look for consistency across multiple splits.

Resources

- **Lecture slides:** Slides 1-30 (Batch 1)
- **R documentation:**
  - `?vif` for variance inflation factors
  - `?cor` for correlation matrix
  - `?createDataPartition` for train/test splitting
- **Further reading:**
  - James et al., "An Introduction to Statistical Learning", Chapter 6 (Regularization)
  - VIF interpretation guide

Reflection Questions

After completing all exercises, consider:

1. **Business Decision:** Based on your diagnostics, would you deploy this model to production?
2. **Feature Selection:** Which predictors show the most evidence of multicollinearity?
3. **Next Steps:** What would you recommend to improve model performance?

Remember: The goal is not just to calculate metrics, but to understand what they mean for business decisions. Every diagnostic tells you something about whether to trust this model with real money.

Ready? Open template.R and begin with Exercise 1!