Handling the Multicollinearity Issue

Exploring Placement Data: Finding the Multicollinearity Problem

1. Initial Analysis (Correlation):

- **Objective:** Relate academic scores to salary.
- **Finding:** All academic scores (ssc_p, hsc_p, degree_p, mba_p) were **highly correlated with each other**.

2. Identifying Multicollinearity (The Issue):

- **What is it?** High correlation **among** the predictor variables.
- **Why is it bad?** It "confuses" a standard regression model, making the results unstable and coefficients unreliable.

3. Measurement (VIF):

- Our **VIF scores were extremely high** (e.g., degree_p > 112, mba_p > 99), confirming severe multicollinearity.

Solution: Ridge Regression
Solving High VIF with L2 Regularization

The Challenge: We needed to use all variables but could not use a standard Linear Regression.

Solution to Use: Ridge Regression (L2 Regularization):

•**Why Ridge?** This special model is designed to stabilize results despite severe multicollinearity.

•**How it Works (The "Manager" Analogy):** Ridge introduces a **penalty** (called **Alpha**) that acts like a strict manager. This penalty forces the model to keep all variable coefficients small and stable.

•**Effect:** The high VIF is minimized or Controlled, allowing the model to run reliably while retaining all five academic predictors.

Explained :What is Ridge Regression (L2 Regularization)?

Think of a standard linear regression model as a person trying to give credit to five team members (your variables) for a single project (the salary).

The Problem (High VIF): Your "team members" (ssc_p, hsc_p, degree_p, etc.) all did very similar work. The model gets confused and can't decide who to give credit to. It might give one person a *massive* positive credit (a huge coefficient) and another a *massive* negative credit to balance it out. The model becomes unstable and unreliable.

The Solution (Ridge): Ridge Regression acts like a manager who says, "I know you all worked together, so I'm going to *penalize* any huge, outsized claims of credit." It adds a "penalty" (the L2 penalty) that forces the model to keep all coefficients as small as possible.

This penalty shrinks the coefficients of the correlated predictors, making them more stable and preventing any single one from "inflating." The result is a reliable model, even when VIF is high.

Results & Insights
Model Performance

1. Model Results (After Ridge Regression):

- **Best Alpha Found:** 17.47 (The model tested penalties up to 100. It chose 17.4753 as the optimal penalty. This is a significant penalty (since the test started near 0), confirming that a strong penalty was absolutely required to overcome the high VIF and stabilize the model's coefficients.).
- 17.47 is the sweet spot. It represents the optimal balance where the coefficients are stable and reliable, without penalizing them so much that the model loses its ability to fit the data well.
- **Model Score (R-squared): 0.2797**

2. The Key Insight:

- An R2 of **0.2797** means our model, using all academic scores, can only explain about **28%** of the variation in salary.

- **Conclusion:** Academic performance is a factor, but it leaves **72%** of the salary difference **unexplained**.

```
# Define a range of alphas (penalties) to test.
# test 100 values between 0.1 and 100.
alphas_to_test = np.logspace(-1, 2, 100)

# Create the RidgeCV model
# cv=None uses efficient Leave-One-Out Cross-Va
ridge_model = RidgeCV(alphas=alphas_to_test, cv

# Fit the model
ridge_model.fit(X_train, y_train)

# 1. This is the best alpha (penalty) the model
print(f"Best alpha found: {ridge_model.alpha_:.

# 2. Check the model's score (R-squared) on the
test_score = ridge_model.score(X_test, y_test)
print(f"Model R-squared on test data: {test_sco
```

```
Best alpha found: 17.4753
Model R-squared on test data: 0.2797
```