Rubric

Part 1: Penguin (30 points)

Task 1 (6 points)

1. (+2) Correctly uses `sklearn.model_selection.train_test_split`
2. (+1) Sets `test_size=0.3` or equivalent 30% split
3. (+1) Sets `random_state=2025`
4. (+1) Prints number of training samples
5. (+1) Prints number of test samples

Task 2 (8 points)

1. (+2) Uses only the first two columns (`bill_length_mm`, `bill_depth_mm`) as features
2. (+2) Fits `LinearRegression` model on training data
3. (+2) Makes predictions on test data using the same features
4. (+1) Reports RMSE on training data
5. (+1) Reports RMSE on test data

Task 3 (10 points)

1. (+2) Separates numerical features (`bill_length_mm`, `bill_depth_mm`) correctly
2. (+2) Separates categorical features (`species`, `sex`) correctly
3. (+2) Uses `OneHotEncoder` fitted on training data
4. (+2) Fits `LinearRegression` model on combined numerical and one-hot encoded features
5. (+1) Reports RMSE on training data for combined model
6. (+1) Reports RMSE on test data for combined model

Task 4 (6 points)

1. (+1) Prints coefficients from Task 2 model
2. (+1) Prints coefficients from Task 3 model

3. (+2) Explains why `bill_depth_mm` coefficient changed from negative to positive (must reference visualizations)
4. (+2) Explains why correlation does not prove causation using this example

---

Part 2: Diabetes (30 points)

Task 1 (4 points)

1. (+1) Uses `train_test_split` with correct parameters (`test_size=0.3`, `random_state=2025`)
2. (+1) Splits into `X_train, X_test, y_train, y_test`
3. (+2) Prints number of training samples (309) and testing samples (133)

Task 2 (6 points)

1. (+2) Computes mean of `y_train` using `np.mean()` or equivalent
2. (+2) Creates baseline predictions (training mean repeated for all test samples)
3. (+2) Calculates and reports RMSE between baseline predictions and `y_test`

Task 3 (8 points)

1. (+3) Uses `LinearRegression()` from sklearn, fits on `(X_train, y_train)`
2. (+3) Makes predictions on both training (`y_train_pred`) and test (`y_test_pred`) data
3. (+2) Reports RMSE for both training and test predictions

Task 4 (12 points)

1. (+3) Uses cross-validation or train/validation split (not test data) for hyperparameter selection
2. (+3) Finds alpha that yields ≥3 zero coefficients (coefficients < 0.0001)
3. (+2) Final model test RMSE ≤ 110% of Task 3 linear model test RMSE
4. (+3) Reports final test RMSE and identifies which 3+ features have zero coefficients
5. (+1) Explains why the chosen alpha was selected

Part 3: DIY Least Squares (40 points)

Task 1 (10 points)

1. (+2) Implements constant baseline by computing mean of `y_train`
2. (+2) Creates constant predictions (training mean repeated for all test samples)
3. (+3) Uses `LinearRegression()` from sklearn, fits on training data, predicts on train/test sets
4. (+3) Calculates and reports MSE for both baselines on both train and test sets

---

Task 2 (16 points)

1. (+4) Implements `__init__` method initializing weights to `None` and storing `random_state`
2. (+4) Implements `fit` method using normal equation with `np.linalg.solve` (not matrix inversion)
3. (+4) Implements `predict` method returning `X @ self.weights` (vectorized dot product)
4. (+1) Adds error handling in `predict` (raises error if weights are `None`)
5. (+3) Includes `verbose` parameter in `fit` that prints debugging info when `True`

Task 3 (6 points)

1. (+1) Creates instance of DIY LinearRegression and fits on training data
2. (+1) Makes predictions on training and test sets using DIY model
3. (+2) Calculates and reports MSE on both train and test sets
4. (+2) Compares DIY results to sklearn baseline and reports essentially identical performance

---

Task 4 (8 points)

1. (+2) Uses `PolynomialFeatures(degree=2, include_bias=False)` (or equivalent manual implementation)
2. (+2) Fits DIY LinearRegression on polynomial features, predicts on train/test sets
3. (+2) Calculates and reports MSE for polynomial model on both train and test sets
4. (+2) Provides written analysis discussing overfitting behavior (lower train MSE, higher test MSE than linear model)