**Technical Report: Model Training Fundamentals with Custom Dataset**

**Task Title:** Model Training Fundamentals with Custom Dataset  
**Dataset Used:** House Prices - Advanced Regression Techniques (from Kaggle)  
**Model Type:** Supervised Regression  
**Goal:** Predict housing sale prices using various linear models

### 2. Dataset Description

* **Name:** House Prices - Advanced Regression Techniques
* **Source:** Kaggle (https://www.kaggle.com/c/house-prices-advanced-regression-techniques)
* **Features:** 80+ variables about residential homes in Ames, Iowa
* **Target Variable:** SalePrice
* **Type:** Regression task

**Preprocessing Steps:** - Removed rows with missing target values - Filled numeric missing values with median - Categorical columns filled with mode - One-hot encoding applied to categorical variables - Features standardized using StandardScaler

### 3. Model Implementation Summary

The following models were implemented and compared: - Linear Regression - Polynomial Regression (degree=2) - Ridge Regression (L2 regularization) - Lasso Regression (L1 regularization) - SGD Regressor (Stochastic Gradient Descent)

### 4. Model Comparison Table

| Model | RMSE | R² Score | Best Alpha | Training Time | Notes |
| --- | --- | --- | --- | --- | --- |
| Linear Regression | 75,694,361,898.02 | -746,987,320,796.42 | — | Fast | Likely exploded due to scaling |
| Polynomial (deg=2) | 35,695.24 | 0.8339 | — | Medium | Captured non-linearity |
| Ridge Regression | 31,098.78 | 0.8739 | alpha = 100 | Medium | Good generalization |
| Lasso Regression | 44,661.83 | 0.7399 | alpha = 100 | Medium | Slight underfitting |
| SGD Regressor | 946,345.43 | -115.76 | eta0 = 0.01 | Fast | Diverged or poor convergence |

### 5. Learning Curve Analysis

* **Linear Regression:** Erratic due to unstable coefficients from large feature dimensions.
* **Polynomial Regression:** Showed potential overfitting at high training set sizes but still useful.
* **Ridge & Lasso:** Improved generalization. Ridge performed better than Lasso.

Plots were generated using learning\_curve() for these models and illustrated model performance over increasing training data sizes.

### 6. Feature Importance

Using Lasso regression, feature importance was inferred from non-zero coefficients. Top predictive features included: - OverallQual - GrLivArea - GarageCars - TotalBsmtSF - YearBuilt

These were visualized using bar plots (optional).

### 7. Analysis & Interpretation

#### Why certain models performed better:

* **Linear Regression** failed due to numerical instability or multicollinearity.
* **Polynomial Regression** captured non-linear patterns, improving fit.
* **Ridge Regression** penalized large coefficients and performed best in generalization.
* **Lasso** shrunk more coefficients to zero, possibly removing useful features.
* **SGD Regressor** failed due to poor convergence (wrong learning rate or data scaling).

#### Overfitting & Polynomial Degree:

* Degree=2 significantly improved the model but also increased computation and risk of overfitting.
* Learning curves supported this by showing higher variance in validation scores.

#### Practical Applications:

* Real estate agencies can estimate property values.
* Banks and insurers can assess property-related financial risk.
* Investment platforms can analyze housing market trends.

### 

### **Model Performance Analysis**

#### Why Certain Algorithms Performed Better or Worse

Each model behaves differently depending on the data's structure, dimensionality, and noise:

* **Linear Regression** performed **extremely poorly** (very high RMSE, extremely low R²). This is most likely due to multicollinearity or numeric instability after one-hot encoding, especially since the dataset had a high number of features post-encoding. Without regularization, the model overfits to noise and fails to generalize.
* **Polynomial Regression (Degree=2)** improved the performance by capturing **non-linear interactions** between features. It offered a good balance between fit and complexity. However, it also increased the number of features significantly, which could lead to **overfitting** if not controlled.
* **Ridge Regression** applied **L2 regularization**, penalizing large coefficients, which helped prevent overfitting. This resulted in a **strong generalization performance** with the best R² score (≈ 0.87). Ridge handled multicollinearity effectively.
* **Lasso Regression** used **L1 regularization**, which forces many coefficients to become exactly zero. While this helps with **feature selection**, it may also remove relevant features, slightly reducing performance compared to Ridge. It showed signs of **underfitting**, with a lower R² score (≈ 0.74).
* **SGD Regressor** performed poorly, producing a very high RMSE and negative R². This typically indicates that the model **failed to converge**. Likely causes include an inappropriate learning rate (eta0=0.01) or sensitivity to feature scale/noise in the high-dimensional dataset. Tuning the learning rate schedule and applying feature normalization carefully could improve results.

#### Impact of Polynomial Degree on Overfitting

Polynomial regression allows the model to fit more complex relationships between features and the target, but it also increases the number of features exponentially.

* In this task, **degree=2** polynomial expansion significantly improved model performance (R² ≈ 0.83), showing that the data has **non-linear patterns** that linear models cannot capture.
* However, adding too many polynomial features leads to:
  + **Overfitting**, where the model fits training data too closely and performs poorly on unseen data.
  + **Increased computational cost** and slower training.
* A balance must be found — often by tuning the degree with cross-validation.

#### Practical Applications of the Trained Models

The trained models — especially Ridge and Polynomial Regression — have real-world use cases in:

1. **Real Estate Platforms**  
   Predict housing prices for buyers/sellers using house features like square footage, location, and amenities.
2. **Mortgage and Loan Assessment**  
   Banks and lenders can use the model to **evaluate home values** for mortgage approvals.
3. **Property Investment Analysis**  
   Investors can use price predictions and feature importance to determine **high ROI areas or property types**.
4. **Smart Home Valuation Tools**  
   Integration into apps/websites that offer users an instant estimated value of their home based on current market data.

### 8. Final Notes

* Polynomial and Ridge models offered the best balance between performance and generalization.
* Proper preprocessing and hyperparameter tuning are crucial.
* Gradient Descent models require careful learning rate scheduling and normalization.

**Submitted Files:** - house\_price\_model.ipynb (code) - train.csv (dataset) - report.docx (this document) - Visuals (learning curves, feature plots)