1. Introduction and Problem Statement

Translation efficiency (TE) measures how effectively an mRNA transcript is translated into protein. Ribosomal protein genes (RPL/RPS) are tightly regulated and play essential roles in cellular function, making them a meaningful target for modeling. Predicting TE from genome-wide gene expression data is a challenging machine learning problem due to high dimensionality, potential nonlinear relationships, and correlated biological signals.

The goal of this project is to build an end-to-end machine learning system that predicts the **translation efficiency of ribosomal protein genes** using publicly available ribosome profiling and RNA-seq data. The work will include full data preprocessing, exploratory analysis, model training, and performance evaluation. A central objective is to compare classical ML approaches with more advanced nonlinear and representation-learning methods to determine which techniques best capture patterns associated with TE.

This project fits directly within the example project types listed in the assignment. It applies advanced classical algorithms such as XGBoost and Support Vector Regression to a high-dimensional tabular dataset, exactly as suggested in the prompt. It also incorporates neural network–based modeling and dimensionality-reduction methods, aligning with the example encouraging exploration of neural architectures and modern deep learning techniques. Overall, the project follows the intended spirit of comparing multiple ML methods on a real dataset of personal interest.

2. Data Sources

The project will use publicly available datasets that contain both **RNA-seq expression values** and **ribosome profiling–derived translation efficiency measurements**. Suitable sources include:

- Ribosome profiling datasets from the **NCBI Gene Expression Omnibus (GEO)**
- Supplementary datasets from published Ribo-seq studies of human cell lines (e.g., HEK293T)

The dataset will be treated as raw input. All preprocessing will be performed within this project, including:

- Importing and aligning RNA-seq and TE tables

- Handling missing data
- Filtering or transforming low-expression genes
- Normalizing and scaling features
- Creating final feature/target matrices
- Train/validation/test dataset splits

3. High-Level Methods, Techniques, and Technologies

The project will evaluate multiple machine learning approaches to determine which methods best predict TE.

Classical Regression Models

- Linear Regression
- Ridge, Lasso, ElasticNet
- Principal Component Regression (PCR)

Advanced Classical / Nonlinear Models

- Random Forest Regressor
- Gradient Boosting (XGBoost)
- Support Vector Regression (RBF kernel)

Dimensionality Reduction & Representation Learning

- PCA or Kernel PCA
- Autoencoders for latent feature learning

Neural Network Approaches

- Fully connected neural networks for multi-gene TE prediction
- Multi-task regression architectures

Technologies & Tools

- Python, NumPy, pandas
- scikit-learn
- XGBoost
- PyTorch or TensorFlow/Keras
- Matplotlib/Seaborn for visualizations

Evaluation Metrics

- RMSE
- R^2
- Predicted vs. actual TE scatterplots
- Feature importance analyses
- Latent space visualizations

4. Products to Be Delivered

The final deliverables will include:

A. Code & Workflow

- Jupyter notebooks for:

  - Data preprocessing
  - Exploratory data analysis
  - Model development and evaluation
  - Dimensionality-reduction analysis
  - Comparisons of all models tested

B. Visual Outputs

- Predicted vs. actual TE plots
- PCA/latent space visualizations
- Feature importance figures
- Summary tables of model performance