1. Regression for Forecasting Asset Expected Returns

Linear Models

- **Ridge Regression**
  - Uses L2 regularization (penalty on sum of squared coefficients) to handle multicollinearity among financial indicators
  - Controls overfitting by shrinking coefficient magnitudes toward zero while keeping all variables in the model
  - Particularly effective when predictors are highly correlated, as is common with financial indicators
  - Empirical evidence shows ridge portfolios can achieve annualized returns of 23.28% with volatility of 10.62%

- **Lasso Regression**
  - Employs L1 regularization to create sparse models by setting some coefficients exactly to zero
  - Performs variable selection, identifying only the most predictive firm-level characteristics
  - Research shows different predictors are selected depending on time horizon - profitability and customer-momentum dominate short-run, while book-to-market and sectoral effects dominate long-run

- **Elastic Net**
  - Combines both L1 and L2 penalties (Lasso and Ridge) to optimize the bias-variance tradeoff
  - Better suited for big datasets with many potential predictors, as it both selects variables and handles multicollinearity
  - Helps prevent overfitting while maintaining predictive power for stock returns

Tree-based Models

- **Random Forest**
  - Ensemble learning method combining multiple decision trees to predict stock trends
  - Can achieve remarkably high accuracy (85-95%) in predicting stock price trends
  - Effectively captures non-linear relationships in financial data that linear models miss
  - Experimental results demonstrate superior performance with prediction accuracy of 80-99% for longer time window
  - Portfolio constructed using Random Forest predictions showed lower volatility (10.44%) and better Sharpe ratio (2.13) compared to benchmark portfolios

- **XGBoost (Extreme Gradient Boosting)**

    - Gradient boosting framework designed for speed and performance that builds trees sequentially
    - Multi-objective optimization with optimal weights (OW-XGBoost) can balance return and risk metrics
    - Testing on Chinese A-share data showed OW-XGBoost achieved 30.09% return rate during backtesting with 61.05% annualized return
    - Outperforms traditional models in controlling risk while achieving returns

Deep Learning Models

- **LSTM (Long Short-Term Memory)**
    - Specialized recurrent neural network designed for sequential data processing
    - Captures both short and long-term dependencies in time series financial data
    - Significantly outperforms traditional models like ARIMA, with MAE of 175.9 vs. 462.1 and accuracy of 96.41% vs. 89.8%
    - Better handles market volatility by minimizing larger deviations in predictions

- **Transformer Models**
    - Initially developed for NLP but adapted for financial time series prediction
    - Uses self-attention mechanisms to identify dependencies across different time points
    - Shows encouraging results on S&P500 data, particularly for volatility prediction
    - Can model complex patterns in financial data without requiring feature engineering

Feature Engineering

- **Price-Momentum Indicators**
    - Technical analysis tools that measure the strength or weakness of a stock's price trend
    - Help determine if a trend is likely to continue or reverse
    - Serve as important predictive variables in machine learning models

- **Volatility/Volume Indicators**
    - Measure degree of variability in returns over time, often through standard deviation
    - Higher volatility typically associated with higher risk
    - Important for options pricing and risk assessment

- **Fundamental and Macroeconomic Variables**
    - Financial ratios, GDP acceleration, inflation, unemployment, and consumer sentiment have shown predictive power
    - Research identifies variables across five categories (financial ratios, macro, labor market, housing, and sentiment/leverage)

    - Combined models using multiple variable types outperform single-factor models

2. Regression for Forecasting Asset Covariance Matrix

Simple Estimators

- **Rolling-Window**
    - Calculates sample covariance over the last W observations
    - Formula: $\Sigma t = \alpha t * \Sigma(\tau=t\text{-}M \text{ to } t\text{-}1)\ r\tau r\tau^T$ where αt is the normalizing constant
    - Simple to implement but equally weights all observations within the window
    - Less responsive to recent market changes than exponential weighting

- **EWMA (Exponentially Weighted Moving Average)**
    - Assigns exponentially decreasing weights to older observations
    - Formula: $\Sigma t = \alpha t * \Sigma(\tau=1 \text{ to } t\text{-}1)\ \beta^{t-1-t} r\tau r\tau^T$ where β is the forgetting factor
    - Often expressed in terms of half-life $H = -\log(2)/\log(\beta)$
    - More responsive to recent market changes than equal-weighted approaches
    - Can be iterated (IEWMA) with separate half-lives for volatility (Hvol) and correlation (Hcor)

Multivariate GARCH Models

- **BEKK-GARCH**
    - Named after Baba, Engle, Kraft, and Kroner, allows interactions among conditional variances and covariances
    - Formula: $Ht = C + A'\varepsilon t\text{-}1\varepsilon t\text{-}1'A + B'Ht\text{-}1B$ (where matrices capture ARCH and GARCH effects)
    - ARCH coefficients measure impact of previous innovations, while GARCH coefficients examine persistence of return volatility
    - Empirical studies show different markets exhibit different degrees of volatility clustering (Hong Kong showed greatest degree)
    - Requires fewer parameters than other multivariate GARCH specifications

- **DCC-GARCH (Dynamic Conditional Correlation)**
    - Nonlinear combination of univariate GARCH models with time-varying cross-equation weights
    - Separates volatility modeling (diagonal elements) from correlation modeling (off-diagonal elements)
    - Decomposes $Ht = DtRtDt$ where Dt contains standard deviations from univariate GARCH and Rt is the correlation matrix
    - More parsimonious parameterization than BEKK, making it suitable for larger asset sets
    - Estimates parameters via maximum likelihood with log-likelihood function based on multivariate normal distribution

Factor & Shrinkage Methods

- **Statistical Factors (PCA)**
  - Reduces dimensionality while preserving maximum variance in the data
  - In finance, first principal component typically represents market movement, subsequent components reflect sectoral effects
  - Helps identify main factors driving correlations among assets
  - Enables factor-based portfolio construction sensitive to specific market factors

- **Dynamic PCA**
  - Extends traditional PCA to capture time-varying covariance structures
  - Estimates leading eigenvectors dynamically to capture time-varying information
  - Particularly useful for high-dimensional settings where number of variables (p) is comparable to or larger than sample size (n)
  - Provides smooth estimates over entire time periods in an integrative manner

- **Ledoit-Wolf Shrinkage**
  - Shrinks sample covariance toward a structured target using optimal shrinkage intensity
  - Formula derived to minimize expected quadratic loss
  - Particularly effective for large-dimensional covariance matrices
  - Improves conditioning of the estimator, making it more suitable for portfolio optimization
  - Implementation allows splitting covariance matrix into blocks for memory optimization

Sparse/Regularized Estimators

- **Graphical Lasso**
  - L1-penalized covariance estimator that imposes sparsity in precision matrix
  - Regularization parameter α controls sparsity level - higher α means more regularization and sparser inverse covariance
  - Available in two solver modes: coordinate descent (cd) or LARS, with cd preferred for numerical stability
  - Particularly useful for very sparse underlying graphs where $p > n$
  - Controls accuracy through tolerance parameters and maximum iterations

- **Banding/Thresholding**
  - Applies banding or thresholding operations to sample covariance matrices
  - Banding sets all elements more than k positions away from diagonal to zero
  - Thresholding sets elements below a certain magnitude to zero regardless of position
  - Helps control estimation error in high-dimensional settings
  - Can be combined with other methods like Dynamic Principal Component Analysis for improved performance