Big Data in Finance

Lecture 6: Model Interpretability

Dr Daniele Bianchi

Spring 2026

Contents

Overview

We have built powerful models—Random Forests with 500 trees, XGBoost with hundreds of weak learners—that outperform simple benchmarks. But can we answer these questions: *Why* was this loan application denied? *What* is driving the model's return forecast? *How* would the prediction change if income increased? *Which* features matter most for this specific case?

Complex models offer **better predictions** but less **transparency**. In finance, transparency is often required—by regulators, risk managers, clients, and model developers themselves.

Part I develops the theoretical toolkit for interpretability: feature importance methods (tree-based and permutation), Partial Dependence Plots (PDP), Individual Conditional Expectation (ICE) curves, SHAP values, and LIME.

Part II applies these methods to predicting the equity premium using the Goyal-Welch-Zafirov (GWZ) dataset with 25 macroeconomic predictors. We use interpretability tools to understand *why* the model makes its predictions and whether we can trust them.

1 Why Interpretability Matters

Consider a loan applicant denied credit by your ML model. The applicant asks: "Why was I rejected?"

Without interpretability: "The model said no." You cannot explain the decision, creating potential legal liability, customer frustration, and regulatory non-compliance.

With interpretability: "Your debt-to-income ratio of 45% was the primary factor." This provides clear, actionable feedback. The decision is defensible, the customer knows how to improve, and you satisfy regulatory requirements.

In many jurisdictions, lenders must provide **specific reasons** for adverse credit decisions (e.g., the US Equal Credit Opportunity Act).

In an investment context, when your ML model recommends a large position in a stock, the portfolio manager asks: "Why does the model like this stock?" This matters for:

- **Risk management:** Is the model picking up a real signal or noise?
- **Due diligence:** Can we explain this to clients and compliance?
- **Model debugging:** Is the model using sensible features?
- **Regime changes:** Will the signal persist in different conditions?

A particular concern: the model might heavily weight a feature that is actually a **data artifact** (e.g., ticker symbol encoded as a number). Portfolio managers are more likely to follow model recommendations they understand.

1.1 Who Needs Interpretability?

Stakeholder	Why They Need Interpretability
Regulators	Ensure models are fair, not discriminatory, and compliant with rules
Risk Managers	Understand model behavior in stress scenarios
Customers	Right to explanation for decisions affecting them
Model Developers	Debug models, identify data leakage, improve performance
Senior Management	Approve models they can understand and defend
Auditors	Verify model logic and governance

Different stakeholders need different *types* and *levels* of explanation.

1.2 The Accuracy-Interpretability Tradeoff

The **traditional view** holds that we must sacrifice accuracy for interpretability. Linear regression has transparent coefficients but limited predictive power; neural networks capture complex patterns but are opaque.

The **modern view** offers a better solution: use **post-hoc interpretability methods** to explain complex models. We keep our high-performing Random Forest or XGBoost, then apply interpretation methods *after* training.

1.3 Two Approaches to Interpretability

1. **Inherently Interpretable Models:**
   - Linear regression: Coefficients show direction and magnitude
   - Logistic regression: Odds ratios
   - Single decision tree: Visual flowchart
   - Limited complexity $\Rightarrow$ Limited accuracy
2. **Post-Hoc Interpretation of Complex Models:**
   - Keep your Random Forest or XGBoost
   - Apply interpretation methods *after* training
   - Get the best of both worlds

This lecture focuses on post-hoc methods: feature importance, PDP/ICE, SHAP, and LIME.

1.4 Global vs. Local Interpretability

Global interpretability: "How does the model work *in general*?" Which features matter most overall? What is the average effect of each feature? How do features interact? *Methods:* Feature importance, PDP.

Local interpretability: "Why did the model make *this specific* prediction?" Why was this loan denied? What drove this stock's forecast? How can this applicant improve? *Methods:* SHAP, LIME, ICE.

Both matter. Global interpretability supports model validation and debugging. Local interpretability provides individual explanations.

2 Feature Importance Methods

Feature importance answers: which features contribute most to the model's predictions? We care because it helps us identify key drivers, detect potential data leakage (suspicious features), simplify models by removing unimportant features, and communicate what the model has learned.

There are two main approaches: **model-specific** importance built into tree-based models, and **model-agnostic** permutation importance.

2.1 Tree-Based Feature Importance

For Random Forest and XGBoost, recall that regression trees split to reduce variance within nodes. A good split creates child nodes where outcomes are more similar. The improvement equals variance before split minus weighted variance after.

Feature importance idea:

- Each split reduces variance by some amount
- Features that create *bigger* variance reductions are more important
- Sum reductions across all splits using that feature, across all trees

$$\text{Importance}(X_j) = \sum_{\text{splits on } X_j} (\text{Variance reduction from split})$$

Advantages: Fast (computed during training), available via `model.feature_importances_`.

Disadvantages: Biased toward **high-cardinality** and **correlated** features.

Issue 1 (Cardinality bias): Features with more unique values get more chances to split. A random ID number might appear "important."

Issue 2 (Correlated features): If X_1 and X_2 are correlated, the tree might use either. Importance gets "split" between them arbitrarily. Both appear less important than they really are. Example: FICO score and credit rating in a credit model might both show 15% importance, but removing *either* would hurt performance significantly.

2.2 Permutation Feature Importance

Idea: How much does performance *drop* when we break the relationship between a feature and the target?

Algorithm:

1. Train model, compute baseline performance (e.g., R^2, AUC)
2. For each feature X_j:
   (a) Randomly shuffle values of X_j (breaks relationship with Y)
   (b) Compute performance with shuffled X_j

(c) Importance = Baseline − Shuffled performance

3. Repeat multiple times and average

If shuffling X_j hurts performance a lot, then X_j is important.

Advantages:

- **Model-agnostic:** Works for any model
- Computed on *held-out data* (reflects generalization)
- No cardinality bias
- Intuitive interpretation

Caveats:

- **Correlated features:** Still problematic. Shuffling one leaves the other intact, so the model can still predict using the correlated feature. Both features appear less important.
- **Computational cost:** Need to re-evaluate model many times
- **Extrapolation:** Shuffling can create unrealistic combinations

Rule of thumb: Use permutation importance, but be cautious with correlated features.

3 Partial Dependence Plots

Feature importance tells us *which* features matter. But we also want to know *how* the prediction changes as a feature changes. Is the relationship linear? Monotonic? Non-monotonic? Are there threshold effects or interactions?

Partial Dependence Plots (PDP) show the marginal effect of a feature on predictions.

3.1 The Partial Dependence Function

Definition:

$$\text{PD}(x_j) = \frac{1}{n}\sum_{i=1}^{n} \widehat{f}(x_j, X_{-j}^{(i)})$$

In words:

1. Pick a value x_j for the feature of interest
2. For *every* observation in the data, set $X_j = x_j$ but keep other features unchanged
3. Compute predictions for all these modified observations
4. Average the predictions

Repeat for a grid of values of x_j and plot.

3.2 Step-by-Step Example

Suppose we have 3 observations and want PD for DTI ratio:

Obs	DTI	Income	Rate	Original Pred
1	20%	\$50k	12%	$P(\text{def}) = 0.15$
2	35%	\$80k	15%	$P(\text{def}) = 0.25$
3	25%	\$45k	10%	$P(\text{def}) = 0.18$

To compute PD at DTI = 30%, set DTI to 30% for *all* observations:

Obs	DTI	Income	Rate	New Pred
1	**30%**	\$50k	12%	$P(\text{def}) = 0.22$
2	**30%**	\$80k	15%	$P(\text{def}) = 0.20$
3	**30%**	\$45k	10%	$P(\text{def}) = 0.24$

Then PD(DTI = 30%) = $(0.22 + 0.20 + 0.24)/3 = 0.22$. Repeat for DTI = 10%, 15%, 20%, ..., 50% to build the full curve.

3.3 Interpreting PDPs

A typical PDP for DTI ratio might show:

- Default probability increases with DTI ratio (as expected)
- The relationship is **non-linear**: steep increase after DTI > 25%
- This reveals a threshold effect the model has learned

Two-way PDPs can show interactions between two features. A heatmap might reveal that high DTI is less problematic for high-income borrowers (they can afford the debt service).

3.4 Implementation

```
from sklearn.inspection import PartialDependenceDisplay

# Single feature
PartialDependenceDisplay.from_estimator(
    model, X_train, features=['dti_ratio']
)

# Two features (interaction)
PartialDependenceDisplay.from_estimator(
    model, X_train, features=[('dti_ratio', 'income')]
)
```

Use training data to compute PDP, but interpret with caution for regions with few observations.

3.5 The Independence Assumption

PDP assumes features are **independent**. When computing PD for DTI = 50%, we combine it with *all* observed income levels, including very high incomes. But in reality, high DTI and high income rarely occur together!

Result: PDP may show predictions for **unrealistic** feature combinations. Interpret cautiously when features are correlated.

4 SHAP Values

SHAP = SHapley Additive exPlanations

SHAP is the gold standard for model interpretation. It provides:

- **Local** explanation for each prediction
- Additive: contributions sum to prediction

- Based on game theory (Shapley values)
- Theoretically grounded with desirable properties

The key question SHAP answers: "For this specific prediction, how much did each feature contribute to pushing the prediction above or below the average?"

4.1 Shapley Values: Game Theory Foundation

Shapley values are a Nobel Prize-winning concept from cooperative game theory (Lloyd Shapley, 1953).

The game theory analogy:

- **Players** = Features
- **Game** = Prediction task
- **Payout** = Prediction minus average prediction
- **Question:** How to fairly divide the "payout" among players?

Shapley's answer: Give each player their **average marginal contribution** across all possible coalitions (orderings).

4.2 SHAP: Intuition

Consider a loan applicant:

- Average prediction (baseline): $P(\text{default}) = 0.20$
- This applicant's prediction: $P(\text{default}) = 0.35$
- Difference to explain: $0.35 - 0.20 = 0.15$

SHAP decomposes this into feature contributions:

$$\underbrace{0.20}_{\text{base}} + \underbrace{0.08}_{\text{DTI}} + \underbrace{0.05}_{\text{rate}} + \underbrace{0.04}_{\text{history}} - \underbrace{0.02}_{\text{income}} = 0.35$$

Key property: Contributions sum *exactly* to the prediction!

4.3 Formal Definition

For feature j and observation x:

$$\phi_j(x) = \sum_{S \subseteq \{1,\ldots,p\} \setminus \{j\}} \frac{|S|!(p - |S| - 1)!}{p!} \left[f(S \cup \{j\}) - f(S) \right]$$

In words:

- Consider all possible subsets S of features (excluding j)
- For each subset, compute the marginal contribution of adding j
- Weight by the number of orderings that produce this subset
- Average across all subsets

SHAP values are **additive**: $f(x) = \phi_0 + \sum_{j=1}^{p} \phi_j(x)$, where ϕ_0 is the average prediction.

4.4 Desirable Properties

SHAP is the *only* method satisfying all of:

1. **Local accuracy:** Contributions sum to prediction
2. **Missingness:** Features not in the model get $\phi_j = 0$
3. **Consistency:** If a feature's contribution increases in the model, its SHAP value doesn't decrease
4. **Symmetry:** Features that contribute equally get equal SHAP values

These properties uniquely determine the Shapley value formula.

4.5 Computing SHAP: The Computational Challenge

Problem: Exact Shapley values require summing over 2^p subsets! With 20 features: $2^{20} \approx 1$ million subsets per prediction.

Solutions:

Method	Description
TreeSHAP	Exact, fast algorithm for tree-based models. $O(TLD^2)$ complexity.
KernelSHAP	Model-agnostic approximation using weighted regression. Slower but works for any model.
DeepSHAP	Approximation for neural networks using DeepLIFT.

Practical advice: Use TreeSHAP for Random Forest / XGBoost. Use KernelSHAP for other models.

4.6 SHAP in Python

```
import shap

# For tree-based models (fast)
explainer = shap.TreeExplainer(xgb_model)
shap_values = explainer.shap_values(X_test)

# For any model (slower)
explainer = shap.KernelExplainer(model.predict, X_train)
shap_values = explainer.shap_values(X_test)

# Visualizations
shap.summary_plot(shap_values, X_test)
shap.force_plot(explainer.expected_value,
    shap_values[0], X_test.iloc[0])
```

4.7 SHAP Visualizations

Summary plot (bar): Shows mean absolute SHAP value for each feature—global importance.

Summary plot (beeswarm): Each dot = one observation. Color = feature value (red=high, blue=low). Shows importance AND direction of effect.

Force plot: Explains a single prediction, showing how each feature pushes the prediction above or below the baseline.

Dependence plot: SHAP value vs. feature value, showing the functional relationship and potential interactions (via color-coding).

5 Choosing the Right Method

Question	Method
Which features matter overall?	Permutation importance
How does a feature affect predictions?	PDP
Why was *this* prediction made?	SHAP force plot
Global importance + direction?	SHAP summary plot

Typical workflow:

1. Permutation importance for feature selection
2. SHAP summary plot for global understanding
3. PDP for specific feature relationships
4. SHAP force plots for individual explanations

6 Application: Predicting the Equity Premium

Part II applies interpretability methods to a real forecasting problem: predicting the equity premium using macroeconomic variables.

6.1 Why Does This Matter? The Economics

Market timing is a trillion-dollar question:

- A 1% improvement in monthly Sharpe ratio $\approx$ billions in alpha for large funds
- Pension funds, endowments, sovereign wealth funds all engage in tactical allocation
- Even small predictability can justify active management fees

Two competing views in finance:

1. **Efficient Markets (Fama):** Returns are unpredictable; any apparent predictability is spurious or compensation for risk
2. **Behavioral Finance (Shiller):** Predictability exists due to investor irrationality, sentiment, and slow-moving capital

Our goal: Use interpretability to understand *what economic forces* drive ML predictions—is it risk or mispricing?

6.2 Economic Theories of Return Predictability

1. Present Value Identity (Campbell & Shiller, 1988)

$$\underbrace{dp_t}_{\text{dividend yield}} \approx \sum_{j=1}^{\infty} \rho^{j-1} \left(\underbrace{r_{t+j}}_{\text{future returns}} - \underbrace{\Delta d_{t+j}}_{\text{dividend growth}} \right)$$

High dividend yield ⇒ either high future returns OR low future dividend growth.

2. Habit Formation (Campbell & Cochrane, 1999)

- Risk aversion varies with consumption relative to habit
- In recessions: high risk aversion ⇒ high expected returns

3. Investor Sentiment (Baker & Wurgler, 2006)

- Optimism/pessimism affects asset prices
- High sentiment ⇒ overpricing ⇒ low future returns

Different predictors capture different economic mechanisms!

6.3 The GWZ Dataset

The Goyal-Welch-Zafirov dataset contains:

- 825 monthly observations (April 1953 – December 2021)
- Target: `MktRf` (market excess return, %)
- 25 predictor variables

Predictor variables by economic category:

Valuation Ratios (Present Value Theory):

- `d_p`, `d_y`: Dividend-price ratio, dividend yield
- `e_p`: Earnings-price ratio (inverse P/E)
- `b_m`: Book-to-market ratio
- **Theory:** High valuation ratios ⇒ high expected returns

Interest Rates & Spreads (Macro/Credit Risk):

- `tbl`: T-bill rate (monetary policy stance)
- `tms`: Term spread (yield curve slope—recession predictor)
- `dfy`, `dfr`: Default spread/return (credit risk premium)
- **Theory:** Wide spreads ⇒ high risk aversion ⇒ high expected returns

Technical/Sentiment (Behavioral):

- `svar`: Stock variance (fear/uncertainty)
- `ntis`: Net equity issuance (market timing by firms)
- `skvw`: Return skewness (crash risk)
- `dtoy`, `avgcor`: Turnover, correlation (sentiment proxies)

Target variable statistics:

Mean	0.64%
Std Dev	4.22%
Min	−22.09%
Max	16.19%

Key observation: the **signal-to-noise ratio is low**. Monthly returns are dominated by unpredictable noise.

6.4 Model Training

Critical: Use chronological split, not random!

```
# Time-series split: first 70% for training
split_idx = int(len(df) * 0.7)  # = 577
X_train, X_test = X[:split_idx], X[split_idx:]
# Training: 1953-04 to 2001-04 (n=577)
# Testing:  2001-05 to 2021-12 (n=248)
```

Random splits cause **look-ahead bias**—the model sees "future" data during training, inflating apparent performance.

Random Forest configuration:

```
rf = RandomForestRegressor(
    n_estimators=500,
    max_depth=5,  # Shallow trees to avoid overfitting
    random_state=42
)
```

Results:

- In-sample R^2: 58%
- Out-of-sample R^2: 14%

Is 14% OOS R^2 good? For equity premium prediction, this is **excellent**. Goyal & Welch (2008) found most predictors achieve *negative* OOS R^2. Campbell & Thompson argue that even $R^2 = 0.5\%$ is economically meaningful.

6.5 Feature Importance Results

Feature	Tree-Based Rank	Permutation Rank
dtoy	1	1
skvw	2	2
dfr	7	3
svar	8	4
ltr	3	10
avgcor	4	11

Key observations:

- `dtoy` and `skvw` are top 2 in both methods—**robust finding**
- `dfr` (default spread) and `svar` (volatility) rank higher in permutation—they capture crisis/stress periods (important OOS)
- `ltr` and `avgcor` may be overfit in-sample

Recommendation: Trust permutation importance for model validation.

6.6 Economic Interpretation of Feature Importance

Surprising result: Traditional valuation ratios (d_p, e_p, b_m) rank low!

What dominates instead?

1. **dtoy (Detrended Turnover):** Behavioral/sentiment indicator
   - High turnover = high trading activity = investor enthusiasm
   - Related to Baker & Wurgler's sentiment measures
2. **skvw (Skewness):** Crash risk / tail risk
   - Investors dislike negative skewness (lottery preferences)
   - High crash risk ⇒ investors demand higher returns
3. **dfr, svar:** Stress/volatility indicators
   - Capture time-varying risk aversion
   - Counter-cyclical: high in crises, low in booms

Implication: The model relies more on **behavioral/risk** variables than on **valuation** ratios. This suggests returns are driven by sentiment and risk premia, not just mean reversion.

6.7 Partial Dependence Results

PDP for dtoy (Detrended Turnover):

- Low turnover → bearish
- High turnover → very bullish
- **Strongly nonlinear** effect above dtoy = 0.97

PDP for skvw (Market Skewness):

- Negative skewness (crash risk) → bearish
- Positive skewness → bullish
- Monotonic relationship

6.8 SHAP Analysis

For each test observation:

$$\underbrace{\hat{y}_i}_{\text{prediction}} = \underbrace{0.63}_{\text{baseline}} + \underbrace{\sum_{j=1}^{25} \phi_j^{(i)}}_{\text{SHAP contributions}}$$

SHAP summary plot: Mean |SHAP| shows average impact. `dtoy` contributes ±1.78 percentage points on average—huge given baseline is only 0.63%!

Case Study: April 2007 (Bullish Signal)

- Prediction: +5.44%
- dtoy = 1.00 (maximum turnover) pushed prediction up by +3.34 percentage points
- **Economic interpretation:** Sentiment-driven rally, classic late-cycle behavior

Case Study: January 2009 (Bearish Signal)

- Prediction: −8.86%

- dtoy = 0.61 (low turnover) contributed −3.92 pp
- skvw = −0.04 (negative skewness) contributed −2.81 pp
- **Economic interpretation:** Flight to quality, deleveraging, fire sales

The model correctly identified the financial crisis through economic mechanisms!

6.9 Model Validation: Do Effects Match Theory?

Feature	High →	Theory	Mechanism
dtoy	Bullish	Behavioral	Sentiment/momentum
skvw	Bullish	Risk-based	Low crash risk
avgcor	Bullish	Behavioral	Herding/optimism
dfr	Bearish	Risk-based	Credit stress
svar	Bearish	Risk-based	Uncertainty
d_y	Bullish	Valuation	Mean reversion

All effects align with theory! This increases confidence that:

- The model captures real economic relationships, not spurious patterns
- Predictions can be trusted when economic conditions match training data
- We can anticipate when the model might fail (regime changes)

Interpretability lets us **tell an economic story** about each prediction!

7 Portfolio Implications

7.1 From Prediction to Portfolio

Simple market timing strategy:

- If $\hat{r}^e_{t+1} > \bar{r}$: Overweight equities
- If $\hat{r}^e_{t+1} < \bar{r}$: Underweight equities (or go to cash)

Interpretability helps with:

1. **Position sizing:** Large positions only when drivers are clear
2. **Risk management:** Know which factors could reverse the signal
3. **Regime awareness:** Recognize when the model is in/out of regime
4. **Client communication:** Explain why you're bullish/bearish

Example: "We're overweight equities because market turnover is elevated and crash risk is low. Key risk: if credit spreads widen, our signal would flip bearish."

7.2 When to Trust (and Distrust) the Model

Trust the model when:

- SHAP decomposition tells a coherent economic story
- Dominant features (dtoy, skvw) are the main drivers
- Current regime resembles training data

Be cautious when:

- Prediction is driven by minor/unstable features
- Features are at extreme values (extrapolation)
- Market structure has changed (e.g., rise of passive investing)
- SHAP contributions nearly cancel out (low conviction)

7.3 Limitations and Model Risk

Structural breaks: The model trained on 1953–2001 may not capture post-crisis dynamics: quantitative easing, rise of passive investing, algorithmic trading.

Crowding: If many investors use similar predictors, predictability may diminish. GWZ variables are well-known in academic literature.

Extrapolation: PDP shows extreme nonlinearity for dtoy > 0.97. What happens at dtoy $= 1.05$, never seen in training? Behavior is undefined.

Omitted variables: Fed policy, geopolitics, ESG considerations—not in the model but affect markets.

Interpretability doesn't eliminate these risks but makes them **visible**, allowing informed decisions about model deployment.

Key Concepts

- **Feature Importance:** Measures which variables contribute most to predictions; tree-based (fast, biased) vs. permutation (slower, reliable).
- **Partial Dependence Plot (PDP):** Shows average effect of a feature on predictions, marginalizing over other features.
- **ICE Curves:** Individual-level version of PDPs; reveals heterogeneity and interactions.
- **SHAP Values:** Game-theoretic decomposition of predictions into additive feature contributions; theoretically principled and exactly additive.
- **TreeSHAP:** Efficient algorithm for computing exact SHAP values for tree-based models.
- **LIME:** Local surrogate model approach; fits simple model in neighborhood of observation.
- **Global vs. Local Interpretability:** Global explains overall model behavior; local explains individual predictions.
- **Model Validation:** Using interpretability to check whether learned relationships match economic theory.

References

Interpretability Methods:

- Molnar, C. (2022). *Interpretable Machine Learning.* Available free online. [Recomm.]
- Lundberg, S. & Lee, S. (2017). A Unified Approach to Interpreting Model Predictions. *NeurIPS.* [Supp.]

Return Predictability:

- Goyal, A. & Welch, I. (2008). A Comprehensive Look at The Empirical Performance of Equity Premium Prediction. *Review of Financial Studies*. [Recomm.]
- Campbell, J. & Thompson, S. (2008). Predicting Excess Stock Returns Out of Sample. *Review of Financial Studies*. [Supp.]

Economic Theory:

- Baker, M. & Wurgler, J. (2006). Investor Sentiment and the Cross-Section of Stock Returns. *Journal of Finance*. [Supp.]
- Campbell, J. & Cochrane, J. (1999). By Force of Habit. *Journal of Political Economy*. [Supp.]