Predictive Models

- **Inference**: understand the relationship between predictors and an outcome
- **Prediction**: modeling for prediction which prioritizes prediction accuracy

Model Assessment

Coefficient of Determination (**R^2**): the proportion of the variation in Y that is explained by variation in X, measures fit, not correctness

Measure	Formula
Total Sum of Squares	$\sum(y_i - y)^2$
Residual Sum of Squares (RSS) / Sum of Squared Errors (SSE)	$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
Mean Squared Error (MSE)	SSE/n
Root mean squared error (RMSE)	$\sqrt{MSE}$
Average Squared Error (ASE)	SSE/n
Residual Standard Error (RSE)	$\sqrt{\frac{\sum(y_i - \hat{y}_i)^2}{n-p}}$
Coefficient of Determination (R^2) $0 \le R^2 \le 1$ 0: No changes in Y explained by X 1: All changes in Y explained by X (perfect fit)	$1 - \frac{RSS}{TSS}$

Odds: p=0.75, odds=3, success is 3 times more than failure

Odds ratios (OR): OR>1, more likely in group 1

$$Odds\ Ratio = \frac{Odds\ in\ Group\ 1}{Odds\ in\ Group\ 2} \qquad Odds = \frac{p}{1-p}$$

Tree-based Model

Pros:

1. Understand decision path
2. Identify key predictors: top - more important
3. Detect interaction effect: split on different features at different levels, linear model(feature engineering)
4. Threshold effect: keep split at same value forming key decision boundary
5. Classify group

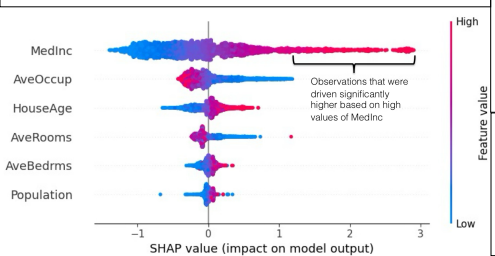
Cons:

1. Overfitting and noise: deep, big, complex (solution: pruning, max tree depth, min node size)
2. Multicollinearity is no considered
3. Lack of statistical significance
4. Can not make inference beyond observation

Feature Importance Score: measuring how much a feature reduces impurity across all splits
→ calculate decrease in impurity for that feature and normalize to get relative importance score
*Random forest: sum the decrease across all trees

SHAP force plot: the base value is 0.45, and the final predicted probability is 0.75. The biggest SHAP contributors are: Credit Score: +0.18, Debt-to-Income Ratio: +0.12.
The prediction started at the base value (0.45) and increased primarily due to Credit Score and Debt-to-Income Ratio. Together, these features pushed the prediction up by +0.30, leading to a final probability of 0.75. This shows how specific features influenced this individual prediction.

SHAP summary plot
Right (positive): feature increase prediction
Left (negative): feature decrease prediction
Widely spread: high variability in impact
Same side blue(red) dots: low(high) feature have consistent effect
Mixed color: complex/nonlinear relationship



$$Y = \beta_0 + \beta_1 X + \epsilon$$

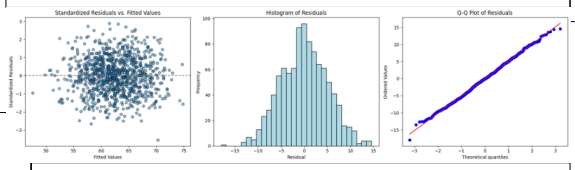
dependent (outcome) · independent (predictor) · intercept · slope (coefficients)

- **Coefficients**: magnitude - strength of the effect / sign - direction of the relationship
- **Intercept**: the expected value of Y when all predictors are 0
- **P-value**: whether the coefficient is statistically significant

H_0: No relationship between X and Y (coefficient = 0)
H_1: No relationship between X and Y (coefficient ≠ 0)

Linear Regression Model Assumptions

- **Linearity** of response-predictor relationship
- **Independence** of residuals
- **Normal Distribution** of residuals → QQ-plot
- **Equal variance** of residuals (Homoscedasticity)



Global Interpretability : how model make decisions across entire data
SHAP (Shapley Additive Explanations): How much each feature contributes to model across all samples
PDPs (Partial Dependence Plots): Show *average effect* of a feature on predictions. How predicted values change as one feature varies, averaging over all other
ICE (Individual Conditional Expectation Plots): show feature effects at the *individual* level

Local Interpretability: explain individual prediction
SHAP: computes the marginal contribution of each feature across all permutations, can be applied to any predictive model, providing consistent, additive explanations for both linear and non-linear models
LIME (Local Interpretable model-agnostic explanations): fits a simple surrogate model (e.g., linear) around a single prediction to approximate the model's local behavior near that point

Interpretability Type	Scope	Answers the Question	SHAP Tools
Global	Entire dataset	"Which features matter most overall?"	Summary plot, bar plot
Local	Single prediction	"Why was this prediction made?"	Force plot, waterfall plot

Aspect	LIME	SHAP
Type	Local linear approximation	Game-theoretic Shapley values
Model	Linear/Decision tree surrogate	Global-local consistent model
Focus	Single-instance explanation	Marginal contribution of each feature
Speed	Faster for large models	Slower but more reliable
Assumptions	Independence of features	No independence assumptions

Use Case	Preferred Method
Explaining a single prediction	LIME
Auditing model behavior globally	SHAP
Debugging a "high-stakes" model	SHAP
Fast local explanations	LIME
Simple, interpretable explanations	LIME

Factors effect coefficient

- **Scaling**: standardizing variables help compare
- **Interaction effects**: combined effect differs from individual effects
- **Multicollinearity**: misleading if predictors are highly correlated → correlation matrix / VIF

Variance Inflation Factor (VIF): measures how much the variance of a model coefficient is inflated due to multicollinearity
1 Low 5 Medium 10

$$VIF(\hat{\beta}_j) = \frac{1}{1 - R^2_{X_j|X_{-j}}}$$

where $R^2_{X_j|X_{-j}}$ is the R^2 of a regression model with X_j as the response variable and predictors $X_1, \ldots, X_{j-1}, X_{j+1}, \ldots, X_p$

Cause of counterintuitive coefficient

- Simpson's paradox: overall trends reverse subgroups, categorical
- Omitted variable bias (OVB): fail to include 1+ relevant variables, numeric

	Regression Tree	Classification Tree
Purity Metrics	SSE/MSE	Classification error rate, Gini Index, Entropy
Values	Average value for prediction	Class probabilities, Most common class

SHAP score algorithm

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N|-|S|-1)!}{|N|!}\left(v(S \cup \{i\}) - v(S)\right)$$

N is the set of all features
S is the subset of N that does not include feature i
$v(S)$ is the prediction (or payoff) when considering the subset of features S
$v(S \cup \{i\})$is the prediction when the feature i is added to the subset S
$|S|$ is the number of features in subset S
$|N|$ is the total number of features
$\phi_i(v)$ is the Shapley value for feature i

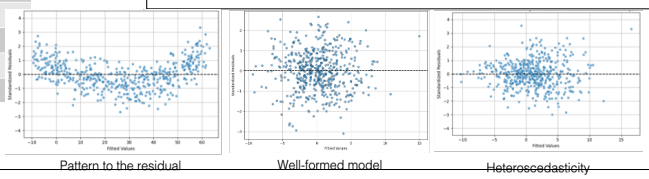
1. Generate every possible subset of predictor (N)
2. Calculate shapley score for target feature (S)

Example: X1, X2, X3, calculate score for X1
N = {X1,X2,X3}, {X1,X2}, {X2,X3}, {X1,X3}, {X1}, {X2}, {X3}, { }
S = {X2,X3}, {X2}, {X3}, { }

X1	X2	X3	Prediction
	9	147	244
94	9	147	664
	9		104
94	9		466
		147	479
94		147	157
			282
94			0

S	$\frac{\lvert S\rvert!\,(\lvert N\rvert - \lvert S\rvert - 1)!}{\lvert N\rvert!}$	$v(S \cup \{i\}) - v(S)$
{X2, X3}	$\frac{2!\,(3-2-1)!}{3!} = \frac{2}{6}$	$664 - 244 = 420$
{X2}	$\frac{1!\,(3-1-1)!}{3!} = \frac{1}{6}$	$466 - 104 = 362$
{X3}	$\frac{1!\,(3-1-1)!}{3!} = \frac{1}{6}$	$479 - 157 = 322$
{ }	$\frac{0!\,(3-0-1)!}{3!} = \frac{2}{6}$	282

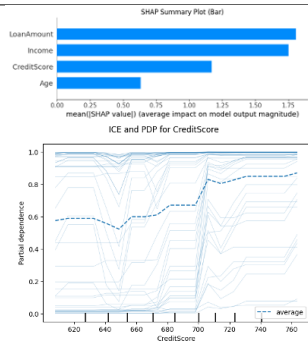
$$\phi_{X1}(v) = \frac{2}{6}(420) + \frac{1}{6}(362) + \frac{1}{6}(322) + \frac{2}{6}(282) = 348$$



PDP (age predict health risk): The plot shows a flat line up to Age 50, then a sharp upward trend. The model predicts little to no effect of age on health risk below 50, but increased risk after 50, suggesting a threshold or non-linear effect that the model captures.

ICE plot (for income): most curves slope upward, but 10% of individuals show downward-sloping lines.
For most individuals, higher income increases predicted approval probability (positive effect), but for a small group, the effect is negative, possibly due to interactions with other features. Heterogeneous effects and a possible interaction or non-monotonic relationship.

1. ICE lines differ from avg PDP line: heterogeneous effects
2. Around 700, the ICE and PDP both show a sharp increase: threshold
3. highest mean SHAP value: contributes the most to prediction decisions
4. upward trend of avg PDP line: positive relationship



Clustering Quality

Hopkins statistic: To find whether the data has meaningful clusters, measures the probability that the data is generated by a uniform random distribution

- Random sample n points and find distance from each point to its nearest neighbor (x_i)
- Random sample n points (not datapoints) and find distance from each point to its nearest neighbor (y_i)

0<no clusters<0.5(uniformly distributed)<cluster<1

Cohesion: How close the point is to other points in the same cluster (min intra-cluster)

Separation: How far the point is from points in the nearest different cluster (max inter-cluster)

$$H = \frac{\sum_{i=1}^{n} y_i}{\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i}$$

Point	X	Y	Dist_to_A1	Dist_to_B1	Dist_to_C1	Assigned Cluster
P1	2	10	0	9.2	9	A1
P2	4	8	4	5.2	9	A1
P3	5	5	8	1.2	7	B1
P4	6	4	10	0.8	7	B1
P5	8	4	12	2.8	9	B1
P6	1	2	9	7.8	0	C1
P7	7	3	12	2.8	7	B1

Cluster	X	Y
A	2	10
B	6	4.8
C	1	2

Aspect	Silhouette Score	Elbow Method (inertia)
What it measures	How well-separated and cohesive the clusters are	How compact (tight) the clusters are
Based on	Distance between each point and: • its own cluster center (cohesion) • the nearest other cluster center (separation) $s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$	Sum of squared distances between each point and its own cluster center
Output range	-1 to +1 (higher is better) weak < 0.2 < acceptable < 0.5 < good	Always positive; lower inertia means tighter clusters
Goal	Maximize the average silhouette score	Finding a good k value where adding more clusters doesn't help much
Limitations	Can be misleading when clusters vary in shape	Always decreases with more clusters – needs interpretation

K-means (sensitive to outliers)

1. Select k cluster centers
2. Assign each object to closest cluster center
3. Update cluster centers (**Average of new cluster**)
4. Repeat 2 and 3 until convergence

K-medoids

3. For each cluster, attempt to swap a medoid with another point in cluster to see if it minimizes the total distance within the cluster

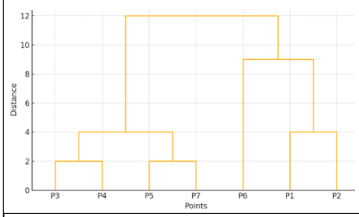
Hierarchical Clustering

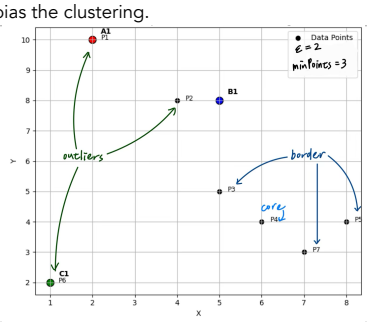
- Agglomerative: individual cluster and merge each step
- Divisive: all-inclusive cluster and split each step

Calculating distance between clusters (Linkage)

- Single: closest entities between clusters
- Complete: furthest entities between clusters
- Group average: average of all pairs
- Centroid: centroid between clusters
- Manhattan distance (given matrix)

	K-means (centroids)	K-mediods
Center	Average, not real point	Center, real data point
Outliers	Sensitive	Robust
Distance-based	Without scaling, variables with larger numeric ranges will dominate the distance calculation and bias the clustering.	





Model-based Clustering

Data is generated from a mixture of underlying probability distributions, typically Gaussian. Each cluster corresponds to 1 component in the mixture.

- **Expectation (E)**: computes the probability that each data point belongs to each cluster, based on current estimates of the parameters.
- **Maximization (M)**: updates the parameters of each distribution (mean, covariance, and mixing proportion) using the probabilities in the E-step
- **Bayesian Information Criterion (BIC)/Akaike Information Criterion (AIC)**: comparing model and choose lower values that indicating better fit with fewer parameters

Density-based Clustering (DBSCAN)

Clusters are defined as areas of high point density separated by regions of low density, rather than by distance to centroids or hierarchical structure.

- **Core points**: have at least *minPts* within ε
- **Border points**: have fewer than *minPts* within ε but within ε of core point
- **Noise points**: are neither core nor border (i.e., outliers)

* ε (epsilon): the radius defining a neighborhood around a point.
* minPts: the minimum number of points required within ε for a point to be considered a core point.

Pros: explicitly labels low-density points as noise, while k-means forces all points into clusters, making it sensitive to outliers. Not require to define number of clusters

Cons: A single ε value cannot simultaneously capture dense and sparse clusters, it may over-cluster sparse areas or merge dense clusters incorrectly.

Association Rule Mining

- **Support**: the proportion of transactions that contain a particular itemset.
- **Confidence**: the conditional probability that a transaction containing the antecedent also contains the consequent.
- **Lift**: how much the variables deviate from being independent, adjusted for the baseline frequency of the consequent and measures how much more likely the consequent is when the antecedent is present, relative to chance.
- A rule may have high confidence simply because the consequent is common, not because there's a real association.
- Rare item problem: Rare but important items (e.g., expensive products) may be filtered out by the minimum support threshold.

$$sup(X) = \frac{|\{T_j \in T; X \subseteq T_j\}|}{|T|} \quad conf(X \Rightarrow Y) = \frac{\sup(X \cup Y)}{\sup(X)} \quad lift(X \Rightarrow Y) = \frac{conf(X \Rightarrow Y)}{supp(Y)}$$

T1: {Milk, Bread, Butter}
T2: {Bread, Butter}
T3: {Milk, Bread}
T4: {Milk, Butter}
T5: {Bread, Butter}

- sup({Milk, Bread})=2/5
- conf({Milk} → {Bread})=2/3
- Lift = Confidence({Milk} → {Bread}) / Support({Bread})=10/12

- **Apriori algorithm**:

1. Generate all frequent itemsets by pruning infrequent ones based on the minimum support threshold.

***Support Monotonicity**: The support of an itemset cannot be greater than the support of any of its subsets → pruning of candidate itemsets, if a subset is infrequent, all its supersets can be ignored, improving efficiency

If $X_1 \subset X_2$, support(X_1) ≥ support(X_2)

2. From those frequent itemsets, generate association rules that meet the minimum confidence threshold.

***Confidence Monotonicity**: conf ({A, B}→{C}) ≥ conf ({A}→{B,C})

Assume the minimum support threshold is 0.4. Given the following itemsets and their supports: {A}=0.6 {B}=0.5 {C}=0.3 {A,B}=0.4 {A,C}=0.25 {B,C}=0.2

Which itemsets are frequent?›{A}, {B}, {A,B}

Which itemsets will be pruned by the Apriori algorithm?›{C}, {A,C}, {B,C}

{Green Tea Cup} → {Pink Tea Cup}

Support = 0.03: 3% of all transactions contain both items.

Confidence = 0.62: When the Green Tea Cup is purchased, there's a 62% chance the Pink one is also purchased.

Lift = 16.4: The co-occurrence of these items is 16.4 times more likely than if they were independent—a strong association.

	Cosine Similarity	Pearson Correlation	Jaccard Similarity
Measure	Angular similarity between raw rating vectors	Linear relationship between adjusted user bias	Measures overlap of binary interactions (e.g., clicks, views)
Best usage	Ratings with no normalization	Explicit ratings	Implicit feedback
Interpretation	1: Identical direction 0: No overlap	1: Perfect positive 0: No linear relationship -1: Opposite pattern	1: Perfect match 0: No overlap
Calculation	$\frac{\sum_{i \in I_{AB}} R_{A,i} R_{B,i}}{\sqrt{\sum_{i \in I_{AB}} R^2_{A,i}} \sqrt{\sum_{i \in I_{AB}} R^2_{B,i}}}$	$\frac{\sum_{i \in I_{AB}} (R_{A,i} - \bar{R}_A)(R_{B,i} - \bar{R}_B)}{\sqrt{\sum_{i \in I_{AB}} (R_{A,i} - \bar{R}_A)^2} \cdot \sqrt{\sum_{i \in I_{AB}} (R_{B,i} - \bar{R}_B)^2}}$	$\frac{\lvert I_A \cap I_B \rvert}{\lvert I_A \cup I_B \rvert}$

	Movie A	Movie B	Movie C	Movie D
User1	5	3	4	
User2	3	1	2	3
User3	4	3	4	

Calculate the Cosine Similarity between User1 and User3

User1: A=5, B=3, C=4

User3: A=4, B=3, C=4 $\frac{5\times4+3\times3+4\times4}{\sqrt{5^2+3^2+4^2}\cdot\sqrt{4^2+3^2+4^2}} = 0.995$

If the cosine similarity between User A and User B is 0.92, and between User A and User C is 0.60, and both have rated the same movie with ratings:

› User B: 4 $\frac{0.92\times4+0.6\times5}{0.92+0.6} = 4.39$

› User C: 5

Collaborative Filtering

Normalization before computing similarity: different rating scales, removing individual user bias and make comparison meaningful

	Item X	Item Y	Item Z
U1	1	1	0
U2	1	0	1

Compute the Jaccard similarity between U1 and U2 = 1/3