Quantitative Social Science in the Age of Big Data and AI

Lab 8: Machine Learning to Prediction(II)

Zhaopeng Qu
Hopkins-Nanjing Center
June 05 2025

Review Classification in Machine Learning

What is Classification?

- Classification is a supervised learning task that predicts discrete categories or class labels for new instances.
- Types of Classification
    - Binary Classification: Two classes only (0/1, Yes/No, Positive/Negative)
    - Example: Email spam detection (spam/not spam)
    - Multiclass Classification: Multiple classes (>2), but each instance belongs to exactly ONE class
    - Example: Handwritten digit recognition (0,1,2,3,4,5,6,7,8,9)
    - Multilabel Classification: Multiple classes, and each instance can belong to MULTIPLE classes simultaneously
    - Example: Text tagging (a document can be both "politics" AND "economy")

Common Classification Algorithms

Linear Methods

- Logistic Regression: Uses logistic function, interpretable
- Lasso/Ridge: Regularized versions prevent overfitting

Tree-Based Methods

- Decision Trees: Easy to interpret, prone to overfitting
- Random Forest: Ensemble of trees, more robust

Distance-Based Methods

- K-Nearest Neighbors (KNN): Lazy learning, no training phase

Other Popular Methods

- Support Vector Machines (SVM): Maximum margin classifier
- Naive Bayes: Probabilistic, assumes feature independence

Key Evaluation Metrics

The confusion matrix

- It is used to display the correct and incorrect predictions for each class of the outcome.

Confusion Matrix: Election Prediction

	Actual Vote Choice	
Prediction	**Challenger**	**Incumbent**
Challenger	True Negative (TN)	False Negative (FN)
Incumbent	False Positive (FP)	True Positive (TP)

Key Evaluation Indicators

Accuracy

- Overall correctness: `(TP + TN) / (TP + TN + FP + FN)`

Precision & Recall

- Precision: `TP / (TP + FP)` - "How many predicted positives are actually positive?"
- Recall: `TP / (TP + FN)` - "How many actual positives did we capture?"

F1-Score

- Harmonic mean of precision and recall: `2 × (Precision × Recall) / (Precision + Recall)`

ROC-AUC

- Area Under the Receiver Operating Characteristic curve
- Measures performance across all classification thresholds

The workflow of machine learning

1. Define the Prediction Task
2. Explore the Data
3. Set Model and Tuning Parameters
4. Perform Cross-Validation
5. Evaluate the Models and Select the Best One or Ensemble Methods

Define the Prediction Task

Predict Credit Default Risk

Research Question

> Can we predict whether a loan applicant will default on their credit obligations?

This is a classic binary classification problem in financial risk assessment, where we aim to:

- Minimize financial losses by identifying high-risk borrowers
- Optimize lending decisions through data-driven approaches
- Balance risk and profitability in credit approval processes

German Credit Dataset Overview

Dataset Background

- German credit data from UCI Machine Learning Repository
- Historical loan applications with known outcomes
- Bank credit risk assessment for loan approval decisions

Dataset Characteristics

- Total Observations: 1,000 loan applications
- Features: 9 variables (including target variable `default`)
- Target Variable: Binary outcome (default/no default)
- Class Distribution: Balanced representation of both outcomes

Key Features

Numerical Variables

- `duration`: Loan duration in months (credit period length)
- `amount`: Credit amount in Deutsche Mark (loan size)
- `installment`: Installment rate as percentage of disposable income
- `age`: Age of the applicant in years

Categorical Variables

- `history`: Credit history status
  - `good`: Positive credit history (A30, A31)
  - `poor`: Some credit issues (A32, A33)
  - `terrible`: Serious credit problems (A34)
- `purpose`: Purpose of the loan
  - `newcar`, `usedcar`: Vehicle purchases
  - `goods_repair`: Goods and repairs
  - `edu`: Education, `biz`: Business, `na`: Other

- `foreign`: Nationality status (`german` / `foreign`)
- `rent`: Housing status (whether applicant rents their residence)

Load and Preprocess Data

First 6 rows of Credit Dataset

default	duration	amount	installment	age	history	purpose	foreign	rent
0	6	1169	4	67	terrible	goods_repair	foreign	FALSE
1	48	5951	2	22	poor	goods_repair	foreign	FALSE
0	12	2096	2	49	terrible	edu	foreign	FALSE
0	42	7882	2	45	poor	goods_repair	foreign	FALSE
1	24	4870	3	53	poor	newcar	foreign	FALSE
0	36	9055	2	35	poor	edu	foreign	FALSE

Data Quality Assessment

- Check missing values

```
#> Missing values per column:

#>     default    duration      amount installment         age     history
#>           0           0           0           0           0           0
#>     purpose     foreign        rent
#>           0           0           0
```

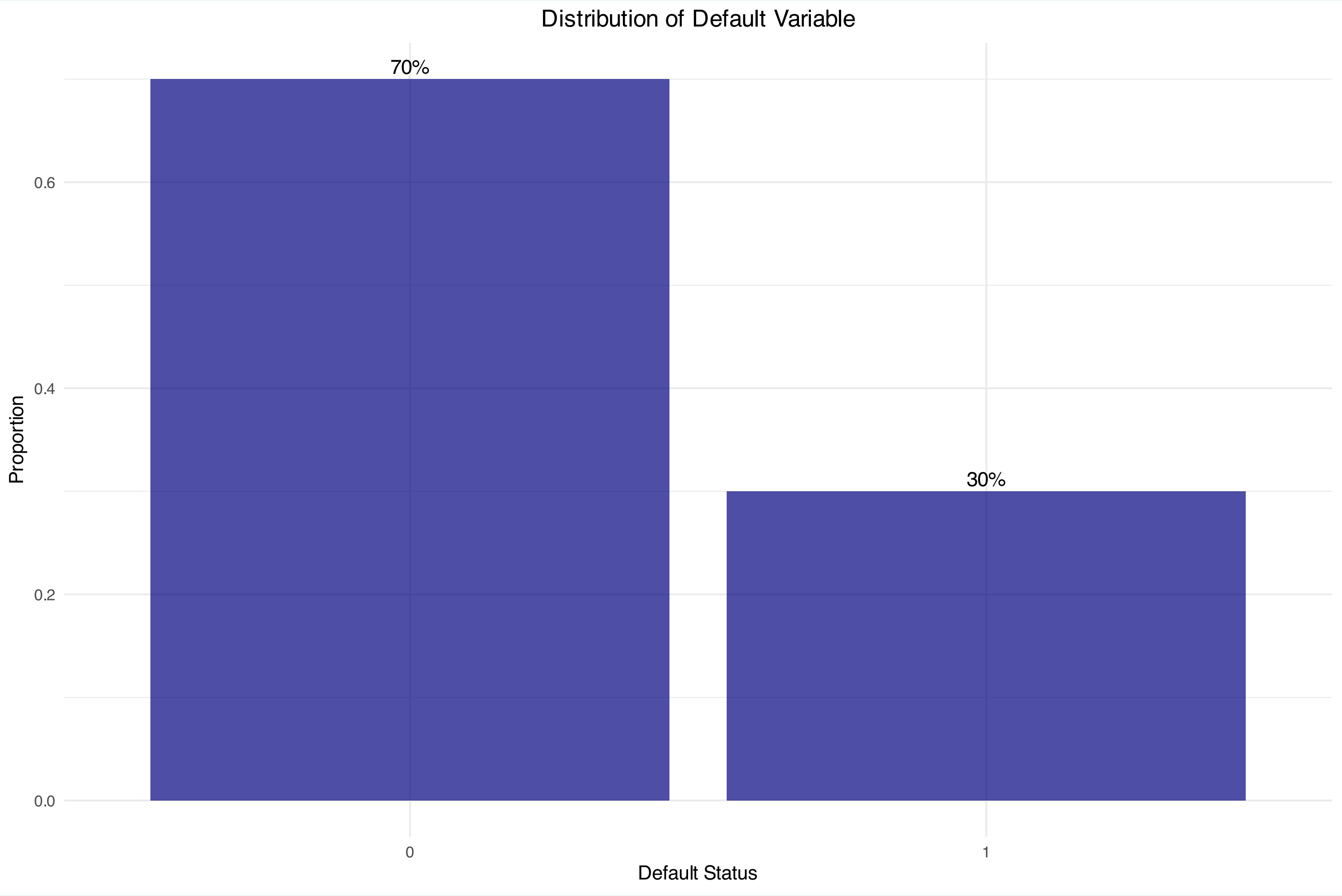
- Check data types

```
#>
#> Data types:

#>     default    duration      amount installment         age     history
#>    "factor"   "numeric"   "numeric"   "numeric"   "numeric"    "factor"
#>     purpose     foreign        rent
#>    "factor"    "factor"    "factor"
```

Exploratory Data Analysis

Target Variable Distribution



- Class Distribution: Balanced dataset with reasonable representation of both classes

Summary Statistics

Descriptive Statistics for Numeric Variables

Variable	Mean	SD	Max	Min
duration	20.903	12.059	72	4
amount	3271.258	2822.737	18424	250
installment	2.973	1.119	4	1
age	35.546	11.375	75	19

Categorical Variables Distribution

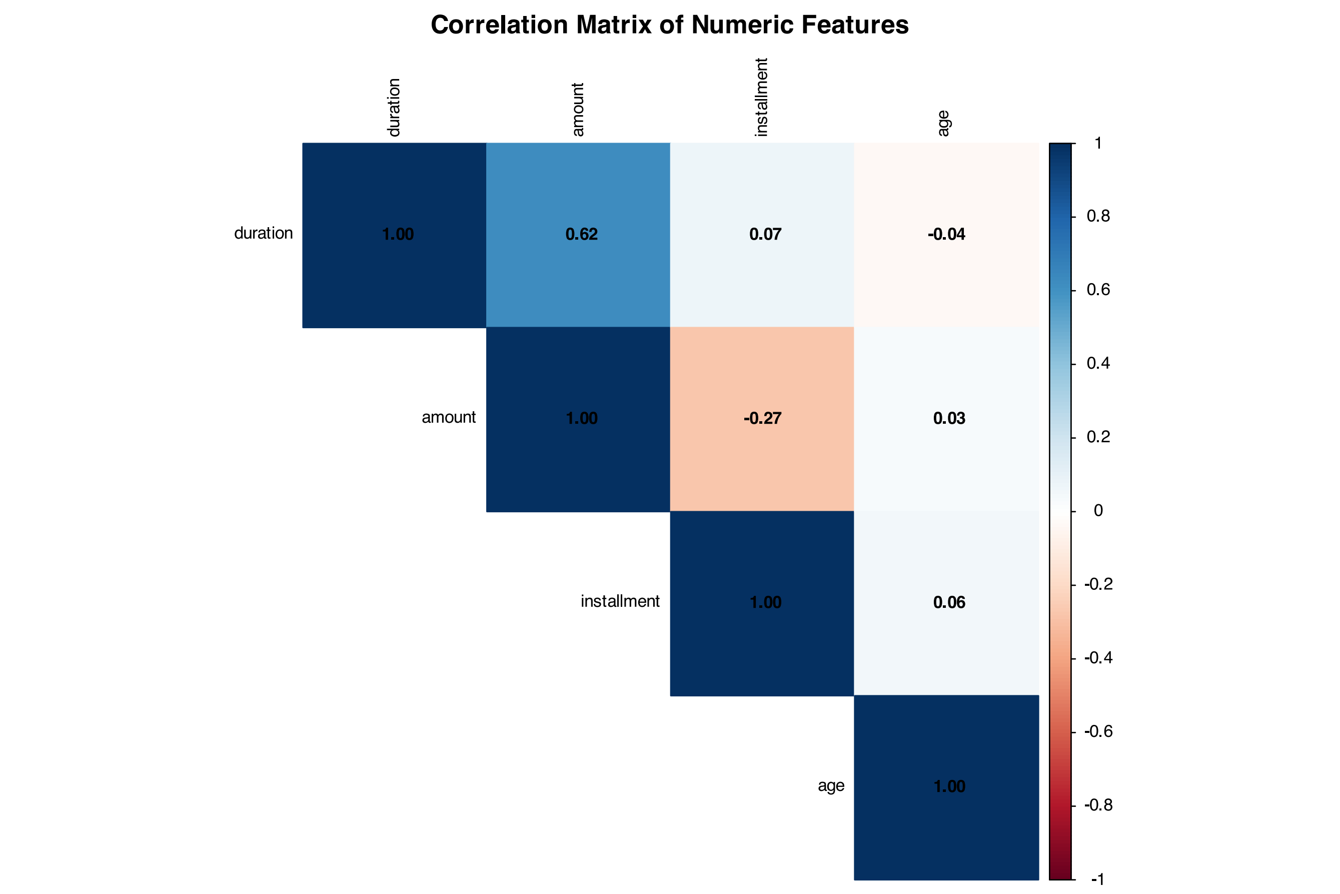
Categorical Variables (Part 1)

Variable	Category	Count	Proportion
default	0	700	0.700
default	1	300	0.300
history	good	89	0.089
history	poor	618	0.618
history	terrible	293	0.293

Categorical Variables (Part 2)

Variable	Category	Count	Proportion
purpose	newcar	234	0.234
purpose	usedcar	103	0.103
purpose	biz	109	0.109
purpose	goods_repair	495	0.495
purpose	edu	59	0.059
foreign	foreign	963	0.963
foreign	german	37	0.037
rent	FALSE	821	0.821
rent	TRUE	179	0.179

Correlation Analysis: Feature Correlations



- Purpose: Detect multicollinearity among predictors
- Key Insight: Strong correlations ($|r| > 0.7$) may indicate redundant features

Feature-Target Associations

Point-Biserial Correlation: A measure of association between a continuous variable and a binary variable (0/1). It's essentially a Pearson correlation when one variable is dichotomous.

Numeric Variables vs Default Risk

Variable	**Correlation**	**Interpretation**
duration	0.215	Moderate
amount	0.155	Moderate
installment	0.072	Weak
age	-0.091	Weak

- Positive correlation: Higher values associated with higher default risk
- Negative correlation: Higher values associated with lower default risk

Note: Point-Biserial Correlation Thresholds:

- $|pb| > 0.3$: Strong association with default risk
- $0.1 < |pb| < 0.3$: Moderate association with default risk
- $|pb| < 0.1$: Weak association with default risk

Categorical Variables vs Default

Cramér's V: A measure of association between two categorical variables, ranging from 0 (no association) to 1 (perfect association). It's based on the chi-square statistic but normalized to be comparable across different table sizes.

Cramér's V: Categorical Variables vs Default Risk

Variable	Cramers_V	Interpretation
history	0.248	Moderate
purpose	0.151	Moderate
foreign	0.076	Weak
rent	0.090	Weak

- Cramér's V: Measures association strength (0 = no association, 1 = perfect association)
- Interpretation: Higher values indicate stronger predictive potential

Note: Cramér's V Thresholds:

- $V > 0.3$: Strong association with default risk
- $0.1 < V < 0.3$: Moderate association with default risk
- $V < 0.1$: Weak association with default risk

Model Training and Tuning

Data Splitting and Scaling

Train-Test Split (80-20)

```
#> Training set size: 800

#> Test set size: 200

#> Cross-validation folds: 5

#> Stratified sampling: Yes
```

Data Scaling: Standardization

```
#> Feature scaling completed using tidymodels recipes

#> Training features mean (should be ~0): 0

#> Training features std (should be ~1): 1

#> Number of dummy variables created: 4
```

Model Training and Tuning: Specifications

Model Specifications

1. Logistic Regression: Basic linear classifier
2. Lasso Regression: L1 regularized logistic regression
3. Ridge Regression: L2 regularized logistic regression
4. Decision Tree: Rule-based tree classifier
5. Random Forest: Ensemble of decision trees
6. K-Nearest Neighbors: Distance-based classifier

Model Evaluation Metrics

- ROC-AUC: Area under the receiver operating characteristic curve
- Accuracy: $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$
- Precision: $Precision = \frac{TP}{TP+FP}$ (of predicted positives, how many were actually positive?)
- Recall: $Recall = \frac{TP}{TP+FN}$ (of actual positives, how many were predicted positive?)
- F1-Score: $F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$

- Cross-Validation: 5-fold stratified sampling for robust evaluation
- Primary Metric: ROC-AUC for comprehensive classification performance

Linear Models: Logistic Regression

Basic Estimation Approach: Logistic regression uses the logistic function to model the probability of binary outcomes.

Logistic Function:

$$P(Y=1|X) = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots+\beta_p X_p)}}$$

Log-odds (Logit):

$$\text{logit}(p) = \ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$$

Linear Models: Lasso Regression

Basic Estimation Approach: Lasso regression adds L1 penalty to logistic regression, performing automatic feature selection.

Logistic Function with L1 Penalty:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p)}}$$

Optimization Objective:

$$\min_{\beta} \left[-\ell(\beta) + \lambda \sum_{j=1}^{p} |\beta_j| \right]$$

where $\ell(\beta)$ is the log-likelihood and λ is the penalty parameter.

Linear Models: Ridge Regression

Basic Estimation Approach: Ridge regression adds L2 penalty to logistic regression, shrinking coefficients towards zero.

Logistic Function with L2 Penalty:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p)}}$$

Optimization Objective:

$$\min_{\beta} \left[-\ell(\beta) + \lambda \sum_{j=1}^{p} \beta_j^2 \right]$$

where $\ell(\beta)$ is the log-likelihood and λ is the penalty parameter.

Tree-Based Models: Decision Tree

Basic Estimation Approach: Decision trees create recursive binary splits based on feature values to maximize information gain or minimize impurity.

Gini Impurity (for node splitting):

$$Gini = 1 - \sum_{i=1}^{c} p_i^2$$

where p_i is the proportion of class i in the node: Minimization Objective: For a potential split on feature A with threshold t, find the split that minimizes weighted impurity:

$$\text{Best Split} = \arg\min_{A,t} \left[\frac{|S_L|}{|S|} \cdot Gini(S_L) + \frac{|S_R|}{|S|} \cdot Gini(S_R) \right]$$

where:

- S = parent node (set of instances)
- S_L = left child node (instances with $A \leq t$)
- S_R = right child node (instances with $A > t$)

Tree-Based Models: Random Forest

Basic Estimation Approach: Random Forest combines multiple decision trees using bootstrap aggregating (bagging) and random feature selection.

Final Prediction (Majority Vote):

$$\hat{y} = \mathrm{mode}\{T_1(x), T_2(x), \ldots, T_B(x)\}$$

For Probability:

$$P(Y = 1|X) = \frac{1}{B}\sum_{b=1}^{B} P_b(Y = 1|X)$$

where B is the number of trees and T_b is the b-th tree.

Distance-Based Models: K-Nearest Neighbors

Basic Estimation Approach: KNN classifies a new observation based on the majority class of its k nearest neighbors in the feature space.

Distance Metric (Euclidean):

$$d(x_i, x_j) = \sqrt{\sum_{l=1}^{p} (x_{il} - x_{jl})^2}$$

Prediction (Majority Vote):

$$\hat{y} = \text{mode}\{y_{(1)}, y_{(2)}, \ldots, y_{(k)}\}$$

Probability Estimation:

$$P(Y = c|X) = \frac{1}{k} \sum_{i \in N_k(x)} I(y_i = c)$$

Cross-Validation Training and Tuning

Tuning Grids

```
#> Tuning grids created:
#> ✓ Lasso: 20 penalty values
#> ✓ Ridge: 20 penalty values
#> ✓ Decision Tree: 5³ = 125 combinations
#> ✓ Random Forest: 5² = 25 combinations
#> ✓ KNN: 5³ = 125 combinations
```

Cross-Validation Training and Tuning

```
#> Starting model training with 5-fold cross-validation...

#> 1/6 Training Logistic Regression...

#> 2/6 Training Lasso Regression...

#> 3/6 Training Ridge Regression...

#> 4/6 Training Decision Tree...

#> 5/6 Training Random Forest...

#> 6/6 Training KNN...

#>
#> ✓ All models trained successfully!
```

```
#> Best hyperparameters selected:

#> ✓ Lasso penalty: 0.000695

#> ✓ Ridge penalty: 0

#> ✓ Tree complexity: 0

#> ✓ RF mtry: 2

#> ✓ KNN neighbors: 20

#>
#> Fitting final models on test set...

#> ✓ All final models fitted on test set!
```

Cross-Validation Results

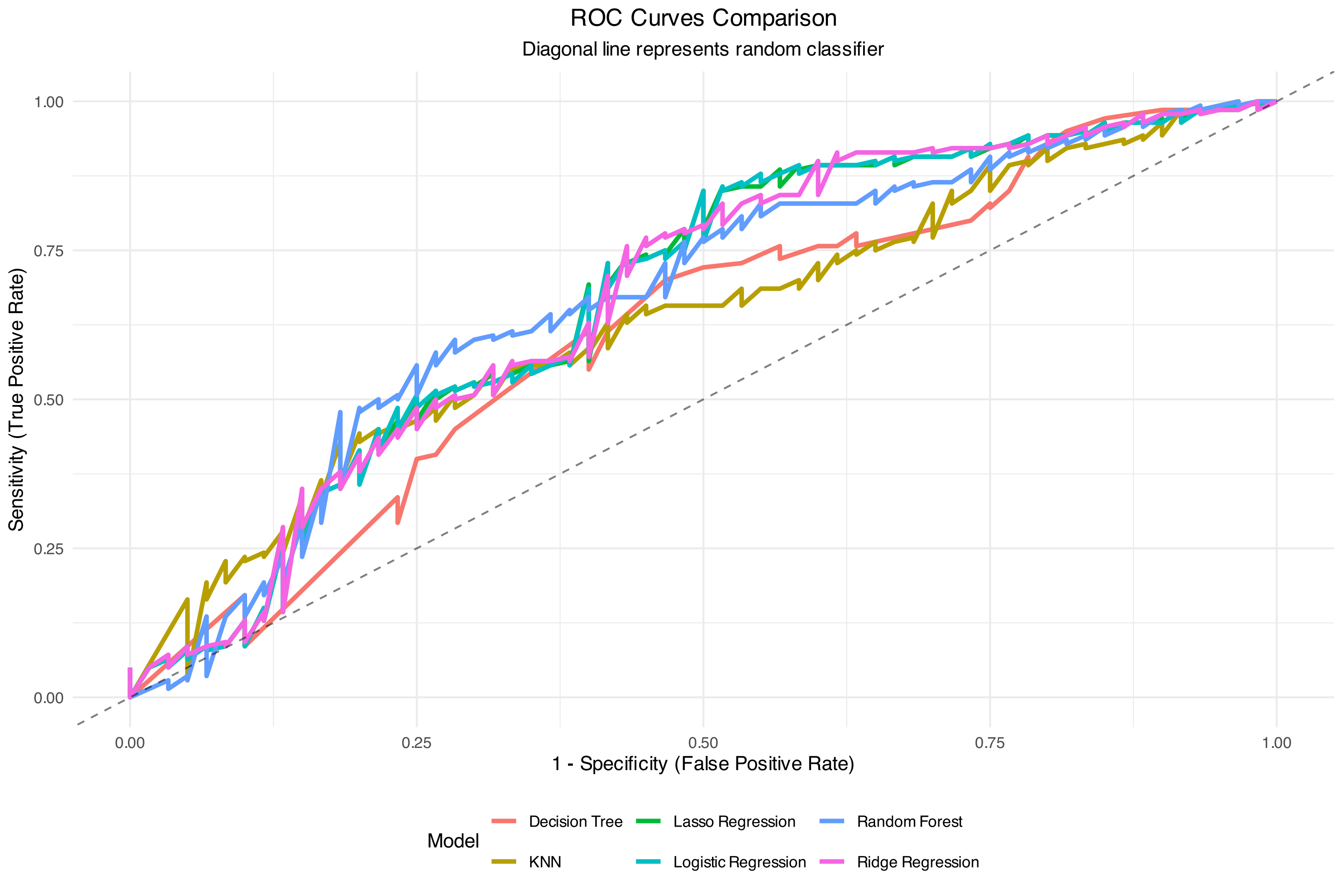
Cross-Validation Performance (ROC-AUC)

model	.metric	mean	std_err
Logistic Regression	roc_auc	0.7327	0.0218
Lasso Regression	roc_auc	0.7326	0.0219
Ridge Regression	roc_auc	0.7314	0.0212
Random Forest	roc_auc	0.7059	0.0167
KNN	roc_auc	0.6867	0.0217
Decision Tree	roc_auc	0.6640	0.0204

- This is the cross-validation results for the models, which is used to select the best hyperparameters for each model.

Final Model Training and Testing

ROC Curves Comparison



- ROC curves are used to compare the performance of the models.

Test Set Performance

Test Set Performance - All Metrics

model	accuracy	precision	recall	f_meas	roc_auc
Logistic Regression	0.720	0.7442	0.9143	0.8205	0.6729
Lasso Regression	0.725	0.7457	0.9214	0.8243	0.6726
Ridge Regression	0.720	0.7414	0.9214	0.8217	0.6711
Random Forest	0.715	0.7196	0.9714	0.8267	0.6689
KNN	0.695	0.7310	0.8929	0.8039	0.6296
Decision Tree	0.640	0.7361	0.7571	0.7465	0.6129

- Logistic Regression is the best model based on the ROC-AUC metric.

Summary and Conclusions

Key Findings and Insights

Final Model Rankings (by ROC-AUC)

rank	model	roc_auc	accuracy	precision	recall	f_meas
1	Logistic Regression	0.6729	0.720	0.7442	0.9143	0.8205
2	Lasso Regression	0.6726	0.725	0.7457	0.9214	0.8243
3	Ridge Regression	0.6711	0.720	0.7414	0.9214	0.8217
4	Random Forest	0.6689	0.715	0.7196	0.9714	0.8267
5	KNN	0.6296	0.695	0.7310	0.8929	0.8039
6	Decision Tree	0.6129	0.640	0.7361	0.7571	0.7465

Model Performance Patterns

🏆 Best Performing Model

Logistic Regression achieved the highest ROC-AUC of 0.6729 Linear Models Excellence:

- Basic logistic regression performs surprisingly well, outperforming regularized versions
- L1 and L2 penalties may not be necessary for this dataset size and complexity
- Simple models can be highly effective with well-prepared data

Tree-based Methods:

- Random Forest shows moderate performance, ranking in the middle tier
- Ensemble benefits not sufficient to outperform simple linear models on this dataset
- Single decision trees show significant overfitting issues and poor generalization

Distance-based Methods: KNN performance depends heavily on optimal hyperparameter selection

Practical Implications

🎯 Model Selection Recommendations

For Production**: Deploy** Logistic Regression

- Highest predictive accuracy on test set
- Simple and interpretable
- Fast training and prediction
- Good generalization without overfitting

For Interpretability: Use Logistic Regression

- Clear coefficient interpretation
- Business-friendly explanations
- Regulatory compliance

For Feature Selection**: Consider** Lasso Regression

- Automatic variable selection capability
- Sparse model creation
- Important features identification
- Second-best performance

For Robustness: Consider Random Forest

- Handles missing values well
- Less sensitive to outliers
- Provides feature importance

Next Steps and Future Work

🚀 Model Enhancement Opportunities

Feature Engineering:

- Try interaction terms and Polynomial features
- Try Domain-specific transformations

Advanced Techniques:

- Try Ensemble methods (stacking, voting)
- Try Deep learning approaches
- Try Gradient boosting methods