Report - Week 2

Created @July 12, 2024 4:02 PM

Report: Week 2 - Feature Selection and Engineering

Dataset: Loan Prediction Dataset

Theory

1. Importance of Feature Selection

Feature selection is crucial in building robust and efficient machine learning models. It helps in:

- **Reducing Over-fitting**: By removing irrelevant or redundant features, the model becomes less complex and more generalizable.
- **Improving Accuracy**: Better features lead to better model performance.
- **Reducing Training Time**: Fewer features mean less computational effort, resulting in faster training.

2. Techniques for Feature Selection

- **Filter Methods**: Evaluate the relevance of features by looking at the intrinsic properties of data. Examples include correlation coefficients, chi-square test, and mutual information.
- **Wrapper Methods**: Evaluate the features by training and testing the model on different combinations of features. Examples include forward selection, backward elimination, and recursive feature elimination (RFE).
- **Embedded Methods**: Perform feature selection during the process of model training. Examples include Lasso regression and tree-based methods.

3. Advanced Feature Engineering

- **Polynomial Features**: Creating interaction features that capture relationships between original features.
- **Domain-Specific Features**: Crafting features based on domain knowledge to improve model performance.

Practical Implementation

Data Preparation

First, we loaded the dataset and performed necessary preprocessing steps, such as encoding categorical variables and dropping irrelevant columns. We also created new features to capture more information and improve the model's predictive power. These new features included Total Assets Value, Debt-to-Income (DTI) ratio, Income x Loan Term, Loan to Asset Ratio, and Total Asset Ratio x Income.

Feature Scaling

To ensure all features contributed equally to the model, we scaled the numerical columns using standardization.

Train-Test Split

We split the dataset into training and testing sets to evaluate the model's performance on unseen data.

Initial Model Performance (Without Feature Selection)

Before applying any feature selection methods, we trained several models and evaluated their performance:

- **Logistic Regression**: 62.8% accuracy
- **Decision Trees**: 96% accuracy
- **K-means**: 57.8% accuracy
- **Random Forest**: 98% accuracy

Filter Methods:

Mutual Information: We calculated the mutual information between each feature and the target variable to determine their relevance. The most informative features were selected for further modeling.

Post information/Mutual info method:

```
Random Forest:
              precision    recall  f1-score   support

           0       0.99      1.00      1.00       523
           1       0.99      0.99      0.99       331

    accuracy                           0.99       854
   macro avg       0.99      0.99      0.99       854
weighted avg       0.99      0.99      0.99       854

Accuracy: 0.9941451990632318
Confusion Matrix:
[[521   2]
 [  3 328]]
Specificity: 0.9962
```

```
Decision Tree:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99       523
           1       0.99      0.98      0.99       331

    accuracy                           0.99       854
   macro avg       0.99      0.99      0.99       854
weighted avg       0.99      0.99      0.99       854

Accuracy: 0.990632318501171
```

Logistic Regression Post scaling information/Mutual info method: 92%

Correlation Method: By examining the correlation matrix, we identified and removed highly correlated features to reduce multi-collinearity.

Post Correlation Method:

```
Random Forest:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       523
           1       1.00      1.00      1.00       331

    accuracy                           1.00       854
   macro avg       1.00      1.00      1.00       854
weighted avg       1.00      1.00      1.00       854

Accuracy: 0.9988290398126464
Confusion Matrix:
[[523   0]
 [  1 330]]
Specificity: 1.0000
```

```
Decision Tree:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       523
           1       1.00      1.00      1.00       331

    accuracy                           1.00       854
   macro avg       1.00      1.00      1.00       854
weighted avg       1.00      1.00      1.00       854

Accuracy: 1.0
```

Logistic Regression Post scaling and Post Correlation Method: 74%

Variance Threshold: We removed features with low variance, as they provided little information to the model.

Post Var Threshold:

```
Random Forest:
              precision    recall  f1-score   support

           0       0.98      0.99      0.99       523
           1       0.99      0.98      0.98       331

    accuracy                           0.99       854
   macro avg       0.99      0.99      0.99       854
weighted avg       0.99      0.99      0.99       854

Accuracy: 0.9871194379391101
Confusion Matrix:
[[520   3]
 [  8 323]]
Specificity: 0.9943
```

```
Decision Tree:
              precision    recall  f1-score   support

           0       0.97      0.98      0.98       523
           1       0.97      0.96      0.96       331

    accuracy                           0.97       854
   macro avg       0.97      0.97      0.97       854
weighted avg       0.97      0.97      0.97       854

Accuracy: 0.9730679156908665
```

Logistic Regression Post Var Threshold: 74%

Wrapper Methods:

Forward Selection: We used forward selection to iteratively add features that improved the model's performance, based on accuracy.

Post Forward Selection:

```
Random Forest:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       523
           1       1.00      1.00      1.00       331

    accuracy                           1.00       854
   macro avg       1.00      1.00      1.00       854
weighted avg       1.00      1.00      1.00       854

Accuracy: 0.9988290398126464
Confusion Matrix:
[[523   0]
 [  1 330]]
Specificity: 1.0000
```

```
 Decision Tree:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       523
           1       1.00      1.00      1.00       331

    accuracy                           1.00       854
   macro avg       1.00      1.00      1.00       854
 weighted avg      1.00      1.00      1.00       854

 Accuracy: 1.0
```

Logistic Regression Post Forward Selection: 92%

Backward Elimination: Starting with all features, we iteratively removed the least significant feature based on p-values until only significant features remained. This improved Logistic Regression accuracy from 89.9% to 92.7%.

Recursive Feature Elimination (RFE): We recursively eliminated the least important features based on a specified model, ensuring that only the most impactful features were retained.

Post RFE:

```
Random Forest:
              precision    recall  f1-score   support

           0       0.99      1.00      1.00       523
           1       1.00      0.99      1.00       331

    accuracy                           1.00       854
   macro avg       1.00      1.00      1.00       854
weighted avg       1.00      1.00      1.00       854

Accuracy: 0.9964871194379391
Confusion Matrix:
[[523   0]
 [  3 328]]
Specificity: 1.0000
```

```
Decision Tree:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       523
           1       1.00      0.99      1.00       331

    accuracy                           1.00       854
   macro avg       1.00      1.00      1.00       854
weighted avg       1.00      1.00      1.00       854

Accuracy: 0.9976580796252927
```

Logistic Regression Post RFE: 74%

Embedded Methods:

Lasso Regression: We applied Lasso regression, which performed both feature selection and regularization, by penalizing less important features and shrinking their coefficients to zero. This helped improve Logistic Regression accuracy from 89.9% to 90.5%.

Post Lasso:

```
Random Forest:
              precision    recall  f1-score   support

           0       0.95      0.97      0.96       523
           1       0.95      0.92      0.94       331

    accuracy                           0.95       854
   macro avg       0.95      0.94      0.95       854
weighted avg       0.95      0.95      0.95       854

Accuracy: 0.9508196721311475
Confusion Matrix:
[[508  15]
 [ 27 304]]
Specificity: 0.9713
```

```
Decision Tree:
              precision    recall  f1-score   support

           0       0.95      0.95      0.95       523
           1       0.93      0.92      0.92       331

    accuracy                           0.94       854
   macro avg       0.94      0.94      0.94       854
weighted avg       0.94      0.94      0.94       854

Accuracy: 0.9402810304449649
```

Logistic Regression Post Lasso: 74.2%

Random Forest Feature Importance: We utilized the feature importance scores provided by a Random Forest model to select the most significant features.

Post Random Forest Feature Importance method:

```
Random Forest:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       523
           1       1.00      1.00      1.00       331

    accuracy                           1.00       854
   macro avg       1.00      1.00      1.00       854
weighted avg       1.00      1.00      1.00       854

Accuracy: 1.0
Confusion Matrix:
[[523   0]
 [  0 331]]
Specificity: 1.0000
```

```
Decision Tree:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       523
           1       1.00      1.00      1.00       331

    accuracy                           1.00       854
   macro avg       1.00      1.00      1.00       854
weighted avg       1.00      1.00      1.00       854

Accuracy: 1.0
```

Logistic Regression Post Random Forest Feature Importance: 92%

Regularization Techniques

1. **Lasso Regression (L1)**: Helped improve Logistic Regression accuracy from **89.9%** to **90.5%.**
2. **Ridge Regression**: Improved Logistic Regression accuracy to **90.3%** and SVM accuracy to **90.5%.**
3. **Elastic Net**: Enhanced Logistic Regression accuracy to **90.5%.**

Cross-Validation

To ensure the robustness of our models, we performed 5-fold cross-validation, assessing the models' performance across multiple splits of the dataset. This provided a more reliable estimate of the models' generalization capabilities.

Summary

- **Feature Engineering**: Created new features like Total Assets Value, DTI, Income x Loan Term, Loan to Asset Ratio, and Total asset ratio x Income.
- **Feature Selection**: Applied various feature selection methods, including filter methods (mutual information, correlation, variance threshold), wrapper methods (forward selection, backward elimination, RFE), and embedded methods (Lasso, Random Forest feature importance).
- **Regularization Techniques**: Utilized Lasso, Ridge, and Elastic Net to improve model performance.
- **Initial Model Performance**: Evaluated models without feature selection, resulting in lower accuracies.
- **Model Training**: Retrained Logistic Regression, Decision Trees, K-means, SVM, and Random Forest models, achieving significant accuracy improvements.
- **Evaluation**: Evaluated models using classification report, accuracy, and confusion matrix. Performed cross-validation to ensure model robustness.

This comprehensive approach to feature selection and engineering, followed by rigorous model training and evaluation, provides a solid foundation for building accurate and efficient machine learning models for loan prediction.
