[W7] Feature Selection Lab

Comparing Feature Selection Methods and Building Fair, Trustworthy ML Pipelines

Part 1 — Pima Diabetes: Feature Selection Comparison (5-Fold CV)

Objective

Compare three feature selection methods:

- VarianceThreshold
- SelectKBest(chi^2)
- RFE(LogisticRegression)

and determine which yields the highest test accuracy.

Implementation Steps

Step 1 — Load and Inspect Data

1. Load the Pima Diabetes dataset (`pima.csv`).
2. Identify the target variable (`Outcome`) and separate features (`X`) from the target (`y`).
3. Check for missing values and feature ranges.

Step 2 — Prepare Cross-Validation and Metrics

1. Set up 5-fold stratified cross-validation to maintain class balance.
2. Choose **accuracy** as the main evaluation metric.

Step 3 — Define Pipelines

1. Create three pipelines:
   - One with `VarianceThreshold` .
   - One with `SelectKBest(chi2)` (remember chi2 needs non-negative values).
   - One with `RFE(LogisticRegression)` .
2. Each pipeline should include appropriate scaling and a logistic regression classifier.

Step 4 — Evaluate Each Pipeline

1. Run cross-validation for each pipeline.
2. Record mean and standard deviation of accuracy for all three.

3. Present the results in a comparison table.

Step 5 — Inspect Selected Features (Optional)

1. Fit each selector on the full dataset to see which features were kept.
2. List the selected feature names for `SelectKBest` and `RFE`.

Step 6 — Reflection Write a short paragraph answering:

- Which method achieved the highest test accuracy?
- Why might their performances differ?
- Which method do you think is most suitable for this dataset, and why?

Part 2 — Titanic: Full Pipeline with RandomForest and Ethical Reflection

Objective

Build a complete ML workflow using the Titanic dataset to:

- Preprocess data safely
- Train a RandomForest model
- Visualize feature importance
- Reflect on ethical considerations

Implementation Steps

Step 1 — Load and Explore Data

1. Load the dataset (`titanic_train.csv`).
2. Identify the target column (`Survived`) and features.
3. Select a subset of numeric and categorical features for simplicity (e.g., `Age` , `Fare` , `Sex` , `Pclass` , `Embarked`).
4. Check for missing values.

Step 2 — Preprocessing with ColumnTransformer

1. Create numeric preprocessing steps (impute missing values, scale features).
2. Create categorical preprocessing steps (impute missing categories, one-hot encode).
3. Combine them into a single `ColumnTransformer` .

Step 3 — Build the Full Pipeline

1. Integrate preprocessing with a `RandomForestClassifier` .
2. Set basic hyperparameters (e.g., number of trees, random seed).

Step 4 — Train-Test Split and Model Training

1. Split data into training and testing sets.
2. Fit the pipeline on the training set.
3. Evaluate predictions on the test set.

Step 5 — Evaluate Model Performance

1. Generate a classification report showing precision, recall, F1-score, and accuracy.
2. Record the results for later comparison.

Step 6 — Analyze Feature Importances

1. Extract feature names after one-hot encoding.
2. Retrieve and visualize feature importances from the RandomForest model.
3. Identify which features the model relies on most.

Step 7 — Ethical Discussion Write a short reflection:

- Are features like **Sex** or **Pclass** among the most important predictors?
- What are the ethical implications of using them?
- How might you adjust your model to address fairness concerns?

Part 3 — Bringing It All Together: Trustworthy Titanic Workflow

Objective

Create a **complete, ethical, and reliable** modeling process—from problem framing to evaluation.

Implementation Steps

Step 1 — Problem Framing

1. Define the prediction goal (survival).
2. Discuss whether it's ethical to use features like `Sex` or `Pclass`.
3. Consider how historical bias might influence predictions.

Step 2 — Safe Preprocessing

1. Build a preprocessing pipeline to handle missing values and categorical encoding safely.
2. Ensure no data leakage by performing preprocessing within the pipeline.

Step 3 — Feature Selection

1. Use `SelectFromModel(RandomForest)` to automatically identify key predictors.
2. Note which features were retained and removed.

Step 4 — Prevent Overfitting

1. Apply `StratifiedKFold` for cross-validation.
2. Measure accuracy, precision, recall, and F1-score across folds.
3. Plot a **validation curve** to study how model performance changes with complexity (e.g., `max_depth`).

Step 5 — Comprehensive Evaluation

1. Summarize cross-validation results in a table.
2. Identify which metric best reflects true model performance given the class imbalance.

Step 6 — Reflection and Fairness Check

1. Plot the final feature importances.
2. Discuss whether sensitive variables (like gender or class) dominate the predictions.
3. Perform a small counterfactual check—imagine flipping "Sex" and see if predictions would change.
4. Conclude with a reflection on trustworthiness and fairness in AI models.

Student Deliverables

Each student must submit the **implementation code** (Python notebook or script) **and** a **short written report** for each part of the lab. Please refer to the **CTL assignment page** for the official submission deadline.

Part 1 — Feature Selection Comparison

- Submit your implementation code comparing the three methods.
- Include a table showing the mean and standard deviation of accuracies for:
  - VarianceThreshold
  - SelectKBest(chi^2)
  - RFE(LogisticRegression)
- Add a short paragraph (5–8 sentences) discussing which method performed best and why.

Part 2 — Titanic Pipeline and Ethical Reflection

- Submit your full pipeline implementation code using RandomForest and ColumnTransformer.
- Include:
  - Model performance metrics (accuracy, precision, recall, F1-score).
  - Feature importance table or plot.

    - A brief reflection (5–8 sentences) on the ethical use of features such as *Sex* and *Pclass*.

Part 3 — Trustworthy ML Workflow

- Submit your full implementation showing feature selection, cross-validation, and fairness check.
- Include:
    - A table of cross-validation results (accuracy, precision, recall, F1-score).
    - A validation curve plot analyzing model complexity.
    - A short discussion on fairness, bias, and model trustworthiness.