Feature selection and goals of feature selection:

Feature Selection:

In machine learning and statistics, feature selection, also known as variable selection, attribute selection or variable subset selection, is the process of selecting a subset of relevant features (variables, predictors) for use in model construction.

This step is crucial as it directly influences the model's performance, interpretability, and efficiency.

Goals

Reduce Overfitting: By eliminating irrelevant or redundant features, the risk of overfitting to noise in the training data is minimized.

Improve Model Accuracy: Focuses on the most predictive features, which can enhance the model's accuracy and generalization.

Decrease Complexity: Simplifies the model, making it more interpretable and computationally efficient.

Reduce Training Time: Fewer features mean less data to process, which decreases training and prediction time.

Avoid Multicollinearity: Reduces problems related to feature correlations that can destabilize model estimates.

Reduce Data Dimensionality: High-dimensional data can make modeling more complex and prone to the curse of dimensionality. Feature selection reduces the dimensionality of the dataset, improving model performance.

Effect of irrelevant features in the data set:

Irrelevant predictors can negatively affect a model's performance by introducing noise and confusing the learning process. This can lead to overfitting, where the model learns patterns that are not representative of the true underlying data, ultimately reducing its ability to generalize to new, unseen data. Additionally, these extraneous predictors increase the complexity of the model, making it slower to train and harder to interpret. By focusing on relevant features, the model becomes more accurate, efficient, and easier to understand.

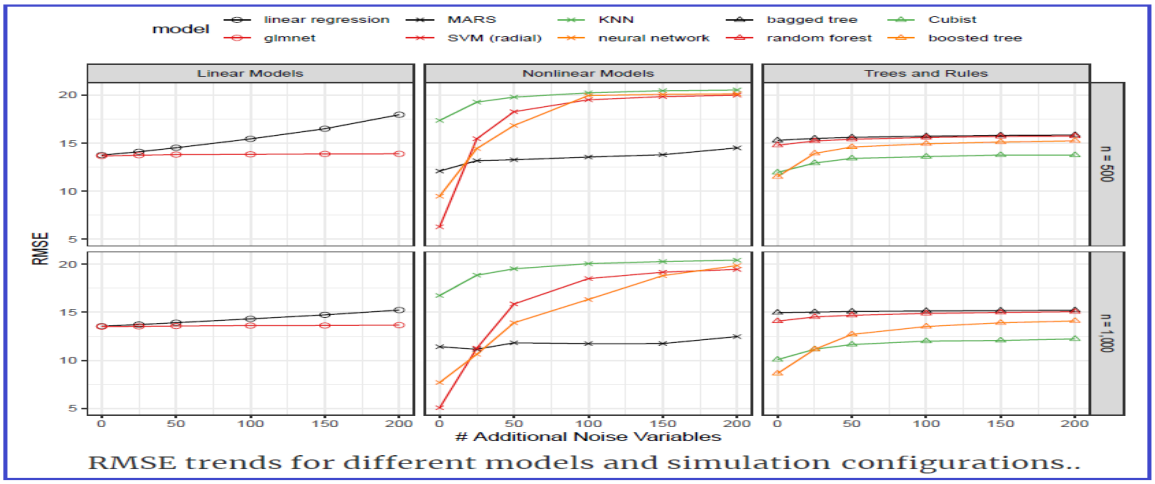
How much do extraneous predictors hurt a model?

- Predictably, that depends on type of model, nature of the predictors, ratio of the size of the training set to the number of predictors
- To investigate this, a simulation was used to emulate data with varying numbers of irrelevant predictors and to monitor performance.
- The simulation system consists of a nonlinear function of the 20 relevant predictors:

$$y = x_1 + \sin(x_2) + \log(|x_3|) + x_4^2 + x_5 x_6 + I(x_7 x_8 x_9 < 0) + I(x_{10} > 0) + x_{11} I(x_1 1 > 0) + \sqrt{(|x_{12}|)} + \cos(x_{13}) + 2x_{14} + |x_{15}| + I(x_{16} < -1) + x_{17} I(x_{17} < -1) - 2x_{18} - x_{19} x_{20} + \epsilon$$

To evaluate the effect of extra variables

- varying numbers of random standard normal predictors (with no connection to the outcome) were added.
- Between 10 and 200 extra columns were appended to the original feature set.
- The training set either n=500 or n=1000.
- The root mean squared error (RMSE) was used to measure quality of model using a large simulated test set.
- A number of models tuned and trained for each of simulates sets including linear, nonlinear, and tree/rule-based models



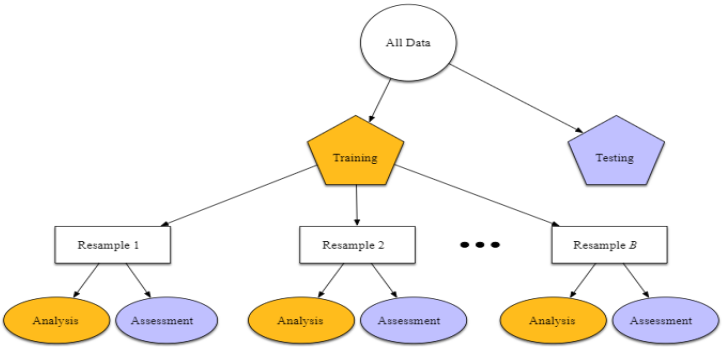
RMSE trends for different models and simulation configurations..

Results clearly show that

- There are a number of models that may require a reduction of predictors to avoid a decrease in performance.
- For models such as random forest or glmnet, it appears that feature selection may be useful to find a smaller subset of predictors without affecting the model's efficacy.

Methods for handling overfitting in feature selection:

Model may overfitting the available data during selection of model tuning parameters. Risk of finding tuning parameter values that over-learn the relationship between the predictors and the outcome in the training set. When models over-interpret patterns in the training set, the predictive performance suffers with new data. The solution to this problem is to evaluate tuning parameters on a data set that is not used to estimate model parameters. An analogous problem can occur when performing feature selection. For many data sets it is possible to find a subset of predictors that has good predictive performance on the training set but has poor performance when used on a test set or other new data set. The solution to this problem is similar to the solution to the problem of overfitting: feature selection needs to be part of the resampling process.



Unfortunately, practitioners often combine feature selection and resampling inappropriately. Most common mistake is to only conduct resampling inside of feature selection procedure.

```
Rank the predictors using the training set;
for subset sizes 5 to 1 do
    for each resample do
        Fit model with subset on the analysis set.;
        Predict the assessment set.;
    end
    Determine the best subset using resampled performance;
    Fit the best subset using the entire training set;
end
```

There are two key problems with this procedure:

- Since the feature selection is external to the resampling, resampling cannot effectively measure the impact (good or bad) of the selection process
- The same data are being used to measure performance and to guide the direction of the selection routine. This is analogous to fitting a model to the training set and then re-predicting the same set to measure performance

A better way of combining feature selection and resampling is to make feature selection a component of the modeling process. Feature selection should be incorporated the same way as preprocessing and other engineering tasks. What we mean is that an appropriate way to perform feature selection is to do this inside of the resampling process.

```
Split data into analysis and assessment sets;
for each resample do
    Rank the predictors using the analysis set;
    for subset sizes 5 to 1 do
        Fit model with subset on the analysis set;
        Predict the assessment set.;
    end
    Average the resampled performance for each model and subset size;
    Choose the model subset with the best performance;
    Fit the best subset using the entire training set;
end
```

Performing feature selection within the resampling process has two notable implications.

1. The first implication is that the process provides a more realistic estimate of predictive performance.
2. The second implication is an increase in computational burden

Intrinsic feature selection method:

Feature selection mythologies fall into three general classes:

1. Intrinsic (or implicit) methods

2. Filter methods
3. Wrapper methods.

Intrinsic methods have feature selection naturally incorporated with the modeling process. Whereas filter and wrapper methods work to marry feature selection approaches with modeling techniques. The most seamless and important of the three classes for reducing features are intrinsic methods. The most seamless and important of the three classes for reducing features are intrinsic methods.

Algorithms that perform automatic feature selection during training.

Some examples include:

Tree- and rule-based models

- These models search for the best predictor and split point such that the outcomes are more homogeneous within each new partition
- Therefore if a predictor is not used in any split, it is functionally independent of the prediction equation and has been excluded from the model.

Multivariate Adaptive Regression Spline (MARS) models

- These models create new features of the data that involve one or two predictors at a time.
- The predictors are then added to a linear model in sequence.
- Like trees, if a predictor is not involved in at least one MARS feature, it is excluded from the model.

Regularization models

- The regularization approach penalizes or shrinks predictor coefficients to improve the model fit.
- The lasso uses a type of penalty that shrinks coefficients to absolute zero.
- This forces predictors to be excluded from the final model.

Advantages-

- Relatively fast since selection process is embedded within model fitting process; no external feature selection tool is required.
- Direct connection between selecting features and objective function
- Objective function is statistic that model attempts to optimize.

Downside - It is model dependent.

If a model does not have intrinsic feature selection, then some sort of search procedure is required to identify feature subsets that improve predictive performance.

Chi-Square test for feature selection:

The Chi-Square test is commonly used for feature selection in machine learning, especially when dealing with categorical data. It helps in determining whether there is a significant relationship between the features (input variables) and the target (output variable). Here's how it works and how you can use it for feature selection:

Overview of Chi-Square Test

The Chi-Square test checks for the independence of two variables, i.e., it tests whether the distribution of categorical data for one variable is independent of another variable.

- Null Hypothesis (H_o): The two variables (feature and target) are independent.
- Alternative Hypothesis (H_1): The two variables are dependent.

If the test finds that the feature and target are not independent (low p-value), the feature is considered significant in predicting the target.

Chi-Square Test Formula

The Chi-Square statistic is calculated as:

$$\chi^2 = \sum \frac{(O - E)^2}{E}$$

Where:

- O = Observed frequency (actual value in the dataset).
- E = Expected frequency (what you would expect if the two variables were independent).

A high value of χ2 indicates that the observed and expected frequencies are different, implying a dependence between the feature and the target.

Steps to Use Chi-Square for Feature Selection

Here's how to use the Chi-Square test for feature selection:

a. Preprocessing Data

- The Chi-Square test works with categorical data. If the data is numerical, you'll need to discretize it (e.g., using binning techniques).

b. Create Contingency Table

- For each feature, create a contingency table that shows the frequency of occurrences for each category in the feature, split by the target categories.

c. Compute the Chi-Square Statistic

- For each feature, calculate the Chi-Square statistic by comparing the observed frequencies in the contingency table with the expected frequencies (assuming independence).

d. Determine Significance

- Compare the computed $\chi 2$ value with a critical value from the Chi-Square distribution table (based on degrees of freedom and significance level, typically 0.05).
- Alternatively, most libraries (like scikit-learn) directly provide the p-value, which tells you whether to reject the null hypothesis.

e. Select Features

- Features with a low p-value (typically < 0.05) are considered significant and should be selected for the model.
- Features with a high p-value are likely independent of the target and can be discarded.

Example

Consider you have a dataset with features like "Gender" (Male, Female) and "Income Bracket" (High, Medium, Low), and a target variable like "Purchased Product" (Yes, No).

- You would compute the Chi-Square statistic for each feature (e.g., Gender, Income Bracket) against the target (Purchased Product).
- If the p-value for "Gender" is low, you conclude that "Gender" is important in predicting whether a person will purchase the product.

Advantages of Using Chi-Square for Feature Selection

- Simplicity: It is easy to implement and interpret.
- Efficiency: It helps reduce dimensionality by selecting only the significant features.
- Works well with categorical data: It is particularly suited for datasets with categorical features.

Limitations

- Cannot handle continuous data directly: It requires the data to be categorical or discretized.
- Assumes independence: The test assumes independence between features, which might not always hold in real-world data.

Statistical test selection:

a. Identification of diabetic or non-diabetic person from gender.
b. Loan approval from the income of person.

a. Identification of Diabetic or Non-Diabetic Person from Gender

1. Test: Chi-Square Test.

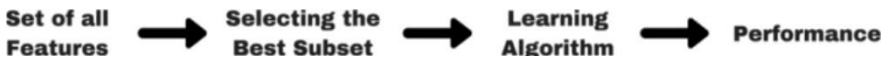
2. Reason: Gender is a categorical variable, and the Chi-Square test is appropriate for examining the relationship between categorical features and outcomes.

b. Loan Approval from the Income of Person

1. Test: t-Test or ANOVA.
2. Reason: Income is a continuous numerical feature, and t-tests (for two groups) or ANOVA (for multiple groups) assess the impact of numerical features on categorical outcomes.

Filter method for feature selection:

Filter methods are generally used as a preprocessing step.



Most basic approach - screen the predictors to see relationship with the outcome prior to including them in a model. To do this, a numeric scoring technique is required to quantify strength of relationship.

- Using the scores, the predictors are ranked and filtered with either a threshold or by taking the top predictors.
- Scoring the predictors can be done separately for each predictor, or can be done simultaneously across all predictors (depending on the technique that is used).
- If the predictors are screened separately, there are a large variety of scoring methods.
- Techniques used depends on the type of predictor and outcome.

When screening individual categorical predictors, there are several options depending on the type of outcome data.

- When the outcome is categorical,
- Relationship between the predictor and outcome forms a contingency table.
- When there are three or more levels for the predictor, the degree of association between predictor and outcome can be measured $\chi 2$ with statistics such as (chi-squared) tests or exact methods
- When there are exactly two classes for the predictor, the odds-ratio can be an effective choice.
- When the outcome is numeric, and the categorical predictor has two levels, then a basic t-test can be used to generate a statistic.
- ROC curves and precision-recall curves can also be created for each predictor and the area under the curves can be calculated.
- When the predictor has more than two levels, the traditional ANOVA - statistic can be calculated.

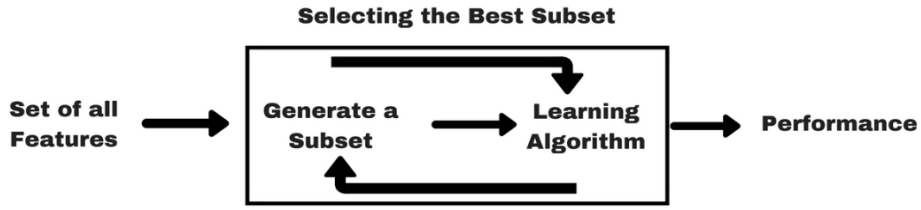
- When outcome is categorical, the same tests can be used in the case above where the predictor is categorical and the outcome is numeric.
- Roles are simply reversed in the t-test, and curve calculations.
- When there are a large number of tests or if the predictors have substantial multi-collinearity, the correlation adjusted t-scores are a good alternative to simple ANOVA statistics.
- When outcome is numeric, a simple pairwise correlation (or rank correlation) statistic can be calculated.
- Alternatively, a generalized additive model (GAM) can be used.

Feature\Response	Continuous	Categorical
Continuous	Pearson's Correlation	LDA
Categorical	Anova	Chi-Square

+

Wrapper method for feature selection:

Wrapper methods for feature selection are techniques that evaluate subsets of features based on the performance of a specific machine learning model. Unlike filter methods (like the Chi-Square test), which rank features independently of the model, wrapper methods rely on iterative model training and testing to select the best subset of features.



Types of Wrapper Methods

Forward Selection

- **Process**: Start with no features and progressively add features one by one, evaluating the model's performance at each step.
- **Steps**:
  1. Start with an empty set of features.
  2. Train a model with each individual feature, evaluate the performance, and choose the feature that gives the best performance.
  3. Add that feature to the feature set.

4. Repeat this process, adding one feature at a time, until adding more features does not improve performance.

2. **Backward Elimination**
   - **Process**: Start with all features and progressively remove the least important ones.
   - **Steps**:
     1. Train a model using all the features.
     2. Remove each feature one at a time, evaluate model performance for each removal.
     3. Drop the feature whose removal has the least negative effect on performance.
     4. Continue the process until the removal of more features worsens the performance significantly.

Recursive Feature Elimination (RFE) method:

Recursive Feature Elimination (RFE) is a popular wrapper method for feature selection that recursively removes the least important features from the dataset until a specified number of features are selected. It works by fitting a model and ranking the importance of features, then eliminating the least important feature(s) and repeating the process. The goal is to select a subset of features that gives the best model performance.

Steps in Recursive Feature Elimination (RFE)

1. **Train the Model**: Fit a model using all features in the dataset.
2. **Rank Features by Importance**: Use the trained model to rank the importance of each feature based on some criteria (e.g., coefficients in linear models or feature importance in decision trees).
3. **Eliminate the Least Important Feature**: Remove the least important feature (or features), retrain the model, and re-rank the remaining features.
4. **Repeat**: Continue removing the least important features one by one (or in batches) until the desired number of features is reached.
5. **Final Model**: The final subset of features is selected based on the highest model performance.

When to Use RFE

- **High Dimensional Data**: It's particularly useful when dealing with high-dimensional datasets, where not all features are relevant or contribute equally to model performance.
- **When Feature Interactions Matter**: RFE takes into account the interaction between features, making it more effective than simpler filter-based methods in some cases.

Advantages of RFE

1. Model-Specific: RFE selects features based on how they impact the performance of a specific model, making the selected features more relevant.
2. Handles Feature Interactions: RFE evaluates subsets of features iteratively, allowing it to account for complex interactions between features.
3. Adaptable to Many Models: RFE can be used with any model that provides feature importance or coefficient values, such as linear models, decision trees, or support vector machines.

Limitations of RFE

1. Computational Cost: RFE can be slow for large datasets or models that are expensive to train, as it requires fitting the model multiple times.
2. Risk of Overfitting: If not combined with cross-validation, there's a risk of overfitting the feature selection process to the training data.
3. Model Dependence: The selected features may work well for the model used in RFE but might not generalize as effectively to other models.

Applications of RFE

- Bioinformatics: Selecting a subset of relevant genes in high-dimensional genetic datasets.
- Text Classification: Choosing the most relevant words or n-grams for natural language processing tasks.
- Image Processing: Identifying the most important pixels or regions in image datasets.

RFE is a powerful technique that leverages the specific behavior of a machine learning model to iteratively select the best subset of features. When used correctly, it can improve model performance by reducing dimensionality and eliminating irrelevant or redundant features.

Stepwise Selection method for feature selection and its drawbacks:

Stepwise Selection is a feature selection technique designed for linear regression models. It incrementally adds or removes features from the model based on statistical criteria, usually p-values.

The process begins by fitting individual linear regression models for each feature. Features are ranked by their ability to explain variation in the target variable, and the one with the lowest p-value—typically below 0.15—is added to the model. If no feature meets this threshold, the process stops.

In subsequent steps, models are built by combining the already selected features with each remaining feature. The next feature to be added is the one that improves the model most, again based on a p-value threshold. Features may also be removed if their p-value exceeds a certain threshold, commonly

0.15. Although this method can identify a subset of significant features, it has notable limitations, such as a tendency to overfit, lack of stability, and sensitivity to small data variations. These drawbacks have led to widespread critique in the statistical literature.

How Stepwise Selection Works

1. **Define Entry and Exit Criteria**:
   - A common criterion for adding a feature is a p-value below a specific threshold (e.g., 0.05 or 0.15).
   - Features can be removed if their p-value exceeds a threshold or based on other criteria such as AIC (Akaike Information Criterion), BIC (Bayesian Information Criterion), or adjusted R^2.
2. **Iterative Process**:
   - Start with no features (in forward selection) and add the most significant feature, then re-fit the model.
   - These steps are repeated until the model converges, meaning that adding or removing any more features does not improve the model significantly.

Stepwise selection has two primary faults

1. Over selection of features
2. Model overfitting

Stepwise selection is a useful method for feature selection when building models, but it should be used with caution due to the risk of overfitting and instability. It's best used when you want to balance simplicity and model accuracy, especially in regression tasks.

****