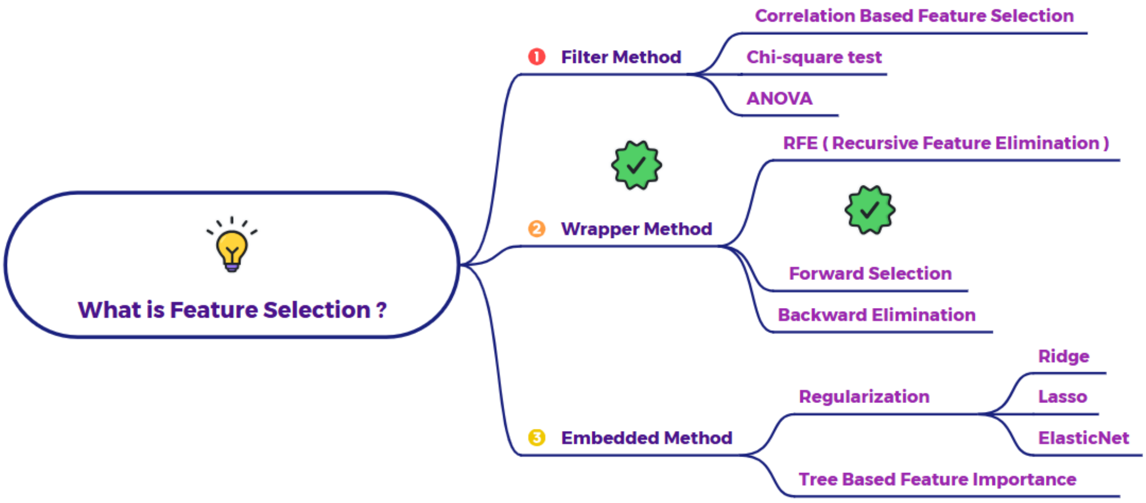
Explain Forward selection-based feature selection



Wrapper Method - Forward Selection for feature selection

Similar to RFE, forward selection uses a machine learning model to evaluate feature subsets, but it takes a different approach – it starts with an empty set of features and iteratively adds the most beneficial feature until a stopping criterion is met. [1]

Core Idea:

Forward selection begins with no features in the model. It then iteratively adds one feature at a time, selecting the feature that leads to the greatest improvement in the model's performance when added to the current set of features. This process continues until adding more features no longer significantly improves performance or a predefined number of features is reached.

How it Works:

1. **Start with an Empty Set:** Begin with an empty set of selected features.
2. **Consider Each Feature Individually:** For each feature not currently in the selected set, train the chosen machine learning model using only that single feature. Evaluate the model's performance on a validation set or using cross-validation.
3. **Select the Best Feature:** Choose the feature that results in the best model performance (according to your chosen metric, e.g., accuracy, R-squared) and add it to the set of selected features.

4. **Repeat:** Repeat step 2 and 3. Now, for each of the remaining unselected features, train the model using the currently selected features *plus* the one additional feature being considered. Evaluate the performance.
5. **Add the Best Remaining Feature:** Select the unselected feature that, when added to the current set, yields the greatest improvement in model performance. Add this feature to the selected set.
6. **Stopping Criterion:** Continue steps 4 and 5 until a predefined stopping criterion is met. Common stopping criteria include:
   - Reaching a desired number of features.
   - When adding a new feature no longer significantly improves the model's performance (the improvement falls below a certain threshold).
   - When the performance on the validation set starts to decrease (indicating potential overfitting).
7. **Final Selected Features:** The set of features selected when the stopping criterion is met is the final set of features chosen by forward selection.

Example: Predicting House Prices (Regression)

Let's say we want to predict the price of a house (numerical target) using the following numerical features:

- Size of the house (in square feet)
- Number of bedrooms
- Number of bathrooms
- Age of the house (in years)
- Location score (a numerical score representing the desirability of the location)

We'll use Linear Regression as our base model and R-squared as our evaluation metric.

Steps:

1. **Start Empty:** Selected features = {}
2. **Individual Evaluation:**
   - Train a Linear Regression model using only "Size of the house" and evaluate R-squared (e.g., $R^2 = 0.70$).
   - Train a Linear Regression model using only "Number of bedrooms" and evaluate R-squared (e.g., $R^2 = 0.55$).
   - Train a Linear Regression model using only "Number of bathrooms" and evaluate R-squared (e.g., $R^2 = 0.60$).

- Train a Linear Regression model using only "Age of the house" and evaluate R-squared (e.g., $R^2 = 0.25$).
- Train a Linear Regression model using only "Location score" and evaluate R-squared (e.g., $R^2 = 0.78$).

3. **Select Best First Feature:** "Location score" yields the highest R-squared (0.78), so we add it to our selected set. Selected features = {"Location score"}
4. **Iteration 1 - Evaluate Adding One More Feature:**
   - Train a Linear Regression model using {"Location score", "Size of the house"} and evaluate R-squared (e.g., $R^2 = 0.85$).
   - Train a Linear Regression model using {"Location score", "Number of bedrooms"} and evaluate R-squared (e.g., $R^2 = 0.80$).
   - Train a Linear Regression model using {"Location score", "Number of bathrooms"} and evaluate R-squared (e.g., $R^2 = 0.82$).
   - Train a Linear Regression model using {"Location score", "Age of the house"} and evaluate R-squared (e.g., $R^2 = 0.79$).
5. **Add Best Second Feature:** Adding "Size of the house" to "Location score" gives the greatest improvement in R-squared (from 0.78 to 0.85). So, we add it. Selected features = {"Location score", "Size of the house"}
6. **Iteration 2 - Evaluate Adding One More Feature:**
   - Train a Linear Regression model using {"Location score", "Size of the house", "Number of bedrooms"} and evaluate R-squared (e.g., $R^2 = 0.86$).
   - Train a Linear Regression model using {"Location score", "Size of the house", "Number of bathrooms"} and evaluate R-squared (e.g., $R^2 = 0.87$).
   - Train a Linear Regression model using {"Location score", "Size of the house", "Age of the house"} and evaluate R-squared (e.g., $R^2 = 0.855$).
7. **Add Best Third Feature:** Adding "Number of bathrooms" provides the largest improvement (from 0.85 to 0.87). Selected features = {"Location score", "Size of the house", "Number of bathrooms"}
8. **Continue until Stopping Criterion:** We would continue this process, evaluating the addition of the remaining features ("Number of bedrooms", "Age of the house") and stopping when the improvement in R-squared becomes negligible or we reach a desired number of features.

Advantages of Forward Selection (Wrapper Method):

- **Computationally More Efficient than RFE (initially):** It starts with a small number of features and gradually adds, which can be faster than RFE, especially when the number of relevant features is small compared to the total number of features.
- **Simple to Implement:** The logic is relatively straightforward.
- **Can be useful when the number of features is very large:** It might be more feasible to start with no features and build up.

Disadvantages of Forward Selection (Wrapper Method):

- **Greedy Approach:** It's a greedy algorithm, meaning it makes locally optimal choices at each step. A feature that seems most useful early on might not be part of the globally optimal set of features. A feature that is not very informative by itself might be very useful when combined with other features, but forward selection might miss this.
- **"Stuck" Once a Feature is Added:** Once a feature is added to the selected set, it cannot be removed later in the process, even if adding subsequent features makes the initially chosen feature redundant or less important.
- **Model Dependent:** The selected feature subset is specific to the chosen machine learning model.

In summary, Forward Selection is a wrapper method that iteratively builds a feature set by adding the feature that provides the greatest improvement in model performance at each step. It's a computationally less expensive alternative to RFE in some scenarios but has the limitation of being a greedy approach that cannot revisit earlier feature selections.