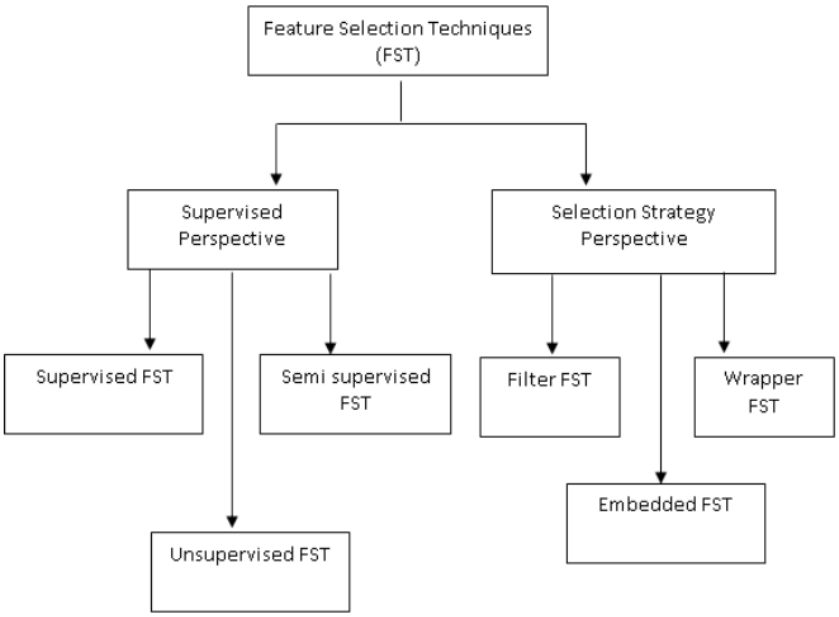
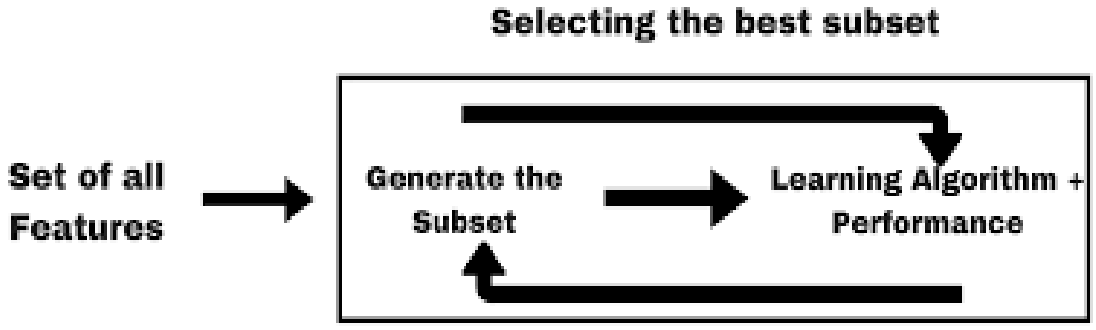
1st Week work:

Features Selection

Feature selection is the process of removing redundant or irrelevant features from the original data set. So, the execution time of the classifier that processes the data reduces, also accuracy increases because irrelevant features can include noisy data affecting the classification accuracy negatively.



Embedded Features Selection



embedded feature selection techniques seamlessly integrate the process of identifying important features with the model training phase. By doing so, they enhance the efficiency and effectiveness of the learning process, producing models that are not only predictive but also optimized for the most relevant features in the given dataset.

How Does It Work?

•Imagine you have a bunch of features (information) about something, like the size, color, and weight of fruits.

•Embedded methods, like Lasso or certain types of trees, learn to predict outcomes (like whether some fruit is tasty) while also figuring out which features are the most useful.

Example 1: Lasso Regression

•Lasso looks at the features and decides some are not very helpful, so it pushes their importance down to zero. It's like saying, "We don't need this information; it's not helping us much."

Example 2: Tree-based Methods (Random Forest)

•Imagine a decision tree that's trying to classify fruits. It asks questions like "Is it big?" or "Is it red?" to figure out what fruit it is.

•Features that help the tree make accurate guesses become more important, while less helpful features get less attention.

Why is it Useful?

•It makes the learning process more efficient and the model more focused on what really matters.

•It's like training a chef to cook by telling them, "Pay attention to these specific ingredients; the rest are not crucial."

In a nutshell, embedded feature selection is like teaching a computer to not only make predictions but also decide which information matters the most for those predictions. It's about finding the signal in the noise!

2nd Week Work:

Embedded methods differ from other feature selection methods in the way feature selection and learning interact. **Filter** methods do not incorporate learning. **Wrapper** methods use a learning machine to measure the quality of subsets of features without incorporating knowledge about the specific structure of the classification or regression function and can therefore be combined with any learning machine. In contrast to filter and wrapper approaches, in embedded methods the learning part and the feature selection part cannot be separated - the structure of the class of functions under consideration plays a crucial role.

Advantages of Embedded Methods

+ They take into consideration the interaction of features like wrapper methods do.

+ They are faster like filter methods.

+ They are more accurate than filter methods.

+ They find the feature subset for the algorithm being trained.

+ They are much less prone to over-fitting.

Approaches of Embedded Methods

Regularization Approach:

the Regularization approach that includes LASSO (**Least Absolute Shrinkage and Selection Operator**) (L1 regularization) and Ridge (L2 regularization) and Elastic Nets (L1 and L2).

1 – LASSO (L1 penalty*):*

+ LASSO helps us build a good prediction model while keeping things simple. It's like finding the best-fitting line while also choosing only the most important factors.

+ LASSO uses a trick called a penalty. This penalty makes some factors (or features) in our model really small or even zero.

+ LASSO adds a special extra part to the equation that punishes big numbers. This extra part is controlled by a number 'λ'.

$$\sum_{i=1}^{M}(y_i-\hat{y}_i)^2=\sum_{i=1}^{M}\left(y_i-\sum_{j=0}^{p}w_j\times x_{ij}\right)^2+\lambda\sum_{j=0}^{p}w_j^2$$

Cost function for ridge regression

And This is how LASSO helps in reducing the over-fitting caused and as well as help in feature selection.

2 – RIDGE (L2):

+ RIDGE penalizes the beta co-efficients for being too large, however it does not bring down(shrinks) the co-efficient to zero rather it brings the co-efficients close to zero.

+ Ridge Regression uses a trick to make sure the numbers representing our features don't become too big. This helps prevent the model from becoming too sensitive to the data.

+ Just like LASSO, Ridge Regression adds a special extra part to the model equation. This extra part, however, punishes big numbers differently. It's controlled by a number 'λ'.

+ RIDGE is not preferable when the data contains huge number of features out of which only few are actually important, as it might make the model simpler, but the model built will have poor accuracy.

3 - Elastic Nets (L1 AND L2):

Lasso has been a popular algorithm for variable selection with high dimensional data. However, it sometimes over regularizes the data.

So, the question arises: What if we could use both L1 and L2 regularization?
Elastic Nets was introduced as a solution to this question.

+ Elastic Nets balance between LASSO and RIDGE penalties.

+ Lasso will eliminate features and reduce over-fitting in the linear model. Ridge will reduce the impact of features that are not important in predicting the target values.

+ This is done with the help of the hyper parameter alpha(α). If α becomes 1 the model would become LASSO and when α becomes 0 the model will become RIDGE.

$$\textbf{L1 Penalty: } R(w) := \frac{1}{2}\sum_{i=1}^{n}|w_i|$$

$$\textbf{L2 Penalty: } R(w) := \frac{1}{2}\sum_{i=1}^{n}w_i^2$$

$$\textbf{Elastic-Net Penalty:}$$

$$R(w) := \frac{\varphi}{2}\sum_{i=1}^{n}w_i^2 + (1-\varphi)\sum_{i=1}^{n}|w|$$

A convex combination of L1 and L2 Penalty.

Algorithm Based Approach:

+ This can be done using any kind of tree-based algorithm like Decision Tree, RandomForest or ExtraTree, XGBoost and so on.

+ The split takes place on a feature within the algorithm to find the correct variable.

+ The algorithm tries all possible ways of splitting for all the features and chooses the one that splits the data best. This basically means it uses wrapper method as all the possible combinations of features are tried and the best one is picked.

+ With the help of this method we can find feature importance's and can remove feature below certain threshold.

3rd Week Work:

LASSO (L1 penalty):

LASSO Regularization is commonly used as a feature selection criterion. It penalizes irrelevant parameters by shrinking their weights or coefficients to zero. Hence, those features are removed from the model.

Advantages of LASSO in Feature Selection:

Automatic Feature Selection: LASSO automatically selects and prioritizes important features by driving irrelevant feature coefficients to zero. This is beneficial in scenarios with a large number of features, streamlining **the** model.

Sparsity Promotion: The sparsity introduced by LASSO results in a simpler and more interpretable model, as it focuses on a subset of the most influential features.

Deals with Multicollinearity: LASSO effectively handles multicollinearity by selecting one variable from a group of highly correlated variables and setting others to zero.

Drawbacks of LASSO in Feature Selection:

Arbitrary Variable Selection: In the presence of highly correlated variables, LASSO may arbitrarily choose one variable over another for inclusion in the model, leading to instability in feature selection.

Parameter Tuning Challenges: Choosing the right regularization parameter (λ) can be challenging. The performance of the feature selection process may be sensitive to the specific choice of λ, requiring careful tuning.

Limited in Handling All Correlated Features: LASSO tends to select one feature from a group of highly correlated features and exclude the others. This may not be suitable in situations where retaining all correlated features is essential for the model's interpretability or performance.

RIDGE (L2 penalty):

Ridge Regression helps create a stable and balanced model by adding a penalty for large coefficients. Unlike LASSO, Ridge doesn't force any features to be completely excluded. It ensures that all features contribute, although with less impact, preventing extreme values. Ridge is especially useful when dealing with highly correlated features, as it redistributes their weight instead of eliminating them.

Advantages of Ridge Regression in Feature Selection:

Handles Multicollinearity: Ridge regularization effectively addresses multicollinearity concerns by redistributing feature weights, allowing all variables to contribute meaningfully without arbitrary exclusion.

No Feature Exclusion: Ridge regression avoids the complete exclusion of any feature, ensuring that all variables play a role in the model, albeit with reduced impact.

Stability in Coefficient Estimates: When the model is being trained to fit the data, Ridge not only minimizes the difference between predicted and actual values (Residual Sum of Squares, RSS) but also minimizes the sum of the squares of the coefficients, thanks to the penalty term.

Drawbacks of Ridge Regression in Feature Selection:

Limited Feature Selection: Ridge regression does not explicitly select features by setting coefficients to zero, lacking the sparsity-inducing benefits seen in LASSO.

Interpretability Challenges: The continuous shrinkage of coefficients in Ridge makes it challenging to interpret the individual importance of features.

Dependency on Proper Scaling: Similar to LASSO, Ridge is sensitive to feature scaling. Ensuring consistent standardization or normalization is vital for unbiased regularization effects.

ELASTIC NETS (L1 AND L2 penalty):

Elastic Net is a powerful technique designed for feature selection It strikes a balance between LASSO and Ridge regression, stands out as an effective tool for feature selection due to its automatic selection capability, ability to handle correlated features, and the fine-tuning options provided by its two hyperparameters. It offers a balanced approach that combines sparsity and stability, making it well-suited for complex datasets in the field of feature selection.

Advantages of ELASTIC NETS Regression in Feature Selection:

Two Hyperparameters: Elastic Net introduces two hyperparameters (α and λ), controlling the trade-off between L1 and L2 penalties and the overall strength of regularization. This flexibility enables fine-tuning for different datasets and modeling objectives.

Versatility in Complex Data Sets: Elastic Net is particularly versatile in datasets where features are numerous, potentially irrelevant, or correlated. Its ability to adapt to varying degrees of sparsity and multicollinearity makes it suitable for a wide range of real-world scenarios.

Automatic Feature Selection: Elastic Net, like LASSO, automatically selects and prioritizes relevant features by driving irrelevant feature coefficients towards zero. This is advantageous in scenarios with a large number of features, aiding in the creation of a streamlined and efficient model.

Drawbacks of Elastic Nets Regression in Feature Selection:

Increased Complexity: The inclusion of two hyperparameters (α and λ) adds complexity to the model tuning process. Finding the right balance between L1 and L2 penalties may require more computational resources and careful optimization.

Dependency on Proper Scaling: Elastic Net, like its individual components (LASSO and Ridge), is sensitive to the scale of features. Ensuring proper standardization or normalization is essential to avoid biased regularization effects based on feature magnitudes.

taxonomy of embedded features selection:

Method	Description	Advantages	Disadvantages	Parameters	Characteristics	Applications
Lasso	Linear regression model with L1 regularization	- Automatic feature selection	- Limited feature selection capabilities	alpha	Provides sparse solutions	Feature selection, high-dimensional data
Ridge	Linear regression model with L2 regularization	- Reduces multicollinearity	- Does not perform feature selection	alpha	Tends to shrink coefficients towards zero	Multicollinear data, regularization
ElasticNet	Linear regression model with combined L1 and L2 regularization	- Balance between Lasso and Ridge	- Requires tuning of alpha and l1_ratio parameters	alpha, l1_ratio	Combines advantages of Lasso and Ridge	High dimensional data, mixed features
LARS	Least Angle Regression	- Efficient for large datasets	- May struggle with multicollinearity	None	Iteratively selects features based on correlations	Large datasets, high-dimensional data
LARS Lasso	LARS with L1 regularization	-Suitable for high-dimensional datasets	- May not perform well with highly correlated features	None	Incorporates Lasso regularization in LARS	Large datasets, high-dimensional data
Random Forest	Ensemble learning method using decision trees	- Handles nonlinear relationships	- May overfit on noisy data	n_estimators, max_depth	Utilizes multiple decision trees for feature importance	Classification, regression, feature importance
Gradient Boosting	Ensemble learning method using decision trees	- Handles missing data and mixed types	- Sensitive to hyperparameters	n_estimators, learning_rate	Builds models sequentially to correct errors	Classification, regression, ranking

XGBoost	Gradient boosting algorithm	- High performance	- Requires tuning of hyperparameters	n_estimators, learning_rate	Implements regularized boosting	Classification, regression, ranking
LightGBM	Gradient boosting framework	- Fast training and high efficiency	- Sensitive to overfitting	n_estimators, learning_rate	Utilizes histogram-based splitting for speed	Large datasets, categorical features
CatBoost	Gradient boosting library	- Handles categorical features naturally	- Requires careful parameter tuning	n_estimators, learning_rate	Implements ordered boosting with categorical features	Categorical features, ranking, classification

5th Week:

Data Loading and Preprocessing:

+ Original dataset is loaded from a CSV file.

+ Each row represents a customer, each column contains customer's attributes.

+ The raw data contains 201 rows (customers) and 28 columns (features).

+ The "Churn" column is our target.

+ if the churn is 0 The customer **has not churned** | 1 has **churned**.

+ Features and target variables are segregated.

+ Dataset is split into training and testing sets.

Model Training and Evaluation with HistGradientBoostingClassifier:

+ A HistGradientBoostingClassifier model is chosen and trained on the training data.

+ Predictions are made on the test set.

+ Accuracy of the original model is computed and printed: 0.75.

Feature Selection with Lasso (L1 Regularization):

+ Feature selection is performed using Lasso regularization (L1) with LassoCV.

+ Features with non-zero coefficients are selected.

+ The HistGradientBoostingClassifier model is trained on the selected features.

+Model accuracy after feature selection with Lasso is computed and printed:0.85.

Feature Importance with Random Forest:

+ Feature importance is calculated using a Random Forest classifier.

+ Feature importance scores are ranked and plotted.

+ The accuracy of the original model is printed:0.825.

Feature Selection with ElasticNet:

+Feature selection is performed using ElasticNet regularization.

+Features with non-zero coefficients are selected.

+The HistGradientBoostingClassifier model is trained on the selected features.

+Model accuracy after feature selection with ElasticNet is computed and printed:0.85.

Feature Selection with Recursive Feature Elimination (RFE):

+Feature selection is performed using Recursive Feature Elimination with a Random Forest classifier.

+The optimal number of features is selected using grid search.

+The HistGradientBoostingClassifier model is trained on the selected features.

+Model accuracy after feature selection with RFE is computed and printed:0.825.

Feature Importance with XGBoost:

+Feature importance scores are obtained using XGBoost.

+Feature importance scores are plotted.

+The accuracy of the original model is printed:0.80.

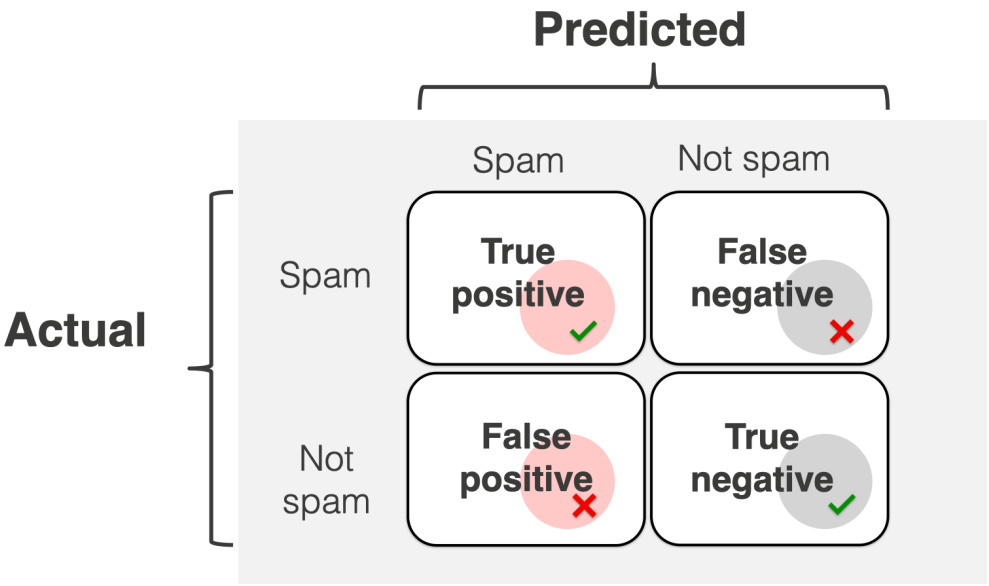
EVALUATION MEASURES:

Classification Tasks:

+ *Accuracy:* Overall proportion of correct predictions (may not be ideal for imbalanced classes).

$$\text{Accuracy} = \frac{\text{Correct predictions}}{\text{All predictions}}$$

+ *Confusion Matrix:* Visualizes the model's performance on each class (true positives, false positives, true negatives, false negatives).



+ *Precision:* Ratio of true positives to all positive predictions (useful for identifying relevant items).

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

+ *Recall:* Ratio of true positives to all actual positive cases (important for catching all positive cases).

+ *F1-Score:* Harmonic mean of precision and recall, balancing both metrics.

Regression Tasks:

+ *Mean Squared Error (MSE):* Average squared difference between predicted and actual values.

+ *Root Mean Squared Error (RMSE):* Square root of MSE, in the same units as the target variable.

+ *Mean Absolute Error (MAE):* Average absolute difference between predicted and actual values, less sensitive to outliers than MSE.

+ *R-Squared (R^2):* Proportion of variance in the target variable explained by the model (ranges from 0 to 1, higher is better).

For the next part of this research, I'll use a different dataset that has more data and features

telecommunications dataset for predicting customer churn with feature selection.

    - Customers who left within the last month — the column is called Churn

- Services that each customer has signed up for — phone, multiple lines, internet, online security, online backup, device protection, tech support, and streaming TV and movies

- Customer account information — how long they had been a customer, contract, payment method, paperless billing, monthly charges, and total charges

- Demographic info about customers — gender, age range, and if they have partners and dependents