Feature Subset Selection

Notes by Mannan Ul Haq (22L-7556)

Dimensionality Reduction:

- **Definition**: Dimensionality Reduction is the process of reducing the number of features (dimensions) in a dataset while retaining its essential information.
- **Objective**: Overcome the **"curse of dimensionality"**, improve model efficiency, and enhance interpretability.
- Data preprocessing is an important part for effective machine learning and data mining.
- Dimensionality reduction is an effective approach to downsizing data.

Why Dimensionality Reduction?

- **Curse of Dimensionality reduction**: As the number of features decreases, the data becomes more accurate, and computational complexity decreases.
- **Noise Removal**: Some features may be noisy, leading to overfitting. So noise removal provides more accuracy in data.
- **Data compression:** Efficient storage and retrieval.
- **Visualization:** Projection of high-dimensional data onto 2D or 3D.
- Reduces training time in machine learning.

Techniques:

- **Feature Selection:** Choosing a subset of relevant features while discarding irrelevant or redundant ones (Only a subset of the original features are selected).
- **Feature Extraction/Reduction:** Feature Extraction/Reduction refers to the process of transforming or combining original features to create a more compact and representative set of features in a dataset.

Feature Selection

Feature or Variable Selection refers to the process of selecting features that are used in predicting the target or output.

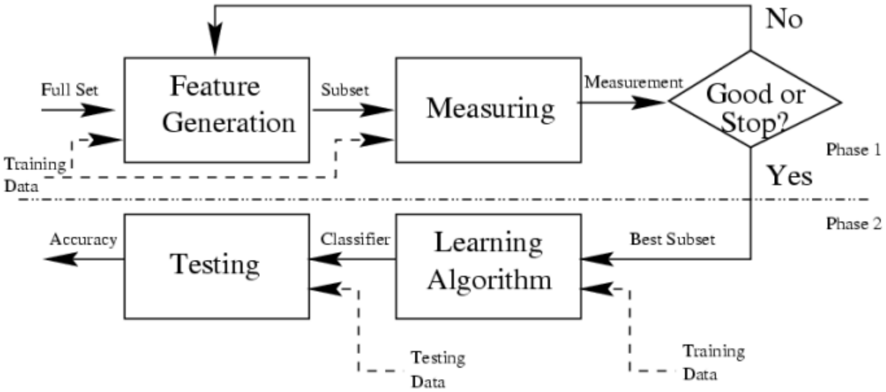
The purpose of Feature Selection is to select the features that contribute the most to output prediction.

- **Methods**:
  - Filter Methods
  - Wrapper Methods
  - Embedded Methods

Filter Methods:

- **Definition**: The Filter Methods involve selecting features based on their various statistical scores with the output column.
- The filter method ranks each feature based on some uni-variate metric and then selects the highest-ranking features. We identify and retain the most relevant features based on statistical measures or predefined criteria.
- The selection of features is independent of any Machine Learning algorithm.
- **Rules:**
  - The more the features are correlated with the output column or the column to be predicted, the better the performance of the model.
  - Features should be least correlated with each other. If some of the input features are correlated with some additional input features, this situation is known as **Multicollinearity**. It is recommended to get rid of such a situation for better performance of the model.
- **Filter Selection Select independent features with:**
  - No constant variables
  - No duplicate rows
  - High correlation with the target variable

    - Low correlation with another independent variable
    - Higher information gain or mutual information of the independent variable
- **mRMR Score:**
    - mRMR (Maximizing Relevance, Minimizing Redundancy) is a feature selection method that aims to find a balance between the relevance of features to the target variable and the redundancy among selected features.
    - **Relevance**:
        - Captures how well a feature discriminates or predicts the target variable.
        - Higher relevance implies a strong association with the target.
    - **Redundancy**:
        - Measures the similarity or correlation between selected features.
        - Lower redundancy indicates less overlap in information content.
    - **mRMR Score**:
        - High mRMR score signifies a feature's effectiveness in contributing unique information.



Common Filter Methods:

1. Removing features with low variance (Low variance filter):

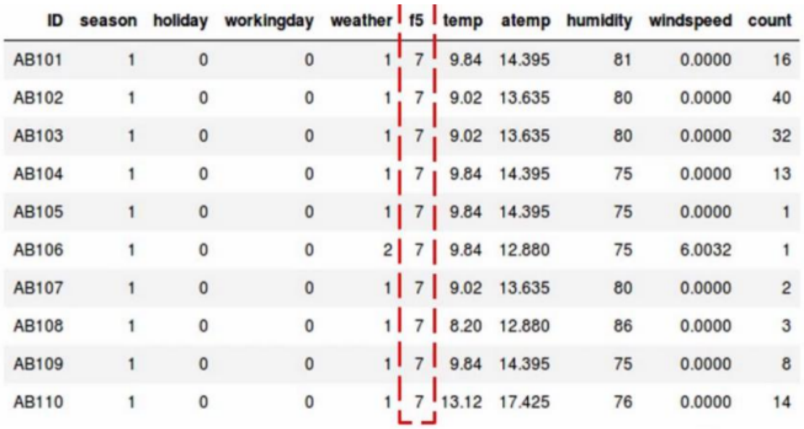
Definition: The Low Variance Filter is a feature selection method that focuses on identifying and eliminating features with minimal variance across the dataset. Features with low variance often carry limited information and may not contribute significantly to the modeling process.

Key Components:

- **Variance**:
  - Measures the spread or variability of values within a feature.
  - High variance indicates a wide range of values, while low variance suggests a more constant or uniform feature.
- **Threshold**:
  - A predetermined threshold is set to determine the acceptable level of variance.
  - Features with variance below the threshold are considered low-variance features.

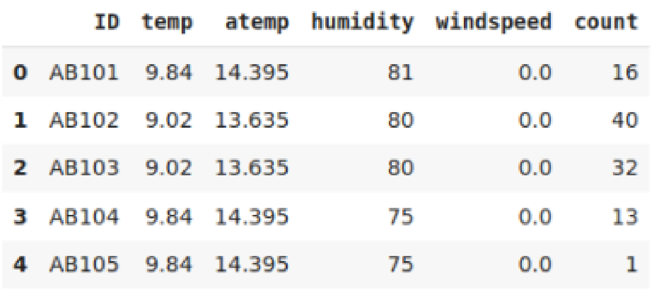
Steps:

- **Step 1:** Remove features with zero variance (Constants).



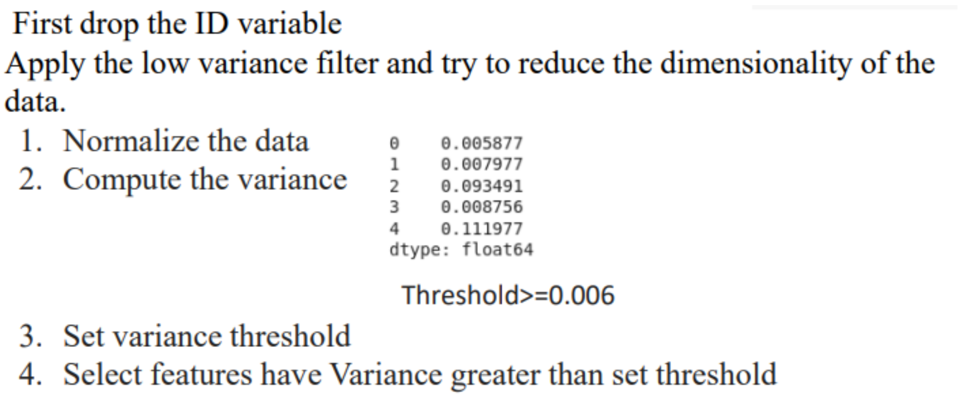
ID	season	holiday	workingday	weather	f5	temp	atemp	humidity	windspeed	count
AB101	1	0	0	1	7	9.84	14.395	81	0.0000	16
AB102	1	0	0	1	7	9.02	13.635	80	0.0000	40
AB103	1	0	0	1	7	9.02	13.635	80	0.0000	32
AB104	1	0	0	1	7	9.84	14.395	75	0.0000	13
AB105	1	0	0	1	7	9.84	14.395	75	0.0000	1
AB106	1	0	0	2	7	9.84	12.880	75	6.0032	1
AB107	1	0	0	1	7	9.02	13.635	80	0.0000	2
AB108	1	0	0	1	7	8.20	12.880	86	0.0000	3
AB109	1	0	0	1	7	9.84	14.395	75	0.0000	8
AB110	1	0	0	1	7	13.12	17.425	76	0.0000	14

- **Step 2:** For the remaining features, first Normalize the data and compute the Variance for each feature.



	ID	temp	atemp	humidity	windspeed	count
0	AB101	9.84	14.395	81	0.0	16
1	AB102	9.02	13.635	80	0.0	40
2	AB103	9.02	13.635	80	0.0	32
3	AB104	9.84	14.395	75	0.0	13
4	AB105	9.84	14.395	75	0.0	1

- **Step 3:** Choose a threshold value that defines the minimum acceptable variance. Common choices include user-defined values or a percentage of the maximum variance.



First drop the ID variable

Apply the low variance filter and try to reduce the dimensionality of the data.

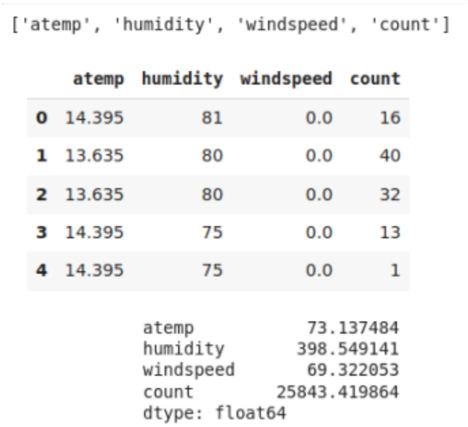
1. Normalize the data
2. Compute the variance

```
0     0.005877
1     0.007977
2     0.093491
3     0.008756
4     0.111977
dtype: float64
```

Threshold>=0.006

3. Set variance threshold
4. Select features have Variance greater than set threshold

- **Step 4:** Identify features with variance below the selected threshold. Remove these low-variance features from the dataset.



```
['atemp', 'humidity', 'windspeed', 'count']
```

	atemp	humidity	windspeed	count
0	14.395	81	0.0	16
1	13.635	80	0.0	40
2	13.635	80	0.0	32
3	14.395	75	0.0	13
4	14.395	75	0.0	1

```
atemp           73.137484
humidity       398.549141
windspeed       69.322053
count        25843.419864
dtype: float64
```

2. Pearson's Correlation Coefficient:

Definition: Pearson's Correlation Coefficient, often denoted as r, is a statistical measure that quantifies the strength and direction of a linear relationship between two continuous variables.

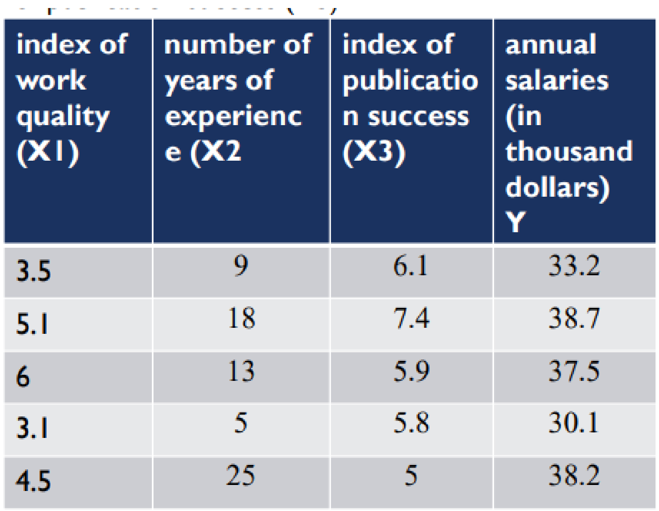
Objective: To determine the degree of linear association between two variables, with values ranging from **-1** (perfect negative correlation) to **1** (perfect positive correlation), and **0** indicating no linear correlation.

Formula:

- The Pearson correlation coefficient is calculated using the following formula:

$$r_{A,B} = \frac{\Sigma(A-\overline{A})(B-\overline{B})}{(n-1)\sigma_A\sigma_B}$$

Example: A researcher in a scientific foundation wished to evaluate the relation between annuals salaries of mathematicians (Y, in thousand dollars) and an index of work quality (X1), number of years of experience (X2), and an index of publication success (X3).

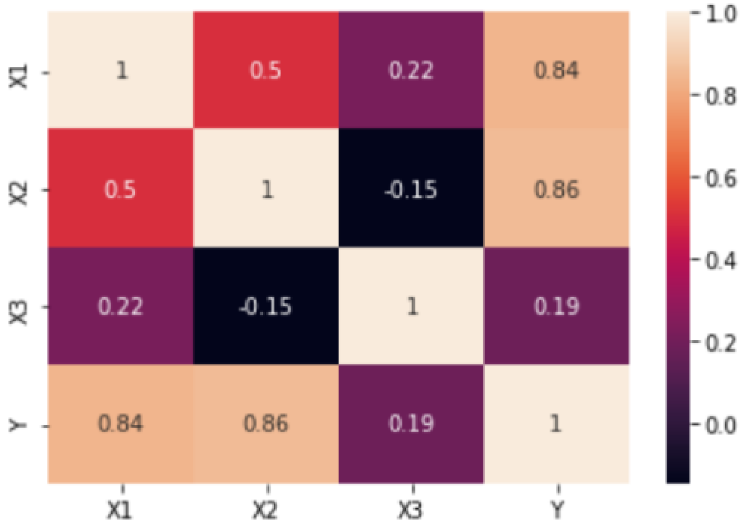


index of work quality (X1)	number of years of experienc e (X2	index of publicatio n success (X3)	annual salaries (in thousand dollars) Y
3.5	9	6.1	33.2
5.1	18	7.4	38.7
6	13	5.9	37.5
3.1	5	5.8	30.1
4.5	25	5	38.2

Heat Map:

A Heat Map is a graphical representation that visualizes the strength and direction of relationships between multiple variables, often using color gradients.

- Calculate Pearson's correlation coefficient for all pairs of variables in the dataset.
- Represent the correlation matrix as a heat map, where each cell color corresponds to the correlation value.



- **Handling Multicollinearity**:
  - Address multicollinearity issues, especially when features are highly correlated.
  - Choose features with the least redundancy.

3. Mutual Information (Information Gain):

Definition: Mutual Information measures the amount of information obtained about one variable through the observation of another variable. In the context of feature selection, it quantifies the degree of dependence between a feature and the target variable.

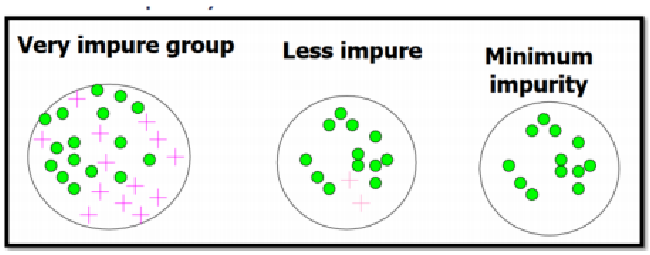
The feature having the most information is considered important by the algorithm and is used for training the model.

The effort is to reduce the entropy and maximize the information gain.

Entropy:

- A measure of uncertainty or randomness in a variable.
- High entropy indicates greater uncertainty.

- Information gain uses entropy to make decisions. If the entropy is less, information will be more.

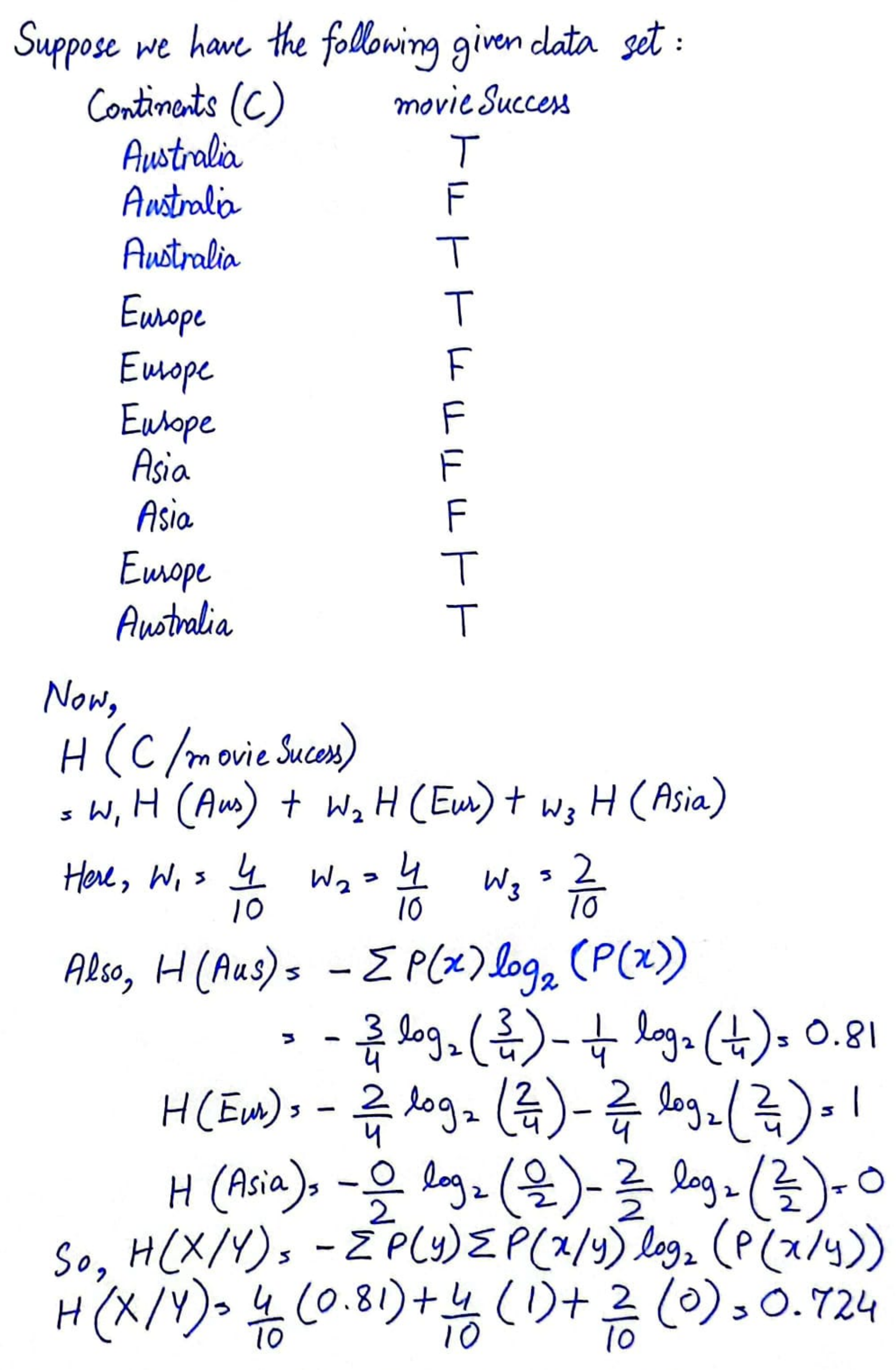


Formula:

- The Mutual Information between two variables *X* and *Y* is calculated using the following formula:

$$IG(X|Y) = \mathbf{1} - H(X|Y)$$

Example:



Suppose we have the following given data set :

Continents (C)	movie Success
Australia	T
Australia	F
Australia	T
Europe	T
Europe	F
Europe	F
Asia	F
Asia	F
Europe	T
Australia	T

Now,

$H\left(C/\text{movie Sucess}\right)$

$= W_1 H\left(Aus\right) + W_2 H\left(Eur\right) + W_3 H\left(Asia\right)$

Here, $W_1 = \frac{4}{10} \quad W_2 = \frac{4}{10} \quad W_3 = \frac{2}{10}$

Also, $H\left(Aus\right) = -\sum P(x) \log_2 \left(P(x)\right)$

$= -\frac{3}{4} \log_2 \left(\frac{3}{4}\right) - \frac{1}{4} \log_2 \left(\frac{1}{4}\right) = 0.81$

$H\left(Eur\right) = -\frac{2}{4} \log_2 \left(\frac{2}{4}\right) - \frac{2}{4} \log_2 \left(\frac{2}{4}\right) = 1$

$H\left(Asia\right) = -\frac{0}{2} \log_2 \left(\frac{0}{2}\right) - \frac{2}{2} \log_2 \left(\frac{2}{2}\right) = 0$

So, $H(X/Y) = -\sum P(y) \sum P(x/y) \log_2 \left(P(x/y)\right)$

$H(X/Y) = \frac{4}{10}(0.81) + \frac{4}{10}(1) + \frac{2}{10}(0) = 0.724$

→ Information Gain

$$IG(X/Y) = 1 - H(X/Y)$$

$$= 0.28$$

Advantages of Filter methods:

- Filter methods are model agnostic(compatible)
- Rely entirely on features in the data set
- Computationally very fast
- Based on different statistical methods

Disadvantages of Filter methods:

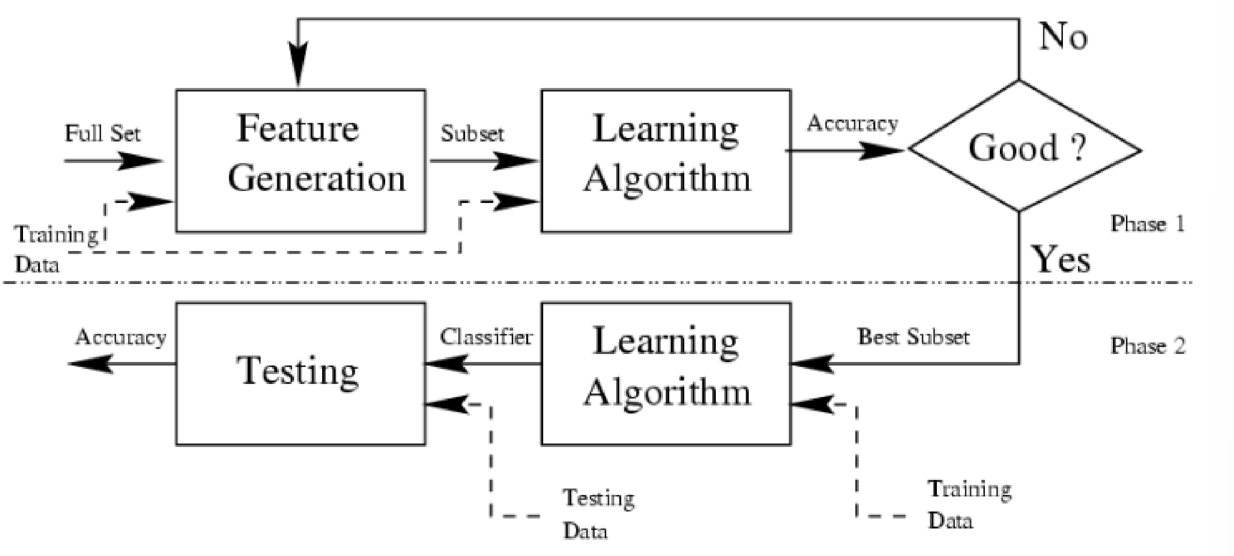
- The filter method looks at individual features for identifying it's relative importance. A feature may not be useful on its own but may be an important influencer when combined with other features. Filter methods may miss such features.
- One thing that should be kept in mind is that the filter method does not correctly remove multicollinearity. So, you must deal with the multicollinearity of features as well before training models for your data.

Wrapper Methods:

Definition: In Wrapper Methods the problem of Feature Selection is reduced to a search problem.

A model is built using a set of features and its accuracy is recorded.

Based on the accuracy, more features are added or removed, and the process is repeated.

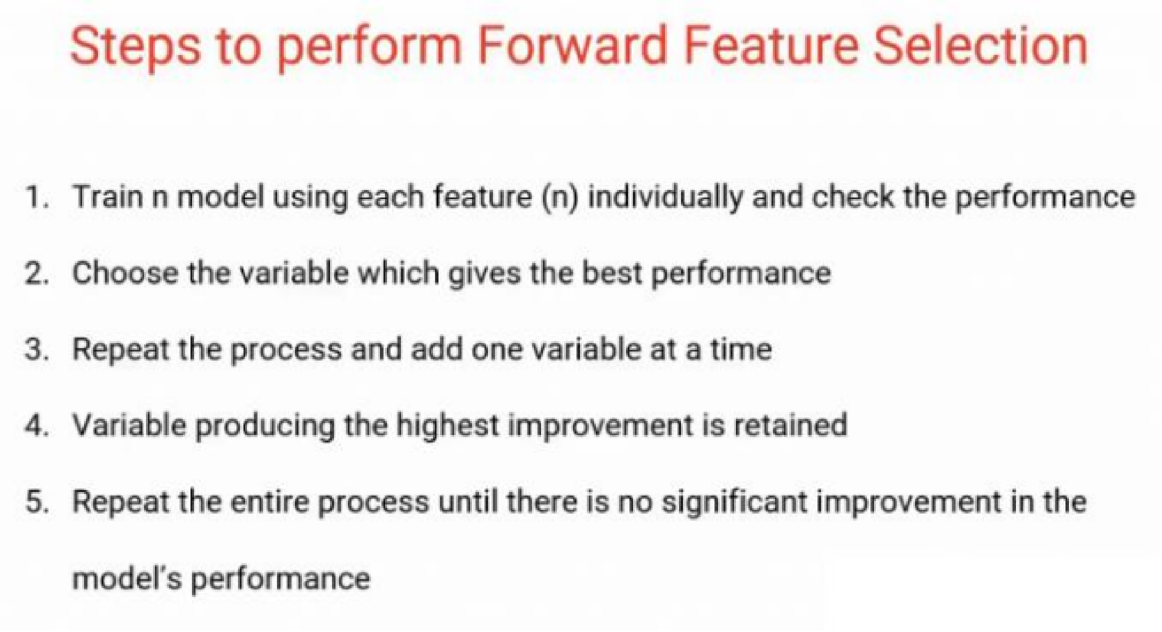


Common Wrapper Methods:

- **Forward Selection**:
  - Forward Selection is an iterative method.
  - In this method, we start with one feature and we keep on adding features until no improvement in the model is observed.
  - The search is stopped after a pre-set criteria is met.
  - This is a greedy approach because it always targets the features in a forward fashion, which gives a boost to the performance.
  - If the number of features are large, it can be computationally expensive.
- **Backward Elimination**:
  - This process is the opposite of the Forward Selection Method.
  - It starts initially with all the features and keeps on removing features until no improvement is observed.
- **Exhaustive Search:**
  - This Feature Selection Method tries all the possible combinations of features to select the best model.
  - This method is quite computationally expensive.
- **Recursive Feature Elimination**:

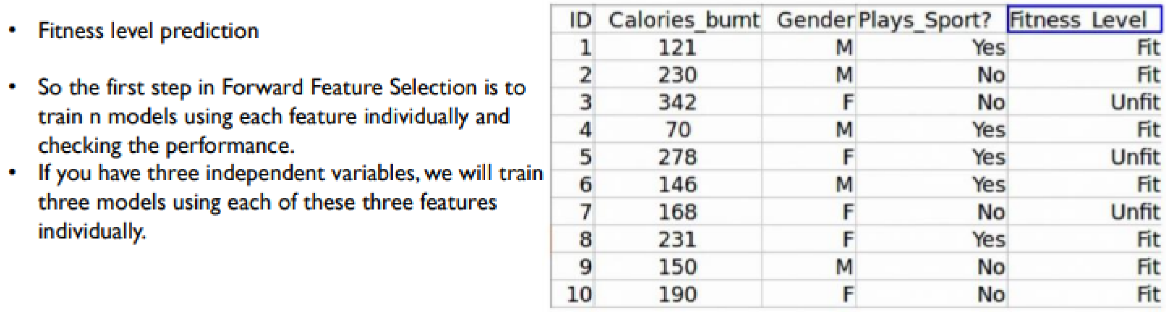
- We train the model on the initial set of features and the importance of each feature is calculated.
- In the second iteration, a model is built again using the most important features and excluding the least important features.
- These steps are repeated recursively until we are left with the most important features for the problem under consideration.

1. Forward Feature Selection:



Steps to perform Forward Feature Selection

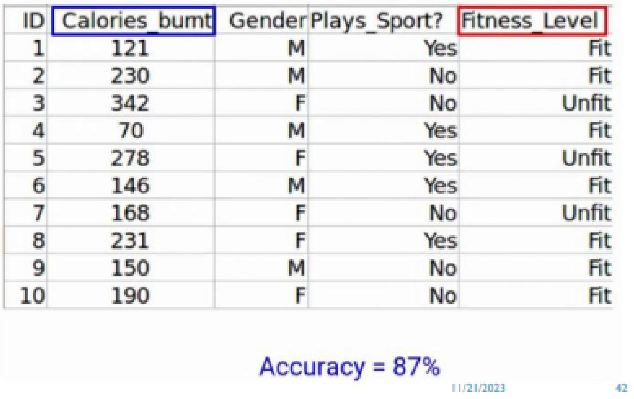
1. Train n model using each feature (n) individually and check the performance
2. Choose the variable which gives the best performance
3. Repeat the process and add one variable at a time
4. Variable producing the highest improvement is retained
5. Repeat the entire process until there is no significant improvement in the model's performance



- Fitness level prediction
- So the first step in Forward Feature Selection is to train n models using each feature individually and checking the performance.
- If you have three independent variables, we will train three models using each of these three features individually.

ID	Calories_bumt	Gender	Plays_Sport?	Fitness Level
1	121	M	Yes	Fit
2	230	M	No	Fit
3	342	F	No	Unfit
4	70	M	Yes	Fit
5	278	F	Yes	Unfit
6	146	M	Yes	Fit
7	168	F	No	Unfit
8	231	F	Yes	Fit
9	150	M	No	Fit
10	190	F	No	Fit

- Let's say we trained the model using the **Calories_Burnt** feature and the target variable, **Fitness_Level** and we've got an accuracy of **87%**

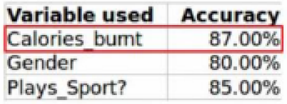


ID	Calories_bumt	Gender	Plays_Sport?	Fitness_Level
1	121	M	Yes	Fit
2	230	M	No	Fit
3	342	F	No	Unfit
4	70	M	Yes	Fit
5	278	F	Yes	Unfit
6	146	M	Yes	Fit
7	168	F	No	Unfit
8	231	F	Yes	Fit
9	150	M	No	Fit
10	190	F	No	Fit

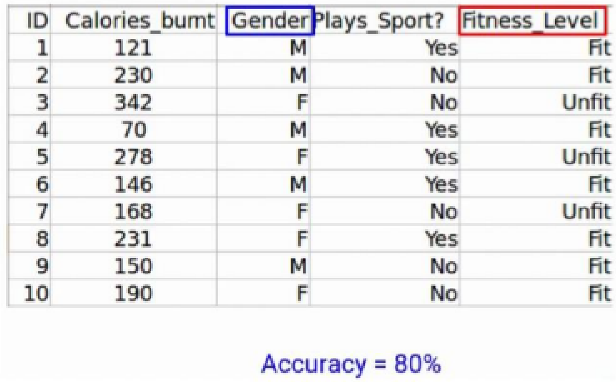
Accuracy = 87%

11/21/2023 42

Next, we'll train the model using the **Gender** feature, and we get an accuracy of **80%**



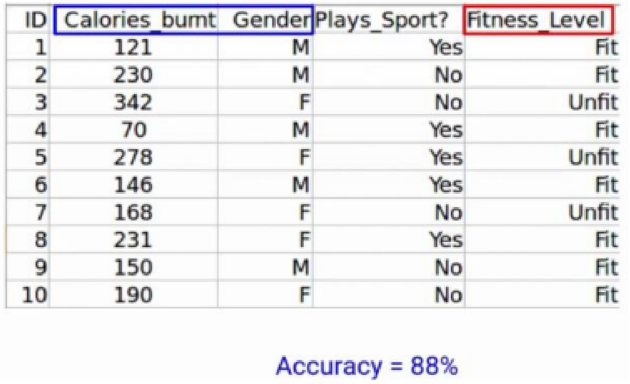
Variable used	Accuracy
Calories_bumt	87.00%
Gender	80.00%
Plays_Sport?	85.00%



ID	Calories_bumt	Gender	Plays_Sport?	Fitness_Level
1	121	M	Yes	Fit
2	230	M	No	Fit
3	342	F	No	Unfit
4	70	M	Yes	Fit
5	278	F	Yes	Unfit
6	146	M	Yes	Fit
7	168	F	No	Unfit
8	231	F	Yes	Fit
9	150	M	No	Fit
10	190	F	No	Fit

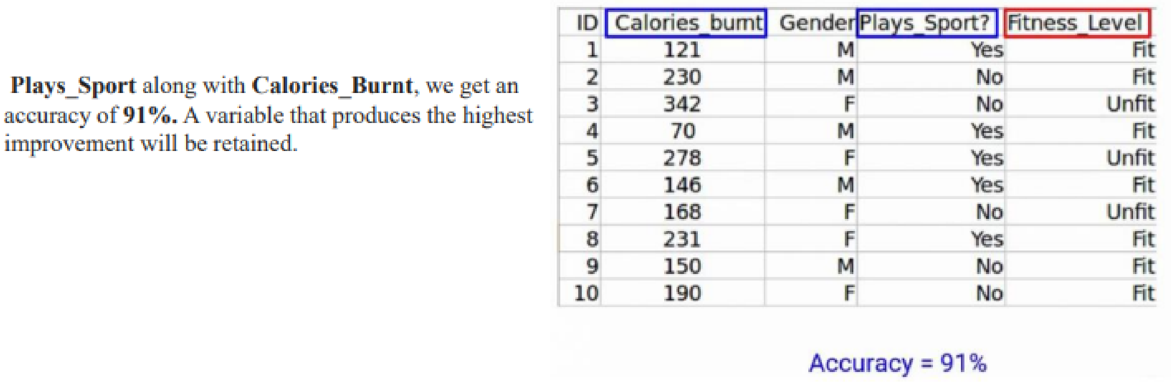
Accuracy = 80%

- Next, we will repeat this process and add one variable at a time. So of course we'll keep the **Calories_Burnt** variable and keep adding one variable. So let's take **Gender** here and using this we get an accuracy of **88%**-



ID	Calories_bumt	Gender	Plays_Sport?	Fitness_Level
1	121	M	Yes	Fit
2	230	M	No	Fit
3	342	F	No	Unfit
4	70	M	Yes	Fit
5	278	F	Yes	Unfit
6	146	M	Yes	Fit
7	168	F	No	Unfit
8	231	F	Yes	Fit
9	150	M	No	Fit
10	190	F	No	Fit

Accuracy = 88%



Plays_Sport along with **Calories_Burnt**, we get an accuracy of **91%.** A variable that produces the highest improvement will be retained.

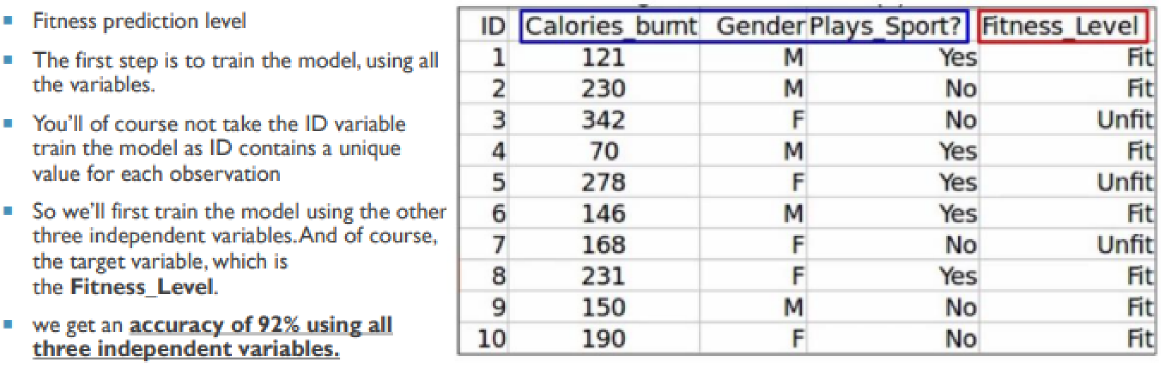
ID	Calories_bumt	Gender	Plays_Sport?	Fitness_Level
1	121	M	Yes	Fit
2	230	M	No	Fit
3	342	F	No	Unfit
4	70	M	Yes	Fit
5	278	F	Yes	Unfit
6	146	M	Yes	Fit
7	168	F	No	Unfit
8	231	F	Yes	Fit
9	150	M	No	Fit
10	190	F	No	Fit

Accuracy = 91%

2. Backward Feature Elimination:

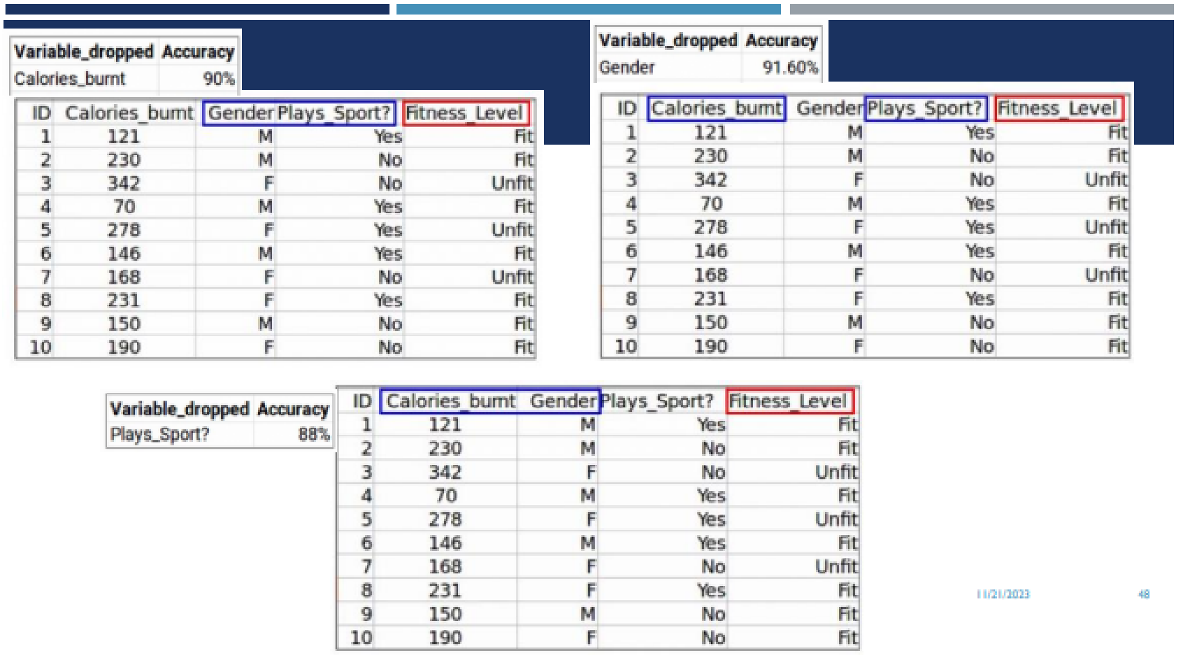
These are our assumptions:

- No missing values in the dataset
- Variance of the variables is high
- Low correlation between the independent variables



- Fitness prediction level
- The first step is to train the model, using all the variables.
- You'll of course not take the ID variable train the model as ID contains a unique value for each observation
- So we'll first train the model using the other three independent variables.And of course, the target variable, which is the **Fitness_Level**.
- we get an **accuracy of 92% using all three independent variables.**

ID	Calories_bumt	Gender	Plays_Sport?	Fitness_Level
1	121	M	Yes	Fit
2	230	M	No	Fit
3	342	F	No	Unfit
4	70	M	Yes	Fit
5	278	F	Yes	Unfit
6	146	M	Yes	Fit
7	168	F	No	Unfit
8	231	F	Yes	Fit
9	150	M	No	Fit
10	190	F	No	Fit



Variable_dropped	Accuracy
Calories_burnt	90%

ID	Calories_burnt	Gender	Plays_Sport?	Fitness_Level
1	121	M	Yes	Fit
2	230	M	No	Fit
3	342	F	No	Unfit
4	70	M	Yes	Fit
5	278	F	Yes	Unfit
6	146	M	Yes	Fit
7	168	F	No	Unfit
8	231	F	Yes	Fit
9	150	M	No	Fit
10	190	F	No	Fit

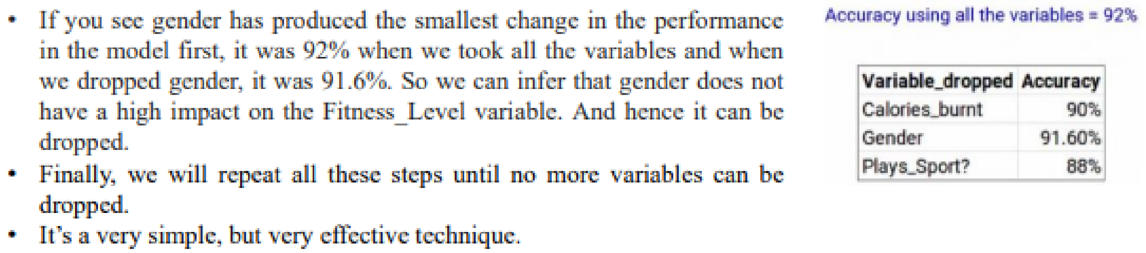
Variable_dropped	Accuracy
Gender	91.60%

ID	Calories_burnt	Gender	Plays_Sport?	Fitness_Level
1	121	M	Yes	Fit
2	230	M	No	Fit
3	342	F	No	Unfit
4	70	M	Yes	Fit
5	278	F	Yes	Unfit
6	146	M	Yes	Fit
7	168	F	No	Unfit
8	231	F	Yes	Fit
9	150	M	No	Fit
10	190	F	No	Fit

Variable_dropped	Accuracy
Plays_Sport?	88%

ID	Calories_burnt	Gender	Plays_Sport?	Fitness_Level
1	121	M	Yes	Fit
2	230	M	No	Fit
3	342	F	No	Unfit
4	70	M	Yes	Fit
5	278	F	Yes	Unfit
6	146	M	Yes	Fit
7	168	F	No	Unfit
8	231	F	Yes	Fit
9	150	M	No	Fit
10	190	F	No	Fit

11/21/2023 48



- If you see gender has produced the smallest change in the performance in the model first, it was 92% when we took all the variables and when we dropped gender, it was 91.6%. So we can infer that gender does not have a high impact on the Fitness_Level variable. And hence it can be dropped.
- Finally, we will repeat all these steps until no more variables can be dropped.
- It's a very simple, but very effective technique.

Accuracy using all the variables = 92%

Variable_dropped	Accuracy
Calories_burnt	90%
Gender	91.60%
Plays_Sport?	88%

Embedded Methods:

- Embedded methods learn about the features that contribute the most to the model's performance while the model is being created. You have seen Feature Selection methods in the previous lessons, and we will discuss several more in future lessons, like Decision Tree based methods.
- Ridge Regression (L2-Regularization)
- Lasso Regression (L1-Regularization)
- Elastic-Net Regression (uses both L1 and L2 Regularization)
- Decision Tree-Based Methods (Decision Tree Classification, Random Forest Classification, XgBoost Classification, LightGBM)