**AI Exam Process Steps with Functions:**

A close up of a device

Description automatically generated

1. **Initial steps**
   1. Load your dataset
   2. Examine it and get rid of anything that doesn’t apply
      1. *examine\_df(df)*
      2. *distribution\_plot(df,column\_name)*
      3. *joint\_plot(df,x,y)*
      4. *pair\_plot(df)*
   3. Split the dataset into the input and output variables
      1. *separate\_components(df, column\_of\_y)*
2. **Pre-processing**

Can be either done using first fit() then transform() function or fit\_transform()

* 1. Rescale data(Normalization)

Important for optimization algorithms that use gradient descent and regressions that weigh inputs(regression/neural networks or with clustering – k-means)

* + 1. *rescale (X)*
  1. Standardize data

Assumes Gaussian distribution but different means and standard deviation. 0 means mean and 1 means standard distribution. Most relevant for linear, log regreassion and LDA(linear discriminant analysis)

* + 1. *standardize(X)*
  1. Normalize data

Normalize by row so that each row length 1. Useful for algorithms that weigh input values as a whole such as Neural Networks and distance algorithms(K-nn)

* + 1. *normalize(X)*
  1. Binarize data

Transform data based on binary threshold. Used when adding a new feature often.

* + 1. *binarize(X)*
  1. Label Encoding

Changing name of column

* + 1. *encode(df, name\_of\_column, new\_name)*
  1. Dummy encoding

Create dummy variables for each class to avoid order when in reality only encoding

* + 1. *get\_dummies(df, column\_name)*

1. **Feature Selection**

Quality of model determined by data features. Select those features that contribute most to the variable of interest. It reduces overfitting through reducing redundancy. Improves accuracy by creating a more accurate model. Reduces time needed to train by excluding irrelevant features.

* 1. **Univariate Selection**

Uses statistical test mostly with KBest to determine most relevant attributes using statistical test

* + 1. *univariate\_chi(x, y, df, target\_var, k=4(default))*
  1. **Recursive feature elimination**

Recursively removes attributes and builds models with remaining. Model accuracy as guidance on which combination of attributes contributes the most.

* + 1. *recursive\_elimination(x, y, df, target\_var, k=3(default))*
  1. **PCA – Principal Component Analysis**

Data reduction method with linear algebra through compressing several dimensions into principal components. Cannot explain them once comporessed.

Method combines highly correlated variables together to form smaller number of an artificial set of variables called “principal components” that account for most variance in the data.

* + 1. *pca(x, k=3(default))*
  1. **Tree Classifier**

Can be used to estimate importance of feature to refine the model and to make decision beteen explainability and accuracy. Many trees with variations to get accurate result

* + 1. *Extra\_trees(x,y,df,target\_var, estimators=100(default))*

1. **Model Evaluation**

Two reasons for evaluation:

* + - 1. Decide which interventions are effective
      2. Get concrete knowledge on how accurate they are and to what extent we can trust them

We cannot use data we used for training the model to assess it. Two approaches to mitigate it – 1) use Train-Test Split works well with a lot of data 2) sampling techniques to augment the amount of data available

1. **Train and test set**

Split dataset into 70-30 or 80-20 split. Simple, fast and good for large datasets. We have to ensure variance is similar. If not, might not be good predictor.

* 1. *test\_split(x,y,test\_size, seed=7(default))*

1. **K-fold Cross-validation**

Used to reduce variance of using test-train-split

1. **Leave one-out cross-validation**
2. **Repeated random test-train splits**