***Action Plan  
 Title:*** Automated Feature Extraction Using Featuretools Library.  
 ***Result:*** Automatically Extracted Meaningful Features.   
 ***Materials Needed:*** Python.  
 ***Position with Overall Accountability:***Data Science Manager.  
 ***Reporting Positions:*** Associate Data Scientist, Data Scientists, and Senior Data Scientist.  
 ***Staff Positions:*** None.

The purpose of the Feature Tools library is to automate the creation of meaningful features from raw data, saving time and effort for data scientists.

**Deep Feature Synthesis (DFS) :**

DFS stands for Deep Feature Synthesis, which is a key component of the Feature Tools library. It is an automated method that leverages the relationships between different data tables to generate new features by aggregating, transforming, and combining existing features, enabling efficient and comprehensive feature engineering.

1. **Creating an entity set and identifier assignment:** This code snippet creates an Entity Set using the Feature Tools library and assigns the identifier (‘identifier name’), in this case, 'iris' to it. An Entity Set is a data structure that holds multiple related data tables (entities) and their relationships, allowing for efficient feature engineering and analysis.

Click the link given below for more details.

<https://github.com/NithaDivakaran/SOP_Data-Contingency/blob/main/Checking%20for%20missing%20values.ipynb>

1. **Adding a Data Frame as an Entity to an Entity Set:** The data frame on which we must perform automated feature engineering is added to the above created Entity Set with any entity name, in this case, ‘data’. The index parameter specifies the column in the data frame which should be used as the index for the entity, which helps in identifying and linking the data correctly with the Entity Set.

Click the link given below for more details.

<https://github.com/NithaDivakaran/SOP_Data-Contingency/blob/main/Handling%20missing%20values%20using%20Pandas.ipynb>

1. **Generating new features using deep feature synthesis (DFS):** This code snippet performs deep feature synthesis using transformation primitives.
   1. ft.dfs: The function used to run the deep feature synthesis.
   2. entityset: The Entity Set Object.
   3. Target\_dataframe\_name : Specifies the name of the target data frame in the Entity Set, which is 'data' in this case.
   4. agg\_primitives : is a list of aggregation primitives such as "mean", "max", "min", "std", and "skew" that will be applied to the data during feature synthesis.
   5. trans\_primitives : is a list of transformation primitives, with 'less\_than\_equal\_to' being one of them, which will be applied to the data during feature synthesis.

Click the link given below for more details.

<https://github.com/NithaDivakaran/SOP_Data-Contingency/blob/main/Handling%20missing%20values%20using%20scikit-learn.ipynb>

**Primitives:**

1. **Aggregate Primitives:**

These are functions or operations that take a group of related data instances and produce a single value that summarizes or combines the information from those instances. They help in generating summary statistics or aggregating data across different groups or categories.

Click the link given below for more details.

<https://github.com/NithaDivakaran/SOP_Data-Contingency/blob/main/Handling%20missing%20values%20using%20scikit-learn.ipynb>

1. **Transform Primitives:**

These are functions or operations that work on individual data columns or features to extract or manipulate information and create new features. They help in transforming or deriving new insights from existing data by applying various calculations, conversions, or manipulations on the data values within a column.

Click the link given below for more details.

<https://github.com/NithaDivakaran/SOP_Data-Contingency/blob/main/Handling%20missing%20values%20using%20scikit-learn.ipynb>