Week 9: Deliverables

Group Name: Medical Data Science

Name: Peter Abban

Email: abbanpeter12@gmail.com

Country: Hungary

Specialization: Data Science

- **Problem Description**

A critical challenge for pharmaceutical companies is **understanding and monitoring drug persistency**, which reflects the extent to which patients adhere to prescribed medication schedules over time. Poor adherence compromises treatment efficacy, increases healthcare costs, and negatively impacts patient outcomes.

To address this challenge, **ABC Pharmaceutical Company engaged an analytics firm** to develop an automated system based on some feature variables of the patients

By leveraging advanced analytics, the solution aims to enhance **patient care, support physicians in monitoring compliance, and inform strategic decisions for the company's pharmaceutical portfolio** by making predictions on whether the patient is persistent or non-persistent

- **Dataset type for analysis:** Healthcare dataset – Pharmaceutical

- **Missing Values:**
  Several feature variables contain missing observations, with varying degrees of impact:

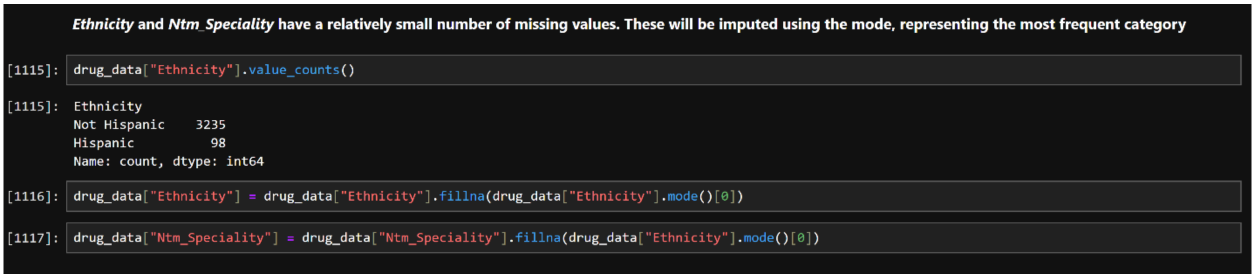
  - **Ethnicity**: 91 missing values
  - **Ntm_Speciality**: 310 missing values
  - **Change_T_Score**: 1,497 missing values
  - **Tscore_Bucket_During_Rx**: 1,497 missing values
  - **Risk_Segment_During_Rx**: 1,497 missing values
  - **Change_Risk_Segment**: 2,220 missing values

❖ Handling of Missing Values

Missing values in the dataset will be addressed based on the **type and proportion of missingness** for each feature:

- **Categorical variables**:
  - *Ethnicity* and *Ntm_Speciality* have a relatively small number of missing values. These will be imputed using the **mode**, representing the most frequent category, which preserves the categorical distribution without introducing bias.

- **Code for Handling Ethnicity and Ntm_Speciality**

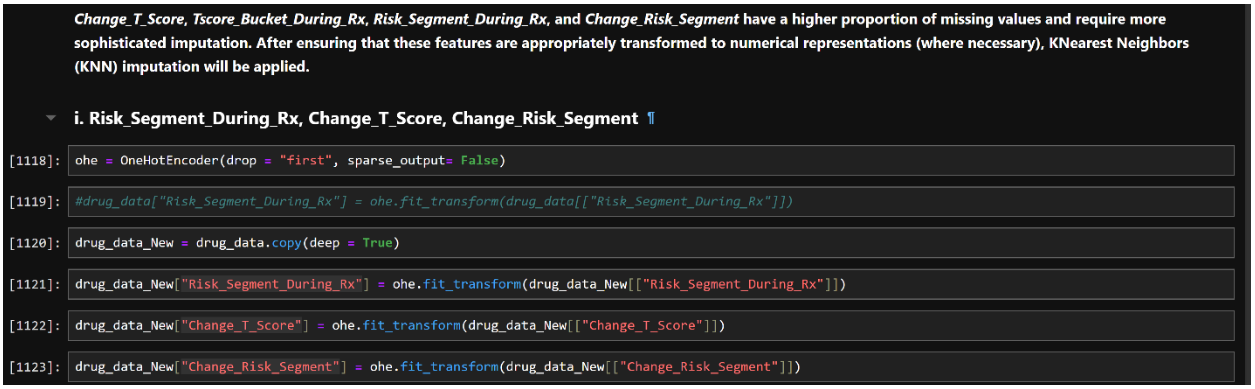


***Ethnicity* and *Ntm_Speciality* have a relatively small number of missing values. These will be imputed using the mode, representing the most frequent category**

```
[1115]: drug_data["Ethnicity"].value_counts()

[1115]: Ethnicity
        Not Hispanic    3235
        Hispanic          98
        Name: count, dtype: int64

[1116]: drug_data["Ethnicity"] = drug_data["Ethnicity"].fillna(drug_data["Ethnicity"].mode()[0])

[1117]: drug_data["Ntm_Speciality"] = drug_data["Ntm_Speciality"].fillna(drug_data["Ethnicity"].mode()[0])
```

- **Numerical/clinical variables**:

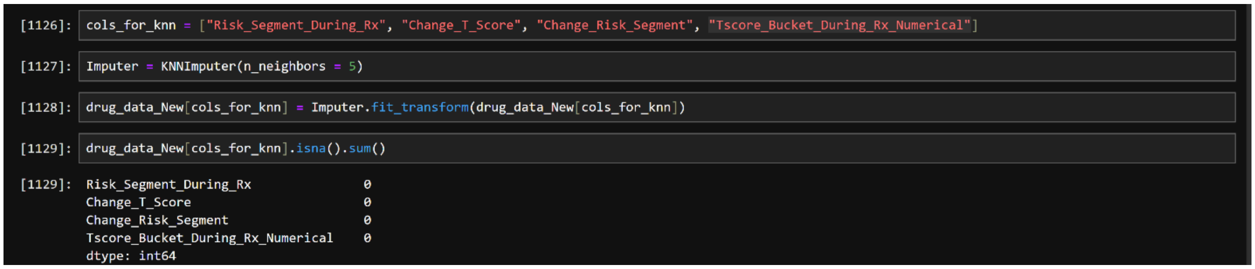
  Change_T_Score, Tscore_Bucket_During_Rx, Risk_Segment_During_Rx, and *Change_Risk_Segment* have a higher proportion of missing values and require more sophisticated imputation. After ensuring that these features are appropriately transformed to numerical representations (where necessary), **K-Nearest Neighbors (KNN) imputation** will be applied.



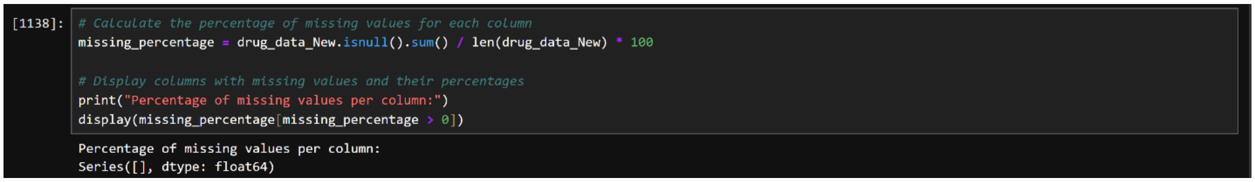
***Change_T_Score, Tscore_Bucket_During_Rx, Risk_Segment_During_Rx,* and *Change_Risk_Segment* have a higher proportion of missing values and require more sophisticated imputation. After ensuring that these features are appropriately transformed to numerical representations (where necessary), KNearest Neighbors (KNN) imputation will be applied.**

i. Risk_Segment_During_Rx, Change_T_Score, Change_Risk_Segment ¶

```
[1118]: ohe = OneHotEncoder(drop = "first", sparse_output= False)

[1119]: #drug_data["Risk_Segment_During_Rx"] = ohe.fit_transform(drug_data[["Risk_Segment_During_Rx"]])

[1120]: drug_data_New = drug_data.copy(deep = True)

[1121]: drug_data_New["Risk_Segment_During_Rx"] = ohe.fit_transform(drug_data_New[["Risk_Segment_During_Rx"]])

[1122]: drug_data_New["Change_T_Score"] = ohe.fit_transform(drug_data_New[["Change_T_Score"]])

[1123]: drug_data_New["Change_Risk_Segment"] = ohe.fit_transform(drug_data_New[["Change_Risk_Segment"]])
```



```
[1126]: cols_for_knn = ["Risk_Segment_During_Rx", "Change_T_Score", "Change_Risk_Segment", "Tscore_Bucket_During_Rx_Numerical"]

[1127]: Imputer = KNNImputer(n_neighbors = 5)

[1128]: drug_data_New[cols_for_knn] = Imputer.fit_transform(drug_data_New[cols_for_knn])

[1129]: drug_data_New[cols_for_knn].isna().sum()

[1129]: Risk_Segment_During_Rx              0
        Change_T_Score                      0
        Change_Risk_Segment                 0
        Tscore_Bucket_During_Rx_Numerical   0
        dtype: int64
```



```
[1138]: # Calculate the percentage of missing values for each column
        missing_percentage = drug_data_New.isnull().sum() / len(drug_data_New) * 100

        # Display columns with missing values and their percentages
        print("Percentage of missing values per column:")
        display(missing_percentage[missing_percentage > 0])

        Percentage of missing values per column:
        Series([], dtype: float64)
```

NB: **All missing values have been successfully handled using the respective imputation methods**