Data Processing, Visualization & ML on Real-World Dataset

1. **Visualization:**
   - Please rerun the Jupyter Notebook I uploaded (Chapter 2). Apply slight modifications to the parameters and understand different visualization techniques.
2. **Dataset Exploration:**
   - **Dataset**: Load **Breast cancer wisconsin** dataset (classification task).
   - **Task**: Summarize the structure of the dataset, including the number of rows, columns, and data types. Identify the target variable and provide a brief description of the dataset's purpose.
3. **One-Hot Encoding and Transformation:**
   - **Encoding**: Perform one-hot encoding on the categorical features of your dataset.
   - **Transformation:** After encoding, apply a transformation to address skewed numerical features. Explain how the transformation improves data distribution.
4. **Test on the Initial Data:**
   a. **Model Training:** Use the given dataset, split it, and train any machine learning model (for e.g SVM).
   b. **Evaluation:** Evaluate the model on the test data and check its performance .
5. **Handling Missing Data:**
   - **Missing Values:** Identify any missing values in your dataset (I have provided an example of missing data).
   - **Method:** Apply two different methods to handle the missing data (e.g., mean/mode imputation or removal). Discuss how each method affects the dataset.
6. **Outlier Detection and Removal:**
   - **Method:** Use the *IsolationForest* method from the scikit-learn library.
7. **Feature Scaling:**
   - **Method:** You can use the min-max or standardization function as per your requirements.
8. **Data Splitting:**
   - **Task:** Split your dataset into training and testing sets. If needed, create a validation set. Discuss how the size of each split impacts model performance.
9. **Test on Pre-processed Data (to observe improvement):**
   - **Method:** Now, use the same machine learning model (SVM) again as specified above.
   - **Evaluation:** evaluate performance metrics and observe the change in performance.
10. **Data Visualization:**
    - **Task:** Create 5 different types of plots (e.g., bar chart, scatter plot, heat map, histogram, violin plot) using your dataset. Explain how each plot helps understand different aspects of the dataset.
11. **Dimensionality Reduction technique:**
    - **t-SNE:** Visualize the 2D scatter plot and interpret any clusters. What insights do the clusters provide regarding the classification problem?
    - **PCA:** Plot the explained variance for each component and visualize the first two principal components.
    - **ICA:** Visualize the independent components and explain how ICA extracts statistically independent features that other methods like PCA may not capture.
    - **SVD:** Apply Singular Value Decomposition (SVD) and visualize the singular values. Discuss how SVD aids in reducing dimensionality while preserving relevant information.
12. Try the ROC-AUC plot (optional).

Helpful code for question 4 & 9: SVM model training, testing, and performance evaluation

```
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, confusion_matrix, recall_score, precision_score
## Split the dataset
train_df, test_df = train_test_split(df.dropna(), test_size=0.4, random_state=42)
```

Training

```
model = SVC()
feats_cols = list(range(2, 32, 1))
x_train = train_df[feats_cols].values
y_train = train_df[[1]].values
model.fit(x_train, y_train)
```

Testing

```
feats_cols = list(range(2, 32, 1))
x_test = test_df[feats_cols].values
y_test = test_df[[1]].values
```

Evaluation

```
y_pred = model.predict(x_test)
accuracy = accuracy_score(y_test, y_pred)
accuracy = accuracy_score(y_test, y_pred)
print(f"Model Accuracy: {accuracy * 100:.2f}%")
recall = recall_score(y_test, y_pred)
print(f"Model Recall: {recall * 100:.2f}%")
precision = precision_score(y_test, y_pred)
print(f"Model Precision: {precision * 100:.2f}%")
```