Name: Durga Prasad Sahoo\ **Roll No:** 22CSE304

Machine Learning Workflow with 3 Datasets

Step 1: Identifying Missing Values

- Use `isnull()` and `sum()` in pandas to detect missing values.
- Handle missing values via:
- **Deletion**: Drop rows or columns (`dropna()`)
- **Imputation**: Fill with mean, median, mode or use ML methods (e.g., KNN Imputer)

```
import pandas as pd

missing_values = df.isnull().sum()
print(missing_values)
df.fillna(df.mean(), inplace=True)  # or use advanced imputers
```

Step 2: Data Cleaning

- Remove duplicates: `df.drop_duplicates()`
- Standardize values (e.g., lowercase, no whitespace)
- Convert data types: `pd.to_datetime()`, `astype()`
- Encode categorical variables: `pd.get_dummies()`, `LabelEncoder`

Step 3: Mutation Techniques

- Feature transformations:
- Log, square root, power transforms
- Binning continuous variables
- One-hot encoding or label encoding for categoricals
- Normalization or standardization (MinMaxScaler, StandardScaler)

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
df[['feature1', 'feature2']] = scaler.fit_transform(df[['feature1',
'feature2']])
```

Step 4: Identifying Outliers

- Use:
- Z-score: `scipy.stats.zscore`

- IQR method
- Boxplot, scatter plot

```
from scipy import stats
import numpy as np
z = np.abs(stats.zscore(df))
df = df[(z < 3).all(axis=1)]
```

Step 5: Creating 5 New Variables (Features)

Examples:

- Interaction terms: `feature1 * feature2`
- Polynomial features
- Aggregates like `total_spent`, `days_since_join`
- Ratios: `amount_per_visit = total_amount / visit_count`

Step 6: New KPIs

- KPIs are key metrics that reflect business goals:
- Conversion Rate
- Average Revenue per User (ARPU)
- Churn Rate
- Fraud Detection Accuracy
- Model Precision for High-Risk Users

Step 7: Classification Models (At Least 10)

1. Logistic Regression
2. Decision Tree Classifier
3. Random Forest
4. Support Vector Machine (SVM)
5. K-Nearest Neighbors (KNN)
6. Naive Bayes
7. Gradient Boosting
8. XGBoost
9. LightGBM
10. Multi-layer Perceptron (Neural Net)

Step 8: Evaluation Metrics

Evaluate with:

- Accuracy
- Confusion Matrix
- ROC-AUC
- Precision
- Recall
- F1-Score
- Log Loss

```
from sklearn.metrics import accuracy_score, confusion_matrix, roc_auc_score,
f1_score, recall_score
```

Step 9: Metrics Calculation

```
from sklearn.metrics import classification_report

y_pred = model.predict(X_test)
print(classification_report(y_test, y_pred))
```

Step 10: Model Creation, Training & Testing (with 3 Datasets)

Repeat for Dataset 1, 2, 3:

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier

# Load dataset
df = pd.read_csv('dataset1.csv')

# Preprocessing steps here...

# Split
y = df['target']
X = df.drop('target', axis=1)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

# Train
model = RandomForestClassifier()
model.fit(X_train, y_train)

# Predict and Evaluate
```

```
y_pred = model.predict(X_test)
print(classification_report(y_test, y_pred))
```

This document outlines the **complete ML workflow**, emphasizing data preparation, feature engineering, modeling, and evaluation using **three different datasets**.

Ready for submission by: **Durga Prasad Sahoo | 22CSE304**