This notebook walks you through a complete machine learning project:

1. **Define the problem**: Predict sleep apnea.
2. **Prepare data**: Load, explore, and preprocess it.
3. **Build models**: Train multiple algorithms.
4. **Evaluate**: Check accuracy and errors.
5. **Interpret**: Understand what drives predictions.

Each step builds on the last, showing how data becomes insights. As a student, you can use this structure for any machine learning task—adjust the features, models, or metrics to fit your project!

Step 1: Understanding the Project Purpose

Step 1. The Purpose of the Project 🔐

The purpose remains largely unchanged but is updated to reflect the new variables:

⭕ ****Problem****: Sleep apnea is a common sleep disorder affecting many individuals, disrupting sleep quality and overall health. Identifying those at risk using accessible data is crucial for timely intervention.

⭕ ****Solution****: This project uses machine learning to predict sleep apnea based on 23 features (plus the target variable `Sleep Apnea`) collected from individuals, such as gender, age group, sleep duration, and lifestyle factors. Models are trained and evaluated to classify individuals as having sleep apnea or not.

Explanation:

- **What it is**: This section introduces the project. It starts with a simple title and author credit, then provides an updated purpose statement. The problem is defined as identifying sleep apnea (a condition where breathing stops temporarily during sleep) because it impacts health. The solution involves using machine learning to analyze data and predict whether someone has sleep apnea.
- **Why it matters**: Understanding the goal helps you know what you're working toward. Here, the aim is to classify people into "Yes" or "No" for sleep apnea based on various features (data points about individuals). This is a binary classification task, a common machine learning problem.

Step 2: Cloning the Repository

Explanation:

- **What it does**: This command uses Git (a version control tool) to download a repository from GitHub to your working environment (e.g., Google Colab). The repository likely contains the dataset or other files needed for the project.
- **Why it's important**: It ensures you have access to the data and resources. In a classroom setting, think of this as getting the textbook or materials you need before starting the assignment.

Step 3: Data Features

2. Data Features 🌐

The dataset now consists of the following 24 variables (23 features + 1 target). '

****Here's a description of each:****

⭕ What is your occupation? (Occupation): Categorical, e.g., Doctor, Engineer, etc.

⭕ On average, how many hours do you sleep per night? (Sleep_*Duration): Numerical, hours slept.*

⭕ *How would you rate the quality of your sleep on a scale of 1 to 10? (Sleep*_Quality): Numerical (1-10), ordinal.

⭕ How would you describe your physical activity level? (Physical_*Activity*_Level): Categorical, e.g., Low, Medium, High.

⭕ How would you rate your stress level on a scale of 1 to 10? (Stress_*Level): Numerical (1-10), ordinal.*

⭕ *What is your BMI category? (BMI*_Category): Categorical, e.g., Normal, Overweight, Obese.

⭕ What is your blood pressure category? (Blood_*Pressure*_Category): Categorical, e.g., Normal, High.

⭕ What is your resting heart rate? (Resting_*Heart*_Rate): Numerical, beats per minute.

⭕ On average, how many steps do you take per day? (Daily_*Steps): Numerical, steps.*

⭕ *If you have sleep apnea, what is the mean duration of apnea-hypopnea events? (Mean*_Apnea_*Duration): Numerical, seconds (0 if no apnea).*

⭕ *Do you have either of the sleep disorders below? (Sleep*_Disorders): Categorical, Yes/No (assumed to exclude sleep apnea).

⭕ How many cigarettes do you smoke per day? (Cigarettes_*Per*_Day): Numerical, cigarettes.

⭕ What is your neck thickness? (Neck_*Thickness): Numerical, cm.*

⭕ *How would you describe your tongue size? (Tongue*_Size): Categorical, e.g., Small, Medium, Large.

- Do you use any muscle-relaxing substances? (Muscle_*Relaxing*_Substances): Categorical, Yes/No.
- What is your smoking habit? (Smoking_*Habit): Categorical, e.g., Never, Current, Former.*
- *Is there a family history of sleep apnea? (Family_*History_*Sleep*_Apnea): Categorical, Yes/No.
- Have you had a stroke in the past? (Past_*Stroke): Categorical, Yes/No.*
- *What is your status regarding type 2 diabetes? (Type_*2_*Diabetes): Categorical, Yes/No.*
- *What is your status regarding type 1 Diabetes? (Type_*1_*Diabetes): Categorical, Yes/No.*
- *Subject Occupation: Ignored as it seems redundant with "What is your occupation?" (assumed typo).*
- *Sleep Apnea: Target variable, Categorical, Yes/No.*

Note: The variable Subject Occupation is excluded as it appears to duplicate Occupation, reducing the feature count to 22 plus the target.*

Explanation:

- **What it is**: This section lists the 22 features (input variables) and 1 target variable (Sleep_Apnea) in the dataset. Features are either numerical (e.g., Sleep_Duration in hours) or categorical (e.g., BMI_Category as Normal/Overweight). The target is what we want to predict: whether someone has sleep apnea (Yes/No).
- **Why it matters**: Features are the building blocks of prediction. Knowing what each represents helps you understand what influences sleep apnea (e.g., lifestyle factors like smoking or physical traits like neck thickness). The note about excluding Subject Occupation shows attention to data quality by removing duplicates.

Step 4: Importing Libraries

Explanation:

- **What it does**: This imports Python libraries (tools) needed for the project:
    - pandas and numpy: Handle and manipulate data.
    - seaborn and matplotlib: Create visualizations.
    - sklearn: Provides machine learning tools (splitting data, scaling, encoding, models, metrics).
    - xgboost: An advanced machine learning algorithm.
    - shap: Explains model predictions.
- **Why it's important**: These libraries are like your toolbox. Each serves a purpose, from loading data to building and evaluating models. Without them, you'd have to write everything from scratch!

Step 5: Reading Data

Explanation:

- **What it does**: This reads a CSV file into a pandas DataFrame (think of it as a table). The file Sara_Random_Final-CHATGPT.csv contains the data with the 22 features and target.
- **Why it's important**: Loading data is the starting point. Without it, there's nothing to analyze or model. The comment suggests you might need to adjust the file path depending on where it's stored.

Step 6: Statistical Information

Explanation:

- **What it does**:
    - df.shape: Shows the number of rows (people) and columns (features + target).
    - df.isnull().sum(): Counts missing values in each column.
    - df.describe(): Gives stats (mean, min, max, etc.) for numerical columns.
- **Why it's important**: This is your first look at the data's structure and quality. For example, if there are many missing values, you'll need to address them. Stats help you spot oddities (e.g., a negative heart rate would be a red flag).

Step 7: Exploratory Data Analysis (EDA)

```
# Column names (simplified for coding)
columns = [
    'Subject Occupation', 'Sleep_Duration', 'Sleep_Quality',
    'Physical_Activity_Level', 'Stress_Level', 'BMI_Category', 'Blood_Pressure_Category',
    'Resting_Heart_Rate', 'Daily_Steps', 'Mean_Apnea_Duration', 'Sleep_Disorders',
    'Cigarettes_Per_Day', 'Neck_Thickness', 'Tongue_Size', 'Muscle_Relaxing_Substances',
    'Smoking_Habit', 'Family_History_Sleep_Apnea', 'Past_Stroke', 'Type_2_Diabetes',
    'Type_1_Diabetes', 'Sleep_Apnea'
]

# Numerical and categorical columns
numerical_cols = ['Sleep_Duration', 'Sleep_Quality', 'Stress_Level', 'Resting_Heart_Rate',
                  'Daily_Steps', 'Cigarettes_Per_Day', 'Neck_Thickness', 'Mean_Apnea_Duration']
categorical_cols = [col for col in columns if col not in numerical_cols and col != 'Sleep_Apnea']
```

```
# Histograms for numerical variables
df[numerical_cols].hist(figsize=(12, 10))
plt.tight_layout()
plt.show()

# Bar plot for target variable
sns.countplot(x='Sleep_Apnea', data=df)
plt.title("Distribution of Sleep Apnea")
plt.show()

# Correlation matrix for numerical variables
corr = df[numerical_cols].corr()
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.title("Correlation Matrix")
plt.show()
```

Explanation:

- **What it does**:
    - **Column Lists**: Defines all columns, then splits them into numerical (numbers) and categorical (categories) for different analyses.
    - **Histograms**: Plots the distribution of numerical features (e.g., how many people sleep 6-7 hours?).
    - **Count Plot**: Shows how many people have sleep apnea (Yes) vs. don't (No).
    - **Correlation Matrix**: A heatmap showing how numerical features relate (e.g., does high stress correlate with poor sleep quality?).
- **Why it's important**: EDA helps you "see" the data. Histograms reveal patterns or outliers (e.g., if most people sleep 8 hours, but some report 20, that's odd). The count plot checks if the target is balanced (equal Yes/No) or imbalanced. Correlations highlight relationships that might affect predictions.

Step 8: Data Preprocessing

```
# Handle missing values
df['Mean_Apnea_Duration'].fillna(0, inplace=True)  # 0 for no apnea
```

```
for col in df.columns:
    if col in numerical_cols and col != 'Mean_Apnea_Duration':
        df[col].fillna(df[col].median(), inplace=True)
    elif col in categorical_cols:
        df[col].fillna(df[col].mode()[0], inplace=True)

# Encode categorical variables
df_encoded = pd.get_dummies(df, columns=categorical_cols, drop_first=True)

# Encode target variable
le = LabelEncoder()
df_encoded['Sleep_Apnea'] = le.fit_transform(df['Sleep_Apnea'])

# Split features and target
X = df_encoded.drop('Sleep_Apnea', axis=1)
y = df_encoded['Sleep_Apnea']

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Scale numerical features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Explanation:

- **What it does**:
    - **Missing Values**: Fills Mean_Apnea_Duration with 0 (assuming no apnea), other numerical columns with their median (middle value), and categorical columns with their mode (most common value).
    - **Encoding**:

        - Categorical features are turned into numbers using one-hot encoding (e.g., BMI_Category becomes columns like BMI_Overweight, BMI_Obese with 0s and 1s).
        - Sleep_Apnea is encoded as 0 (No) or 1 (Yes).
    - **Splitting**: Separates features (X) and target (y), then splits them into training (80%) and testing (20%) sets.
    - **Scaling**: Adjusts numerical features to have a mean of 0 and standard deviation of 1.
- **Why it's important**: Machines can't handle missing data or text directly, so preprocessing prepares the data. Encoding makes categories usable, splitting prevents overfitting (testing on unseen data), and scaling ensures all features contribute equally to the model (e.g., Daily_Steps in thousands doesn't overpower Sleep_Quality from 1-10).

Step 9: Data Modeling

 # Initialize models

rf = RandomForestClassifier(random_state=42)

lr = LogisticRegression(random_state=42, max_iter=1000)

svm = SVC(random_state=42)

xgb = XGBClassifier(random_state=42)

 # Train models

rf.fit(X_train_scaled, y_train)

lr.fit(X_train_scaled, y_train)

svm.fit(X_train_scaled, y_train)

xgb.fit(X_train_scaled, y_train)

Explanation:

- **What it does**:
    - Initializes four machine learning models: Random Forest (tree-based), Logistic Regression (linear), SVM (boundary-based), and XGBoost (boosted trees).
    - Trains each model on the scaled training data.
- **Why it's important**: This is where the "learning" happens. Each model learns patterns in the training data to predict sleep apnea. Using multiple models lets you compare their performance.

Step 10: Models Evaluation

```
# Predictions
rf_pred = rf.predict(X_test_scaled)
lr_pred = lr.predict(X_test_scaled)
svm_pred = svm.predict(X_test_scaled)
xgb_pred = xgb.predict(X_test_scaled)

# Accuracy
print("Random Forest Accuracy:", accuracy_score(y_test, rf_pred))
print("Logistic Regression Accuracy:", accuracy_score(y_test, lr_pred))
print("SVM Accuracy:", accuracy_score(y_test, svm_pred))
print("XGBoost Accuracy:", accuracy_score(y_test, xgb_pred))

# Confusion Matrix for XGBoost
cm = confusion_matrix(y_test, xgb_pred)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.title("XGBoost Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

Explanation:

- **What it does**:
    - Makes predictions on the test set for each model.
    - Prints the accuracy (fraction of correct predictions).
    - Plots a confusion matrix for XGBoost, showing true positives (Yes predicted Yes), true negatives (No predicted No), false positives, and false negatives.
- **Why it's important**: Evaluation tells you how well each model performs on unseen data. Accuracy is a simple metric, but the confusion matrix gives deeper insight (e.g., is the model better at spotting Yes or No cases?).

Step 11: Interpretation of One Model

```
# SHAP interpretation for XGBoost
```

explainer = shap.TreeExplainer(xgb)

shap_values = explainer.shap_values(X_test_scaled)

shap.summary_plot(shap_values, X_test_scaled, feature_names=X.columns)

Explanation:

- **What it does**: Uses SHAP (SHapley Additive exPlanations) to explain the XGBoost model's predictions. It calculates how much each feature contributes to each prediction and plots a summary showing feature importance and impact.
- **Why it's important**: Models like XGBoost can be "black boxes." SHAP opens the box, showing which features (e.g., Neck_Thickness or Stress_Level) drive predictions and how (e.g., higher values increase risk). This builds trust and understanding.

Step 12: Key Changes and Notes

Key Changes and Notes 🔐

⭕ **Dataset:** The original dataset (enhanced_*full*_sleep_*apnea*_dataset.csv) is replaced with a new CSV containing only the specified 24 variables (adjusted to 22 features + target due to redundancy).

⭕ **Features:** Subject Occupation is excluded as it duplicates Occupation. Mean_*Apnea*_Duration is included despite potential data leakage concerns, with missing values filled as 0.

⭕ **Preprocessing:** Categorical variables are one-hot encoded, numerical variables are scaled, and the target is label-encoded (assuming binary: Yes/No).

⭕ **EDA:** Updated to reflect the new variables with appropriate visualizations.

⭕ **Modeling:** Retained the original models (Random Forest, Logistic Regression, SVM, XGBoost) for consistency.

Explanation:

- **What it is**: Summarizes updates from an older version of the project. It notes the new dataset, feature adjustments, and preprocessing details.
- **Why it matters**: Highlights decisions (e.g., removing redundant features) and potential issues (e.g., Mean_Apnea_Duration might leak info about the target, but it's kept as specified). This teaches you to document choices and consider their implications.