1 Example 1

```
import numpy as np
import matplotlib.pyplot as plt

# create time series
np.random.seed(0)
data = np.random.randn(1000).cumsum()

# plot
plt.figure(figsize=(12, 6))
plt.plot(data)
plt.title(’Random Time Series (1000 points)’)
plt.xlabel(’Time’)
plt.ylabel(’Value’)
plt.grid(True)
plt.show()
```

```
import xgboost as xgb
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

def load_data(file_path):
    data = pd.read_csv(file_path)
    return data

def preprocess_data(data):
    X = data.drop([’target’], axis=1)
    y = data[’target’]
    return X, y

def train_model(X_train, y_train):
    model = xgb.XGBClassifier()
    model.fit(X_train, y_train)
    return model

def evaluate_model(model, X_test, y_test):
    y_pred = model.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    return accuracy
```

```
def main():
    file_path = 'credit_score_data.csv'
    data = load_data(file_path)
    X, y = preprocess_data(data)
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state
    model = train_model(X_train, y_train)
    accuracy = evaluate_model(model, X_test, y_test)
    print(f'Accuracy: {accuracy:.3f}')

if __name__ == '__main__':
    main()
```

2 Example 2

```
import argparse
import pandas as pd
import xgboost as xgb
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

def load_data(file_path):
    data = pd.read_excel(file_path)
    return data

def preprocess_data(data):
    X = data.drop(['target'], axis=1)
    y = data['target']
    return X, y

def train_model(X_train, y_train):
    model = xgb.XGBClassifier()
    model.fit(X_train, y_train)
    return model

def evaluate_model(model, X_test, y_test):
    y_pred = model.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    return accuracy

def main():
    parser = argparse.ArgumentParser(description='Train XGBoost model')
    parser.add_argument('-f', '--file', help='Path to Excel file', required=True)
    args = parser.parse_args()
    file_path = args.file
    data = load_data(file_path)
    X, y = preprocess_data(data)
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_stat
    model = train_model(X_train, y_train)
    accuracy = evaluate_model(model, X_test, y_test)
    print(f'Accuracy: {accuracy:.3f}')

if __name__ == '__main__':
    main()
```

```
python program.py -f path/to/file.xlsx
```

```
python program.py --file path/to/file.xlsx
```