```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt

# Load and preprocess data
data = pd.read_csv('EX4A.csv')
data['days_per_week'] = data['days_per_week'].apply(lambda x: len(x.split(',')))

# Features and target
attributes = data.drop('avg_time_in_gym', axis=1)
target = data['avg_time_in_gym']

# Column types
cat_features = ['gender', 'abonoment_type', 'attend_group_lesson', 'fav_group_lesson',
                'fav_drink', 'personal_training', 'name_personal_trainer', 'uses_sauna']
num_features = ['Age', 'visit_per_week', 'days_per_week']

# Preprocessing
preprocess = ColumnTransformer([
    ('num', StandardScaler(), num_features),
    ('cat', OneHotEncoder(), cat_features)
])
attributes = preprocess.fit_transform(attributes)

# Train-test split
```

```
attributes_train, attributes_test, target_train, target_test = train_test_split(
    attributes, target, test_size=0.2, random_state=42)

# Model training and prediction
model = LinearRegression()
model.fit(attributes_train, target_train)
target_pred = model.predict(attributes_test)

# Evaluation
print("Mean Squared Error:", mean_squared_error(target_test, target_pred))
print("R² Score:", r2_score(target_test, target_pred))

# Visualization
plt.scatter(target_test, target_pred, color='blue', alpha=0.6)
plt.plot([target_test.min(), target_test.max()], [target_test.min(), target_test.max()], 'r--')
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.title('Actual vs Predicted')
plt.show()
```