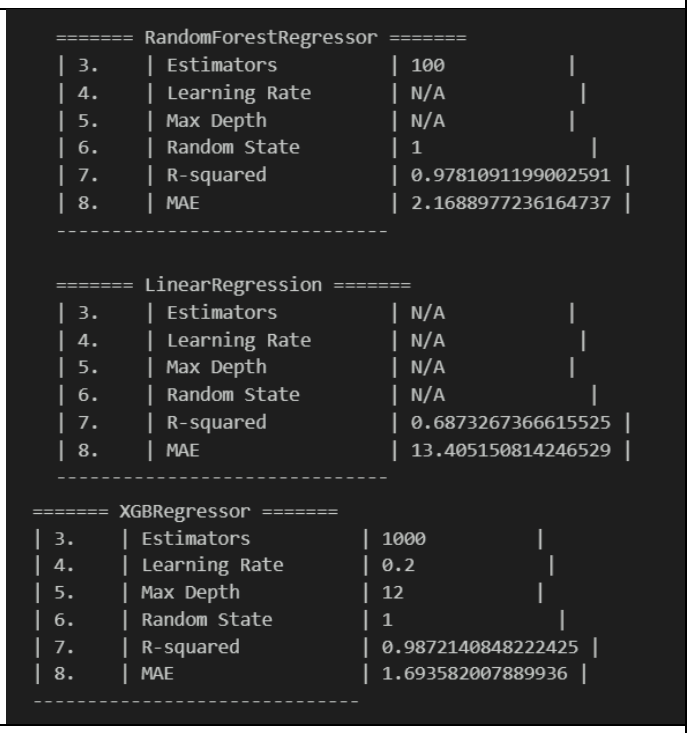
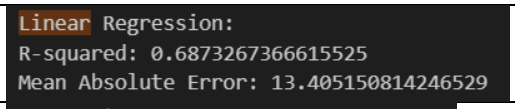
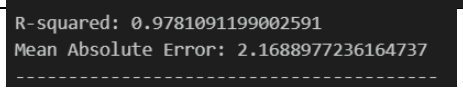
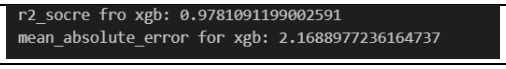
S.NO	Parameter	Values	Screenshot
1.	Model Summary	-	
2.	Accuracy		
	i)Linear regression	0.6873267366615525	
	ii) Random Forest regressor	0.9781091199002591	
	iii)XGBRegressor	0.9872140848222425	

The codes for the screenshots:

```python
# Linear Regression
lr_pipe = Pipeline(steps=[
    ('step1', trf),
    ('step2', StandardScaler()),
    ('step3', LinearRegression())
])

lr_pipe.fit(X_train, y_train)
lr_y_pred = lr_pipe.predict(X_test)
print("Linear Regression:")
print("R-squared:", r2_score(y_test, lr_y_pred))
print("Mean Absolute Error:", mean_absolute_error(y_test, lr_y_pred))
```

✓ 0.2s

```python
# Random Forest Regressor
rf_pipe = Pipeline(steps=[
    ('step1', trf),
    ('step2', StandardScaler()),
    ('step3', RandomForestRegressor(random_state=1))
])

rf_pipe.fit(X_train, y_train)
rf_y_pred = rf_pipe.predict(X_test)
print("Random Forest:")
print("R-squared:", r2_score(y_test, rf_y_pred))
print("Mean Absolute Error:", mean_absolute_error(y_test, rf_y_pred))
print("-" * 40)
```

✓ 30.5s

```python
pipe = Pipeline(steps=[
    ('step1',trf),
    ('step2',StandardScaler()),
    ('step3',XGBRegressor(n_estimators=1000,learning_rate=0.2,max_depth=12,random_state=1))
])
```

✓ 0.0s

```python
pipe.fit(X_train,y_train)
y_pred = pipe.predict(X_test)
print(f"r2_socre fro xgb: {r2_score(y_test,y_pred)}")
print(f"mean_absolute_error for xgb: {mean_absolute_error(y_test,y_pred)}")
```

✓ 30.5s

Model Summary Code:

```python
from sklearn.metrics import r2_score, mean_absolute_error

# Assuming rf_pipe, lr_pipe, and xgb_pipe are your pipelines
pipes = [rf_pipe, lr_pipe, xgb_pipe]

for idx, pipe in enumerate(pipes, start=1):
    # Fit the pipeline
    pipe.fit(X_train, y_train)

    # Make predictions
    y_pred = pipe.predict(X_test)

    # Get model parameters
    n_estimators = getattr(pipe.named_steps['step3'], 'n_estimators', 'N/A')
    learning_rate = getattr(pipe.named_steps['step3'], 'learning_rate', 'N/A')
    max_depth = getattr(pipe.named_steps['step3'], 'max_depth', 'N/A')
    random_state = getattr(pipe.named_steps['step3'], 'random_state', 'N/A')

    # Handle max_depth being None
    max_depth_str = str(max_depth) if max_depth is not None else 'N/A'

    # Calculate R-squared and MAE
    r_squared = r2_score(y_test, y_pred)
    mae = mean_absolute_error(y_test, y_pred)

    # Print the results
    print(f"\n{'='*7} {pipe.named_steps['step3'].__class__.__name__} {'='*7}")
    print(f"| 3.    | Estimators          | {n_estimators:<6}        |")
    print(f"| 4.    | Learning Rate       | {learning_rate:<6}         |")
    print(f"| 5.    | Max Depth           | {max_depth_str:<6}        |")
    print(f"| 6.    | Random State        | {random_state:<6}         |")
    print(f"| 7.    | R-squared           | {r_squared:<17} |")
    print(f"| 8.    | MAE                 | {mae:<17} |")
    print('-' * 30)
```