```
@tracer(cat_col = ['Gender', 'Education'], numerical_col = [])
def loan_pipeline(f_path = '../pipelines/loan_train.csv'):
   data = pd.read_csv(f_path)
   # Loan_ID is not needed in training or prediction
   data = data.drop('Loan_ID', axis=1)
     data = data.drop('Loan_Status', axis=1)
   numeric_features = data.select_dtypes(include=['int64', 'float64']).columns
   categorical_features = data.select_dtypes(include=['object']).drop(['Loan_Status'], axis=1).columns
   # do transformer on numeric & categorical data respectively
   numeric_transformer = Pipeline(steps=[
        ('imputer', SimpleImputer(strategy='median')),
        ('scaler', StandardScaler())])
   categorical_transformer = Pipeline(steps=[
        ('imputer', SimpleImputer(strategy='constant', fill_value='missing')),
        ('onehot', OneHotEncoder(handle_unknown='ignore'))])
   preprocessor = ColumnTransformer(
        transformers=[
           ('num', numeric_transformer, numeric_features),
           ('cat', categorical_transformer, categorical_features)])
   # classifier
   pipeline = Pipeline(steps=[('preprocessor', preprocessor),
                          ('classifier', RandomForestClassifier())])
   return pipeline
```

CoapplicantIncome\_\_imputer,

op=SimpleImputer

CoapplicantIncome\_\_scaler,

op=StandardScaler

LoanAmount\_imputer,

op=SimpleImputer

LoanAmount\_scaler,

op=StandardScaler

Loan\_Amount\_Term\_\_imputer,

op=SimpleImputer

Loan\_Amount\_Term\_\_scaler,

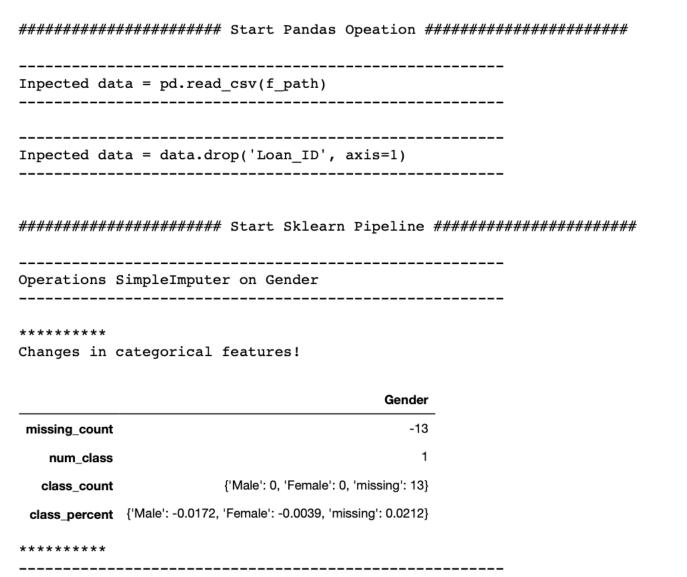
op=StandardScaler

ApplicantIncome\_imputer,

op=SimpleImputer

ApplicantIncome\_\_scaler,

op=StandardScaler



Operations OneHotEncoder on Gender

\*\*\*\*\*

\_\_\_\_\_\_

op=RandomForestClassifier

Changes in categorical features!

	Gender	
missing_count	0	
num_class	-1	
class_count	{0.0: 502, 1.0: 112}	
class_percent	{0.0: 0.8176, 1.0: 0.1824}	
****		
	OneHotEncoder on Education	
*****		
Changes in	categorical features!	

class\_count {1.0: 480, 0.0: 134}
class\_percent {1.0: 0.7818, 0.0: 0.2182}

missing count

num\_class

