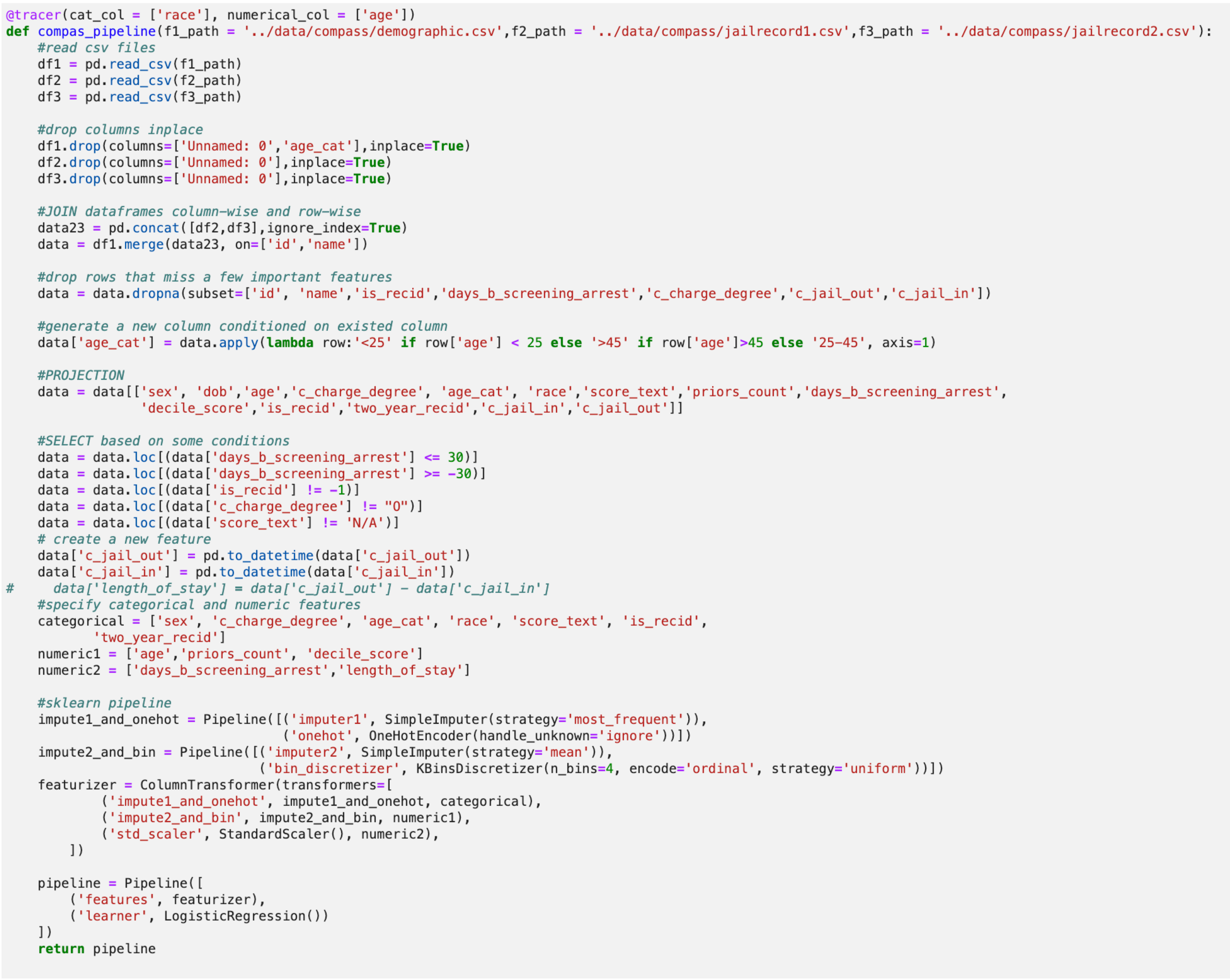
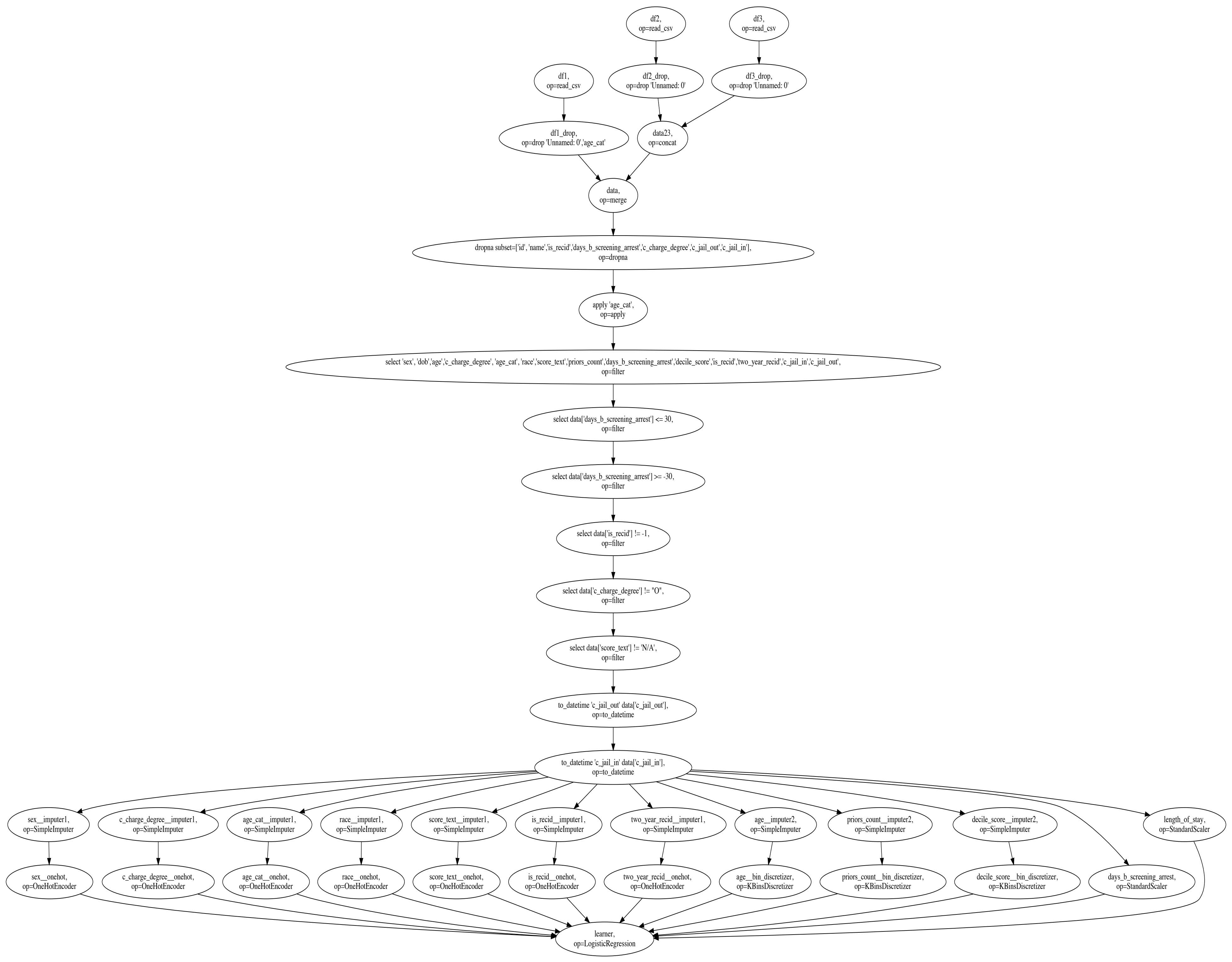
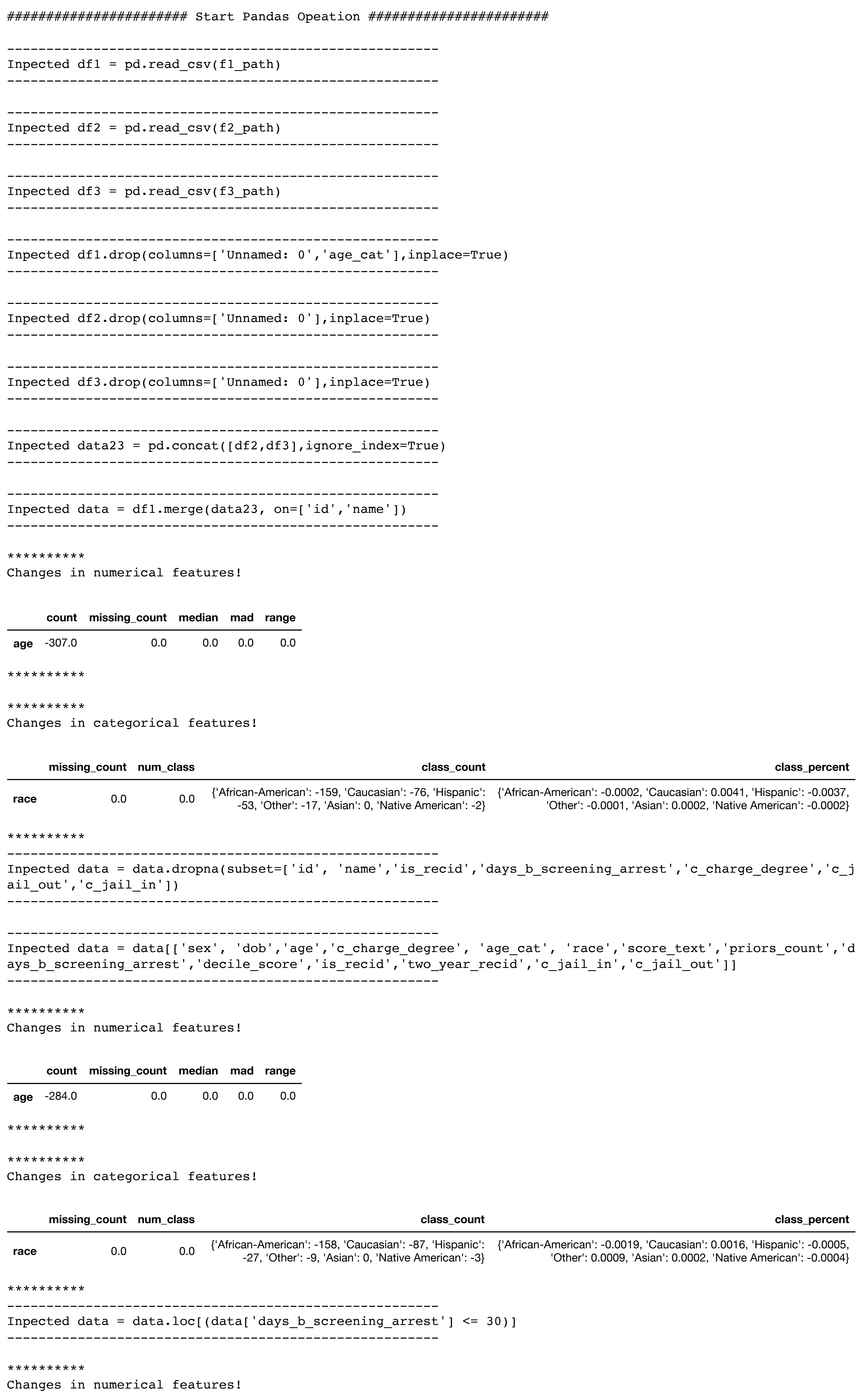


```python
@tracer(cat_col = ['race'], numerical_col = ['age'])
def compas_pipeline(f1_path = '../data/compass/demographic.csv',f2_path = '../data/compass/jailrecord1.csv',f3_path = '../data/compass/jailrecord2.csv'):
    #read csv files
    df1 = pd.read_csv(f1_path)
    df2 = pd.read_csv(f2_path)
    df3 = pd.read_csv(f3_path)

    #drop columns inplace
    df1.drop(columns=['Unnamed: 0','age_cat'],inplace=True)
    df2.drop(columns=['Unnamed: 0'],inplace=True)
    df3.drop(columns=['Unnamed: 0'],inplace=True)

    #JOIN dataframes column-wise and row-wise
    data23 = pd.concat([df2,df3],ignore_index=True)
    data = df1.merge(data23, on=['id','name'])

    #drop rows that miss a few important features
    data = data.dropna(subset=['id', 'name','is_recid','days_b_screening_arrest','c_charge_degree','c_jail_out','c_jail_in'])

    #generate a new column conditioned on existed column
    data['age_cat'] = data.apply(lambda row:'<25' if row['age'] < 25 else '>45' if row['age']>45 else '25-45', axis=1)

    #PROJECTION
    data = data[['sex', 'dob','age','c_charge_degree', 'age_cat', 'race','score_text','priors_count','days_b_screening_arrest',
                 'decile_score','is_recid','two_year_recid','c_jail_in','c_jail_out']]

    #SELECT based on some conditions
    data = data.loc[(data['days_b_screening_arrest'] <= 30)]
    data = data.loc[(data['days_b_screening_arrest'] >= -30)]
    data = data.loc[(data['is_recid'] != -1)]
    data = data.loc[(data['c_charge_degree'] != "O")]
    data = data.loc[(data['score_text'] != 'N/A')]
    # create a new feature
    data['c_jail_out'] = pd.to_datetime(data['c_jail_out'])
    data['c_jail_in'] = pd.to_datetime(data['c_jail_in'])
#     data['length_of_stay'] = data['c_jail_out'] - data['c_jail_in']
    #specify categorical and numeric features
    categorical = ['sex', 'c_charge_degree', 'age_cat', 'race', 'score_text', 'is_recid',
           'two_year_recid']
    numeric1 = ['age','priors_count', 'decile_score']
    numeric2 = ['days_b_screening_arrest','length_of_stay']

    #sklearn pipeline
    impute1_and_onehot = Pipeline([('imputer1', SimpleImputer(strategy='most_frequent')),
                                  ('onehot', OneHotEncoder(handle_unknown='ignore'))])
    impute2_and_bin = Pipeline([('imputer2', SimpleImputer(strategy='mean')),
                               ('bin_discretizer', KBinsDiscretizer(n_bins=4, encode='ordinal', strategy='uniform'))])
    featurizer = ColumnTransformer(transformers=[
            ('impute1_and_onehot', impute1_and_onehot, categorical),
            ('impute2_and_bin', impute2_and_bin, numeric1),
            ('std_scaler', StandardScaler(), numeric2),
        ])

    pipeline = Pipeline([
        ('features', featurizer),
        ('learner', LogisticRegression())
    ])
    return pipeline
```





```
###################### Start Pandas Opeation ######################

---------------------------------------------------------
Inpected df1 = pd.read_csv(f1_path)
---------------------------------------------------------

---------------------------------------------------------
Inpected df2 = pd.read_csv(f2_path)
---------------------------------------------------------

---------------------------------------------------------
Inpected df3 = pd.read_csv(f3_path)
---------------------------------------------------------

---------------------------------------------------------
Inpected df1.drop(columns=['Unnamed: 0','age_cat'],inplace=True)
---------------------------------------------------------

---------------------------------------------------------
Inpected df2.drop(columns=['Unnamed: 0'],inplace=True)
---------------------------------------------------------

---------------------------------------------------------
Inpected df3.drop(columns=['Unnamed: 0'],inplace=True)
---------------------------------------------------------

---------------------------------------------------------
Inpected data23 = pd.concat([df2,df3],ignore_index=True)
---------------------------------------------------------

---------------------------------------------------------
Inpected data = df1.merge(data23, on=['id','name'])
---------------------------------------------------------

**********
Changes in numerical features!
```

	count	missing_count	median	mad	range
age	-307.0	0.0	0.0	0.0	0.0

```
**********

**********
Changes in categorical features!
```

	missing_count	num_class	class_count	class_percent
race	0.0	0.0	{'African-American': -159, 'Caucasian': -76, 'Hispanic': -53, 'Other': -17, 'Asian': 0, 'Native American': -2}	{'African-American': -0.0002, 'Caucasian': 0.0041, 'Hispanic': -0.0037, 'Other': -0.0001, 'Asian': 0.0002, 'Native American': -0.0002}

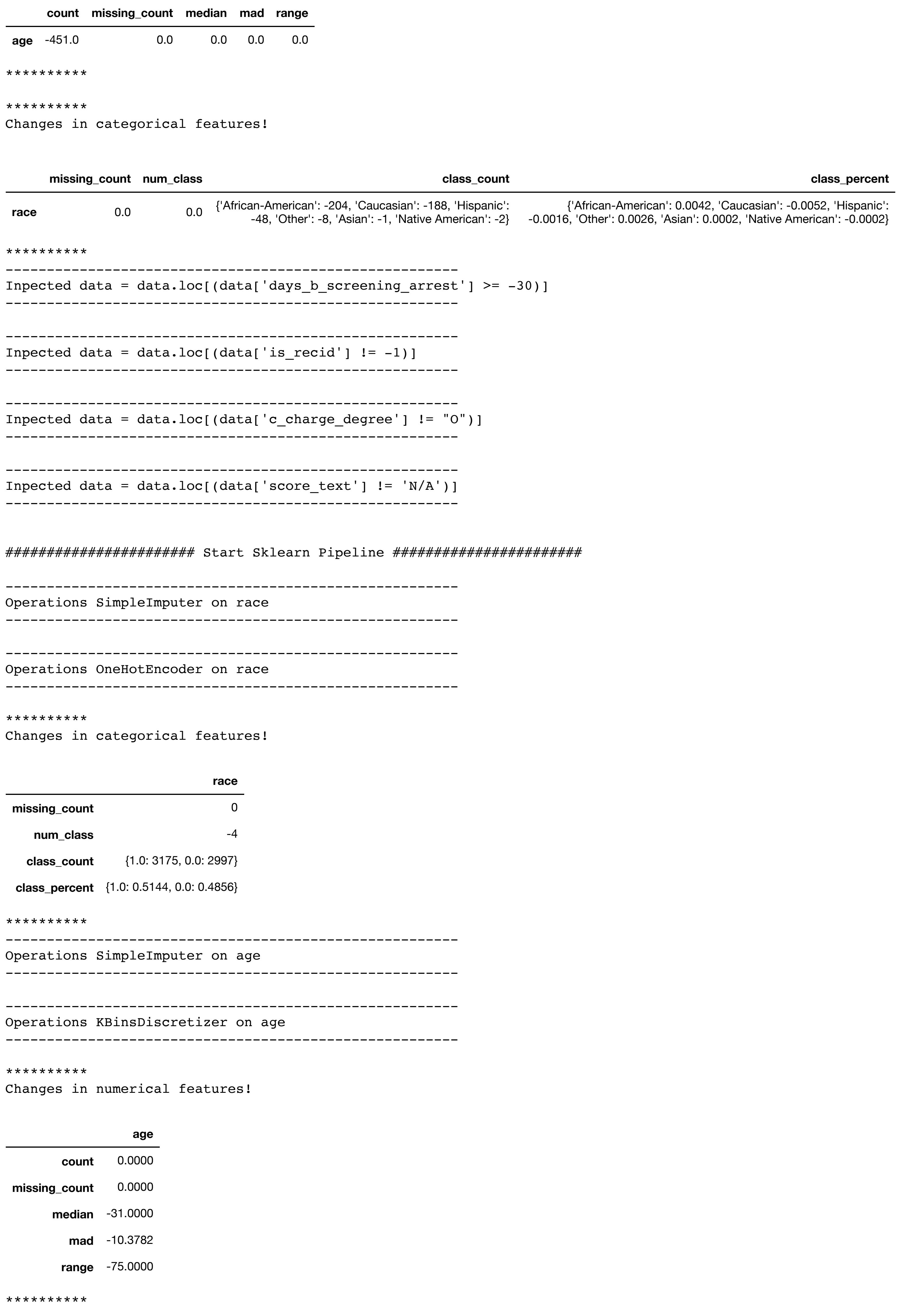
```
**********
---------------------------------------------------------
Inpected data = data.dropna(subset=['id', 'name','is_recid','days_b_screening_arrest','c_charge_degree','c_j
ail_out','c_jail_in'])
---------------------------------------------------------

---------------------------------------------------------
Inpected data = data[['sex', 'dob','age','c_charge_degree', 'age_cat', 'race','score_text','priors_count','d
ays_b_screening_arrest','decile_score','is_recid','two_year_recid','c_jail_in','c_jail_out']]
---------------------------------------------------------

**********
Changes in numerical features!
```

	count	missing_count	median	mad	range
age	-284.0	0.0	0.0	0.0	0.0

```
**********

**********
Changes in categorical features!
```

	missing_count	num_class	class_count	class_percent
race	0.0	0.0	{'African-American': -158, 'Caucasian': -87, 'Hispanic': -27, 'Other': -9, 'Asian': 0, 'Native American': -3}	{'African-American': -0.0019, 'Caucasian': 0.0016, 'Hispanic': -0.0005, 'Other': 0.0009, 'Asian': 0.0002, 'Native American': -0.0004}

```
**********
---------------------------------------------------------
Inpected data = data.loc[(data['days_b_screening_arrest'] <= 30)]
---------------------------------------------------------

**********
Changes in numerical features!
```



	count	missing_count	median	mad	range
age	-451.0	0.0	0.0	0.0	0.0

```
**********

**********
Changes in categorical features!
```

	missing_count	num_class	class_count	class_percent
race	0.0	0.0	{'African-American': -204, 'Caucasian': -188, 'Hispanic': -48, 'Other': -8, 'Asian': -1, 'Native American': -2}	{'African-American': 0.0042, 'Caucasian': -0.0052, 'Hispanic': -0.0016, 'Other': 0.0026, 'Asian': 0.0002, 'Native American': -0.0002}

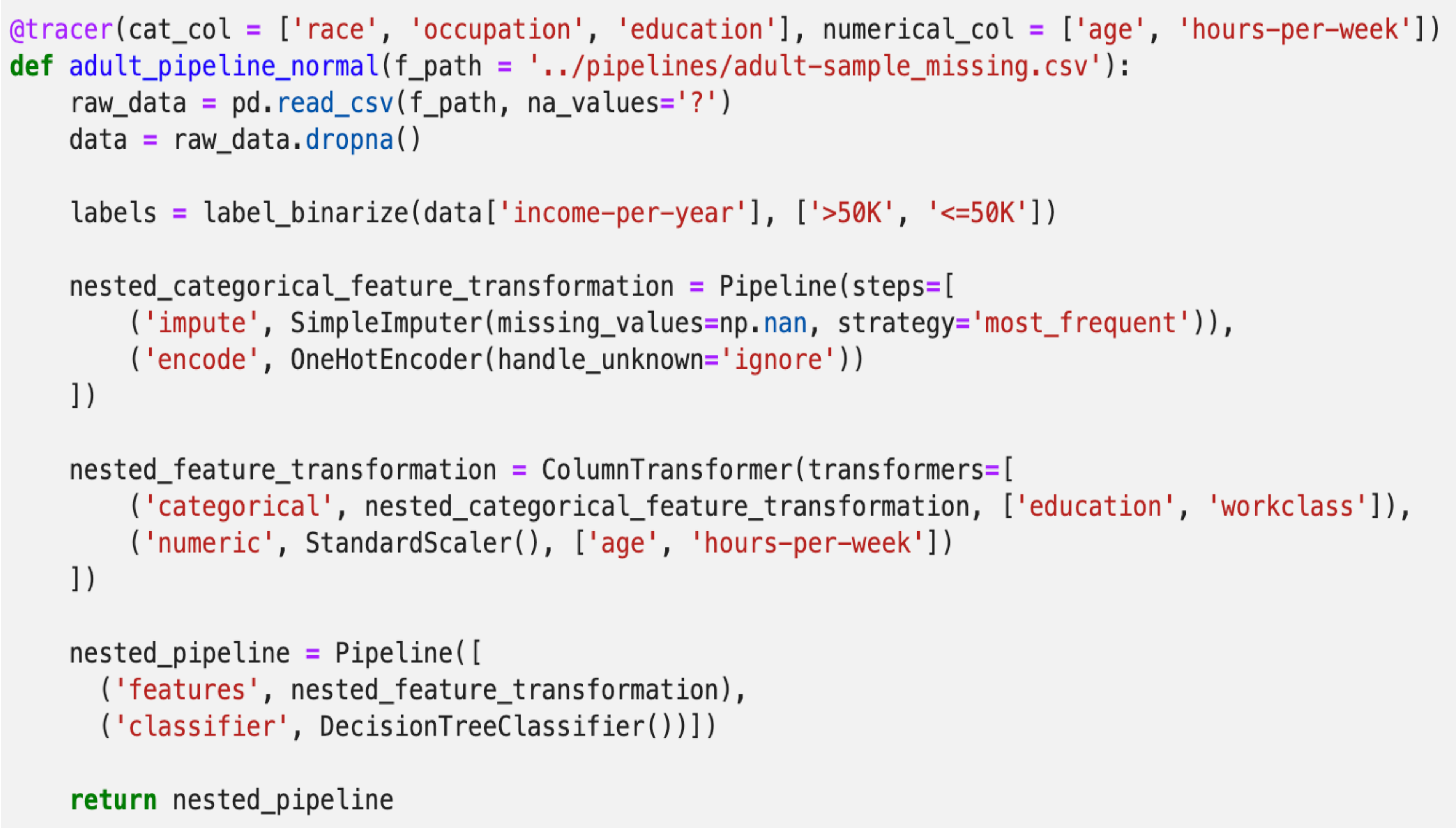
```
**********
---------------------------------------------------------
Inpected data = data.loc[(data['days_b_screening_arrest'] >= -30)]
---------------------------------------------------------

---------------------------------------------------------
Inpected data = data.loc[(data['is_recid'] != -1)]
---------------------------------------------------------

---------------------------------------------------------
Inpected data = data.loc[(data['c_charge_degree'] != "O")]
---------------------------------------------------------

---------------------------------------------------------
Inpected data = data.loc[(data['score_text'] != 'N/A')]
---------------------------------------------------------

###################### Start Sklearn Pipeline ######################

---------------------------------------------------------
Operations SimpleImputer on race
---------------------------------------------------------

---------------------------------------------------------
Operations OneHotEncoder on race
---------------------------------------------------------

**********
Changes in categorical features!
```

	race
missing_count	0
num_class	-4
class_count	{1.0: 3175, 0.0: 2997}
class_percent	{1.0: 0.5144, 0.0: 0.4856}

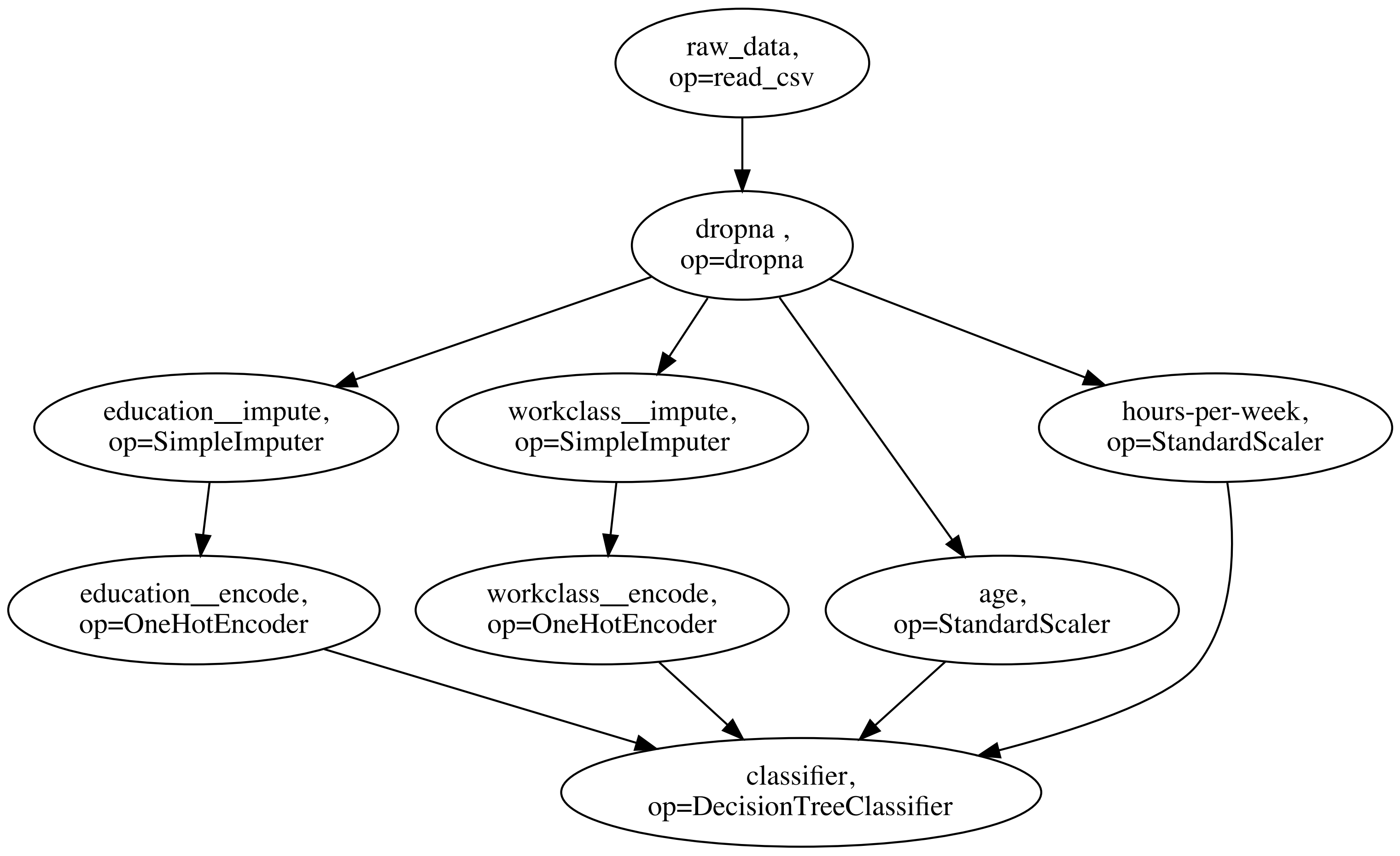
```
**********
---------------------------------------------------------
Operations SimpleImputer on age
---------------------------------------------------------

---------------------------------------------------------
Operations KBinsDiscretizer on age
---------------------------------------------------------

**********
Changes in numerical features!
```

	age
count	0.0000
missing_count	0.0000
median	-31.0000
mad	-10.3782
range	-75.0000

```
**********
```



```python
@tracer(cat_col = ['race', 'occupation', 'education'], numerical_col = ['age', 'hours-per-week'])
def adult_pipeline_normal(f_path = '../pipelines/adult-sample_missing.csv'):
    raw_data = pd.read_csv(f_path, na_values='?')
    data = raw_data.dropna()

    labels = label_binarize(data['income-per-year'], ['>50K', '<=50K'])

    nested_categorical_feature_transformation = Pipeline(steps=[
        ('impute', SimpleImputer(missing_values=np.nan, strategy='most_frequent')),
        ('encode', OneHotEncoder(handle_unknown='ignore'))
    ])

    nested_feature_transformation = ColumnTransformer(transformers=[
        ('categorical', nested_categorical_feature_transformation, ['education', 'workclass']),
        ('numeric', StandardScaler(), ['age', 'hours-per-week'])
    ])

    nested_pipeline = Pipeline([
      ('features', nested_feature_transformation),
      ('classifier', DecisionTreeClassifier())])

    return nested_pipeline
```



```
###################### Start Pandas Opeation ######################

-------------------------------------------------------
Inpected raw_data = pd.read_csv(f_path, na_values='?')
-------------------------------------------------------

**********
Changes in numerical features!
```

	count	missing_count	median	mad	range
age	-14.0	0.0	0.0	-0.7413	-23.0
hours-per-week	-14.0	0.0	0.0	0.0000	0.0

```
**********

**********
Changes in categorical features!
```

	missing_count	num_class	class_count	class_percent
race	-4.0	0.0	{'White': -6, 'Black': -2, 'Amer-Indian-Eskimo': -2, 'Other': 0, 'Asian-Pac-Islander': 0}	{'White': 0.0271, 'Black': -0.0111, 'Amer-Indian-Eskimo': -0.0184, 'Other': 0.0012, 'Asian-Pac-Islander': 0.0012}
occupation	-8.0	0.0	{'Exec-managerial': 0, 'Adm-clerical': 0, 'Craft-repair': -1, 'Sales': -1, 'Other-service': 0, 'Prof-specialty': -2, 'Transport-moving': -1, 'Machine-op-inspct': 0, 'Farming-fishing': 0, 'Handlers-cleaners': 0, 'Tech-support': 0, 'Protective-serv': -1}	{'Exec-managerial': 0.0106, 'Adm-clerical': 0.0099, 'Craft-repair': -0.0018, 'Sales': -0.0033, 'Other-service': 0.0068, 'Prof-specialty': -0.0157, 'Transport-moving': -0.0056, 'Machine-op-inspct': 0.0046, 'Farming-fishing': 0.0023, 'Handlers-cleaners': 0.0015, 'Tech-support': 0.0008, 'Protective-serv': -0.0101}
education	-2.0	0.0	{'HS-grad': -3, 'Bachelors': -1, 'Some-college': -4, '11th': -2, 'Masters': -2, '7th-8th': 0, '10th': 0, 'Assoc-voc': 0, 'Prof-school': 0, 'Assoc-acdm': 0, '12th': 0, '5th-6th': 0}	{'HS-grad': 0.0078, 'Bachelors': 0.0183, 'Some-college': -0.0138, '11th': -0.0133, 'Masters': -0.0147, '7th-8th': 0.0043, '10th': 0.0028, 'Assoc-voc': 0.0028, 'Prof-school': 0.0014, 'Assoc-acdm': 0.0014, '12th': 0.0014, '5th-6th': 0.0014}

```
**********
-------------------------------------------------------
Inpected data = raw_data.dropna()
-------------------------------------------------------

###################### Start Sklearn Pipeline ######################

-------------------------------------------------------
Operations SimpleImputer on education
-------------------------------------------------------

-------------------------------------------------------
Operations OneHotEncoder on education
-------------------------------------------------------

**********
Changes in categorical features!
```

	education
missing_count	0
num_class	-10
class_count	{0.0: 84, 1.0: 2}
class_percent	{0.0: 0.9767, 1.0: 0.0233}

```
**********
```

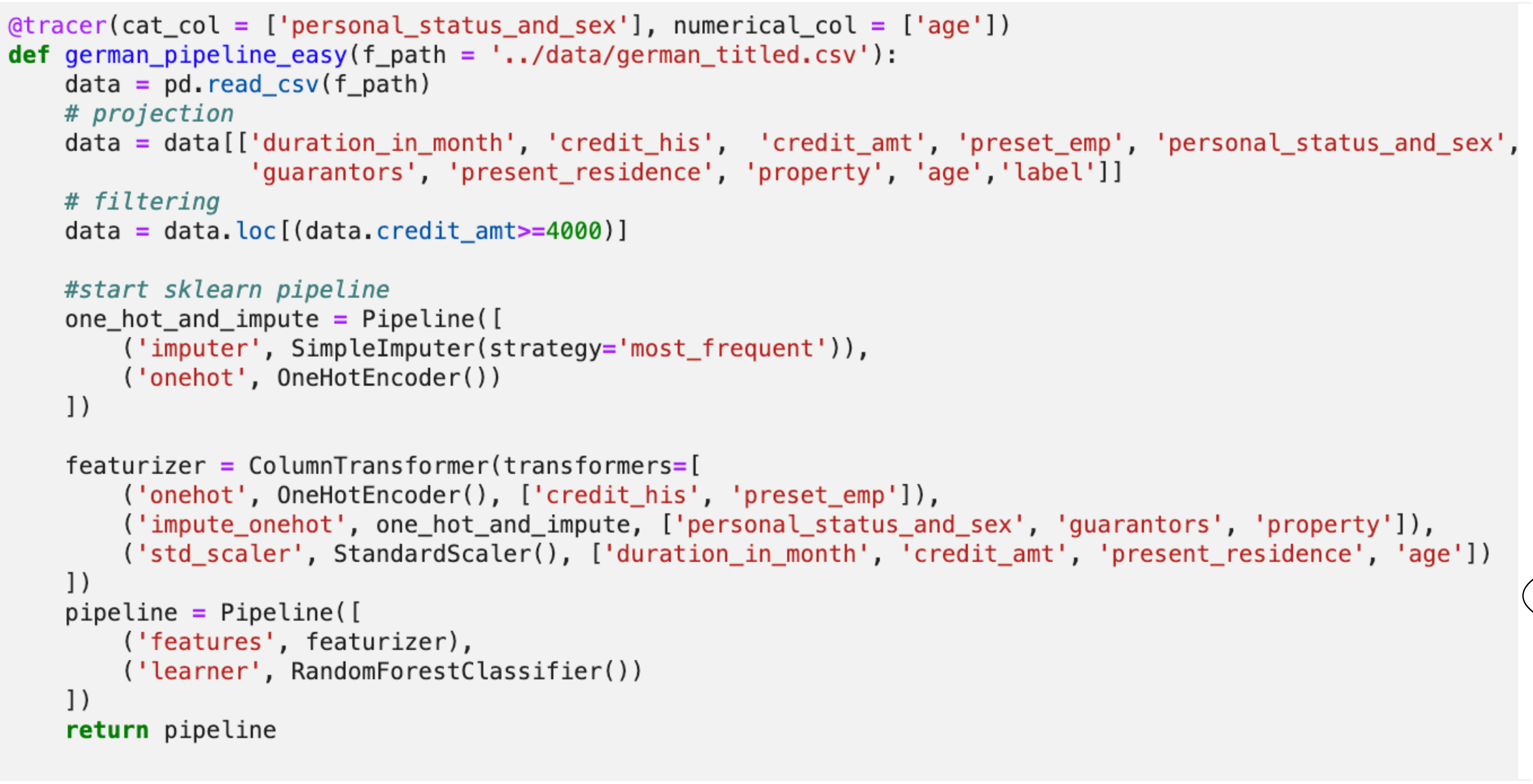
```
-------------------------------------------------------
Operations StandardScaler on age
-------------------------------------------------------

**********
Changes in numerical features!
```

	age
count	0.0000
missing_count	0.0000
median	-36.0972
mad	-12.8320
range	-44.6418

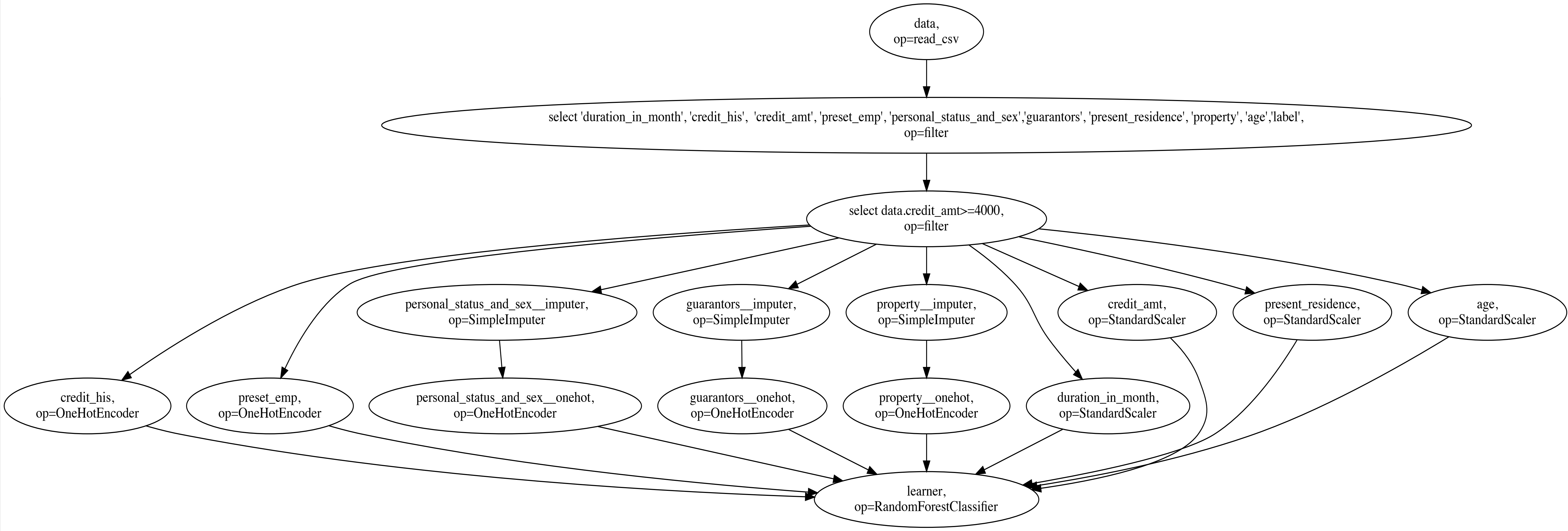
```
**********

-------------------------------------------------------
Operations StandardScaler on hours-per-week
-------------------------------------------------------

**********
Changes in numerical features!
```

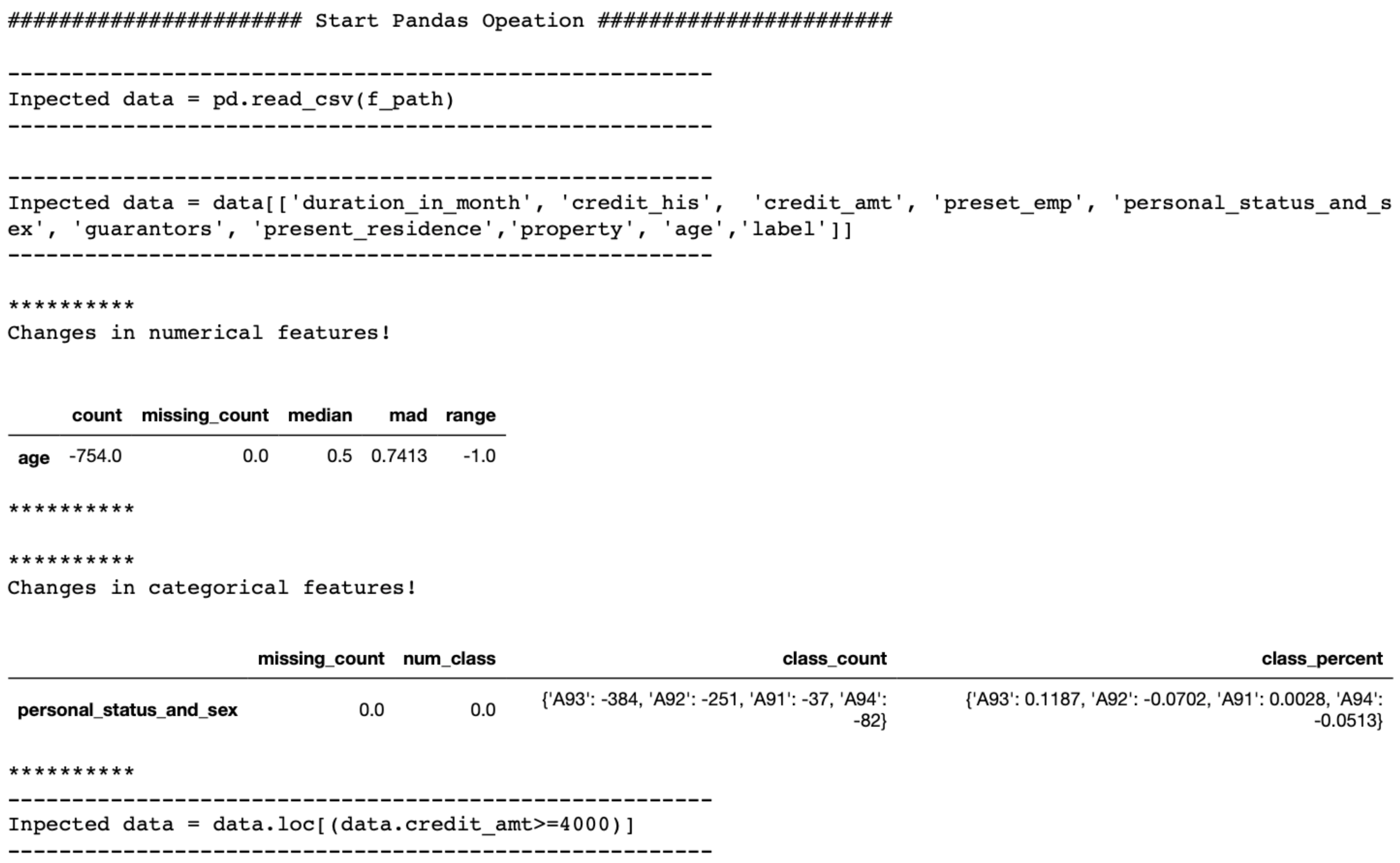
	hours-per-week
count	0.0000
missing_count	0.0000
median	-40.1126
mad	-1.3509
range	-63.7813

```
**********
```



```python
@tracer(cat_col = ['personal_status_and_sex'], numerical_col = ['age'])
def german_pipeline_easy(f_path = '../data/german_titled.csv'):
    data = pd.read_csv(f_path)
    # projection
    data = data[['duration_in_month', 'credit_his',  'credit_amt', 'preset_emp', 'personal_status_and_sex',
                 'guarantors', 'present_residence', 'property', 'age','label']]
    # filtering
    data = data.loc[(data.credit_amt>=4000)]

    #start sklearn pipeline
    one_hot_and_impute = Pipeline([
        ('imputer', SimpleImputer(strategy='most_frequent')),
        ('onehot', OneHotEncoder())
    ])

    featurizer = ColumnTransformer(transformers=[
        ('onehot', OneHotEncoder(), ['credit_his', 'preset_emp']),
        ('impute_onehot', one_hot_and_impute, ['personal_status_and_sex', 'guarantors', 'property']),
        ('std_scaler', StandardScaler(), ['duration_in_month', 'credit_amt', 'present_residence', 'age'])
    ])
    pipeline = Pipeline([
        ('features', featurizer),
        ('learner', RandomForestClassifier())
    ])
    return pipeline
```





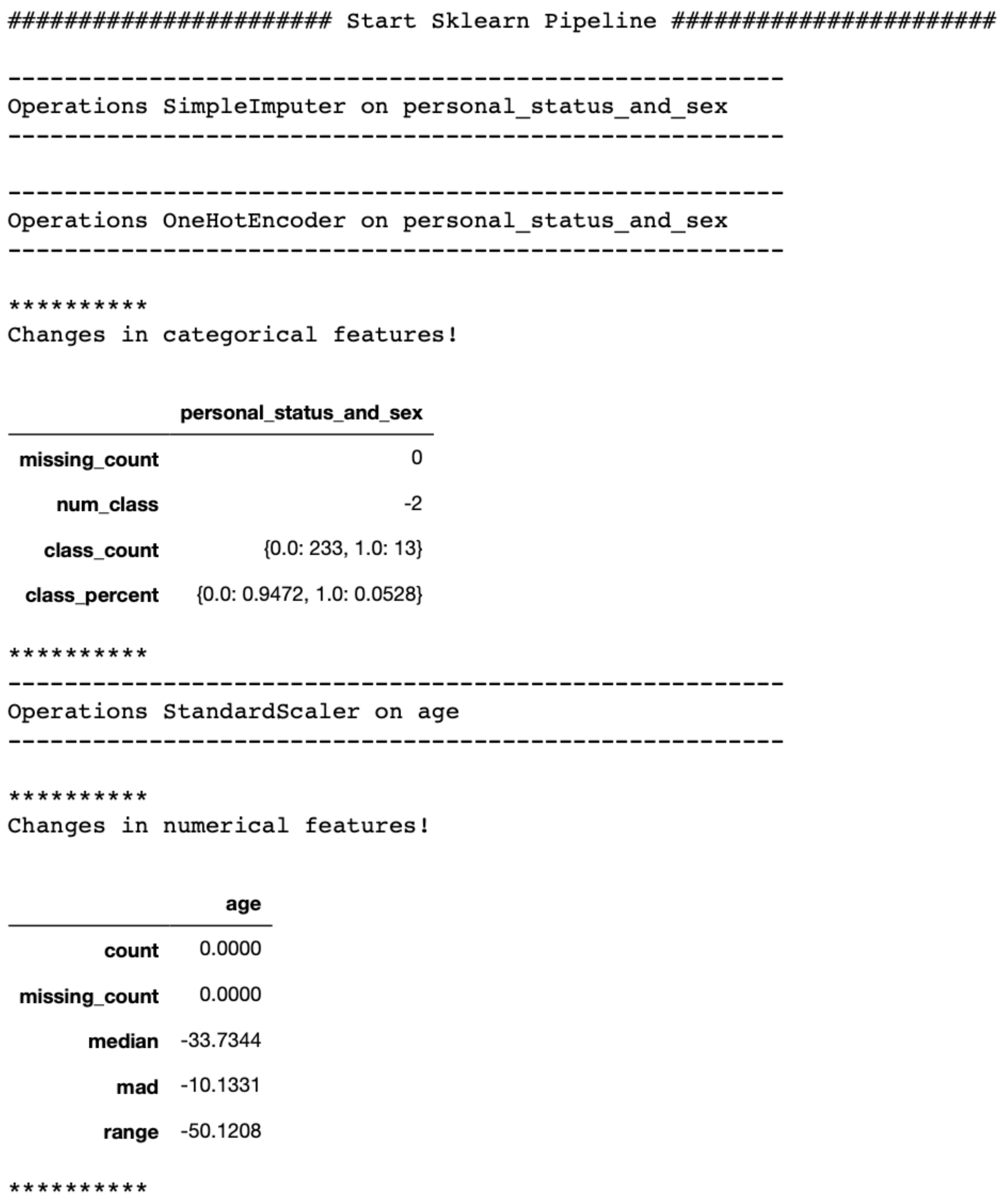
```
###################### Start Pandas Opeation ######################

-----------------------------------------------------------
Inpected data = pd.read_csv(f_path)
-----------------------------------------------------------

-----------------------------------------------------------
Inpected data = data[['duration_in_month', 'credit_his',  'credit_amt', 'preset_emp', 'personal_status_and_s
ex', 'guarantors', 'present_residence','property', 'age','label']]
-----------------------------------------------------------

**********
Changes in numerical features!
```

	count	missing_count	median	mad	range
age	-754.0	0.0	0.5	0.7413	-1.0

```
**********

**********
Changes in categorical features!
```

	missing_count	num_class	class_count	class_percent
personal_status_and_sex	0.0	0.0	{'A93': -384, 'A92': -251, 'A91': -37, 'A94': -82}	{'A93': 0.1187, 'A92': -0.0702, 'A91': 0.0028, 'A94': -0.0513}

```
**********
-----------------------------------------------------------
Inpected data = data.loc[(data.credit_amt>=4000)]
-----------------------------------------------------------
```



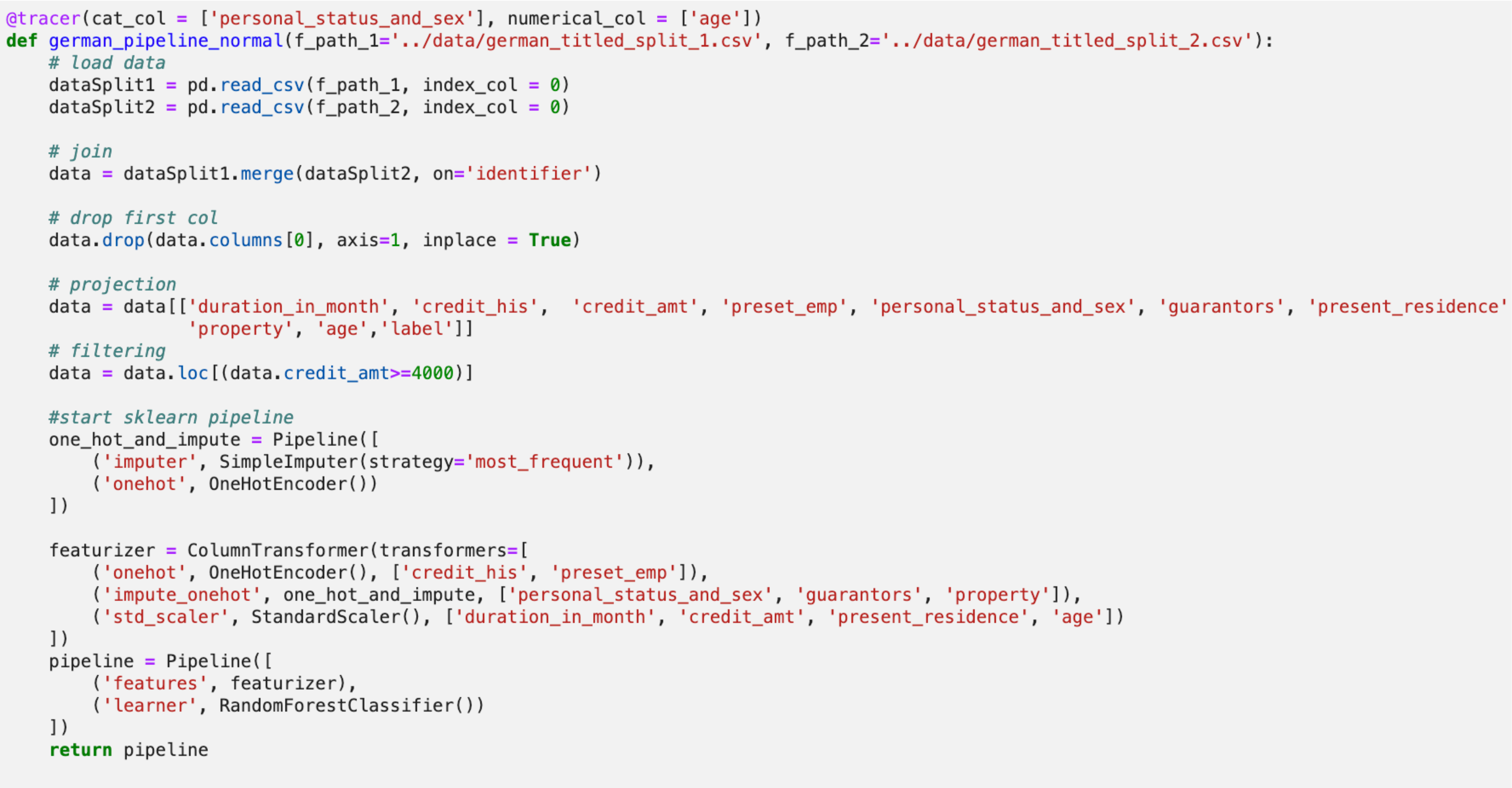
```
###################### Start Sklearn Pipeline ######################

-----------------------------------------------------------
Operations SimpleImputer on personal_status_and_sex
-----------------------------------------------------------

-----------------------------------------------------------
Operations OneHotEncoder on personal_status_and_sex
-----------------------------------------------------------

**********
Changes in categorical features!
```

	personal_status_and_sex
missing_count	0
num_class	-2
class_count	{0.0: 233, 1.0: 13}
class_percent	{0.0: 0.9472, 1.0: 0.0528}

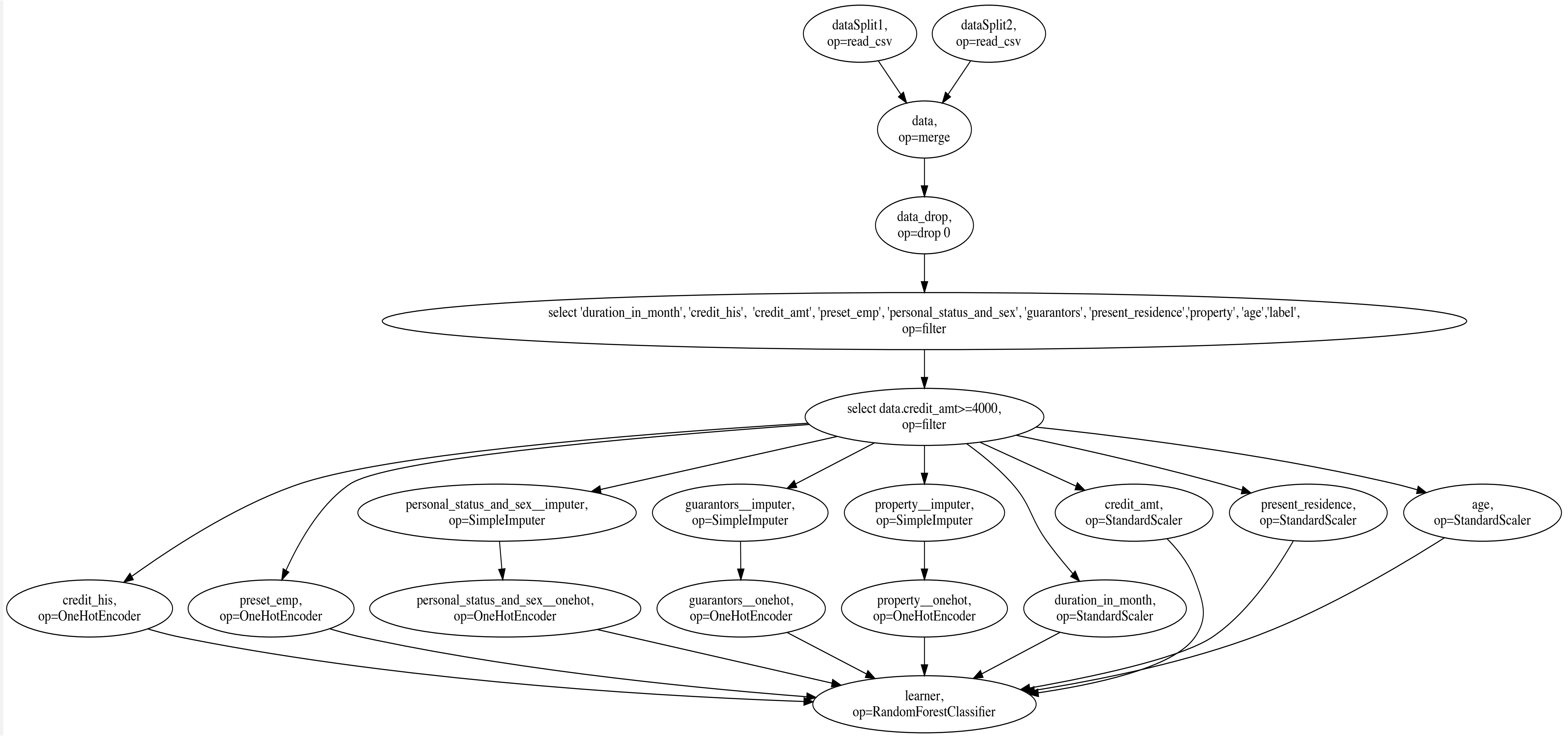
```
**********
-----------------------------------------------------------
Operations StandardScaler on age
-----------------------------------------------------------

**********
Changes in numerical features!
```

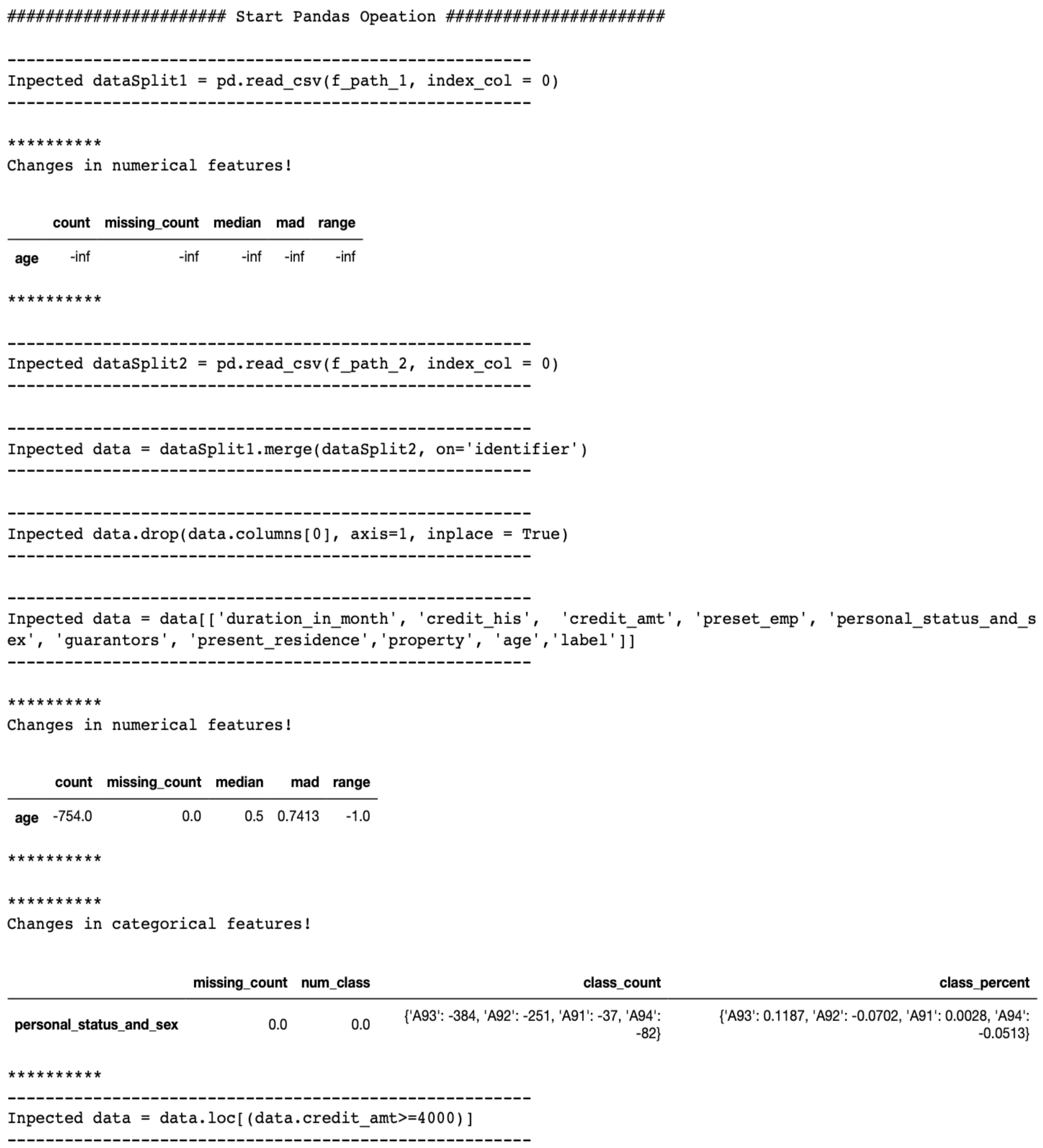
	age
count	0.0000
missing_count	0.0000
median	-33.7344
mad	-10.1331
range	-50.1208

```
**********
```



```python
@tracer(cat_col = ['personal_status_and_sex'], numerical_col = ['age'])
def german_pipeline_normal(f_path_1='../data/german_titled_split_1.csv', f_path_2='../data/german_titled_split_2.csv'):
    # load data
    dataSplit1 = pd.read_csv(f_path_1, index_col = 0)
    dataSplit2 = pd.read_csv(f_path_2, index_col = 0)

    # join
    data = dataSplit1.merge(dataSplit2, on='identifier')

    # drop first col
    data.drop(data.columns[0], axis=1, inplace = True)

    # projection
    data = data[['duration_in_month', 'credit_his',  'credit_amt', 'preset_emp', 'personal_status_and_sex', 'guarantors', 'present_residence'
                 'property', 'age','label']]
    # filtering
    data = data.loc[(data.credit_amt>=4000)]

    #start sklearn pipeline
    one_hot_and_impute = Pipeline([
        ('imputer', SimpleImputer(strategy='most_frequent')),
        ('onehot', OneHotEncoder())
    ])

    featurizer = ColumnTransformer(transformers=[
        ('onehot', OneHotEncoder(), ['credit_his', 'preset_emp']),
        ('impute_onehot', one_hot_and_impute, ['personal_status_and_sex', 'guarantors', 'property']),
        ('std_scaler', StandardScaler(), ['duration_in_month', 'credit_amt', 'present_residence', 'age'])
    ])
    pipeline = Pipeline([
        ('features', featurizer),
        ('learner', RandomForestClassifier())
    ])
    return pipeline
```





```
###################### Start Pandas Opeation ######################

--------------------------------------------------------
Inpected dataSplit1 = pd.read_csv(f_path_1, index_col = 0)
--------------------------------------------------------

**********
Changes in numerical features!
```

	count	missing_count	median	mad	range
age	-inf	-inf	-inf	-inf	-inf

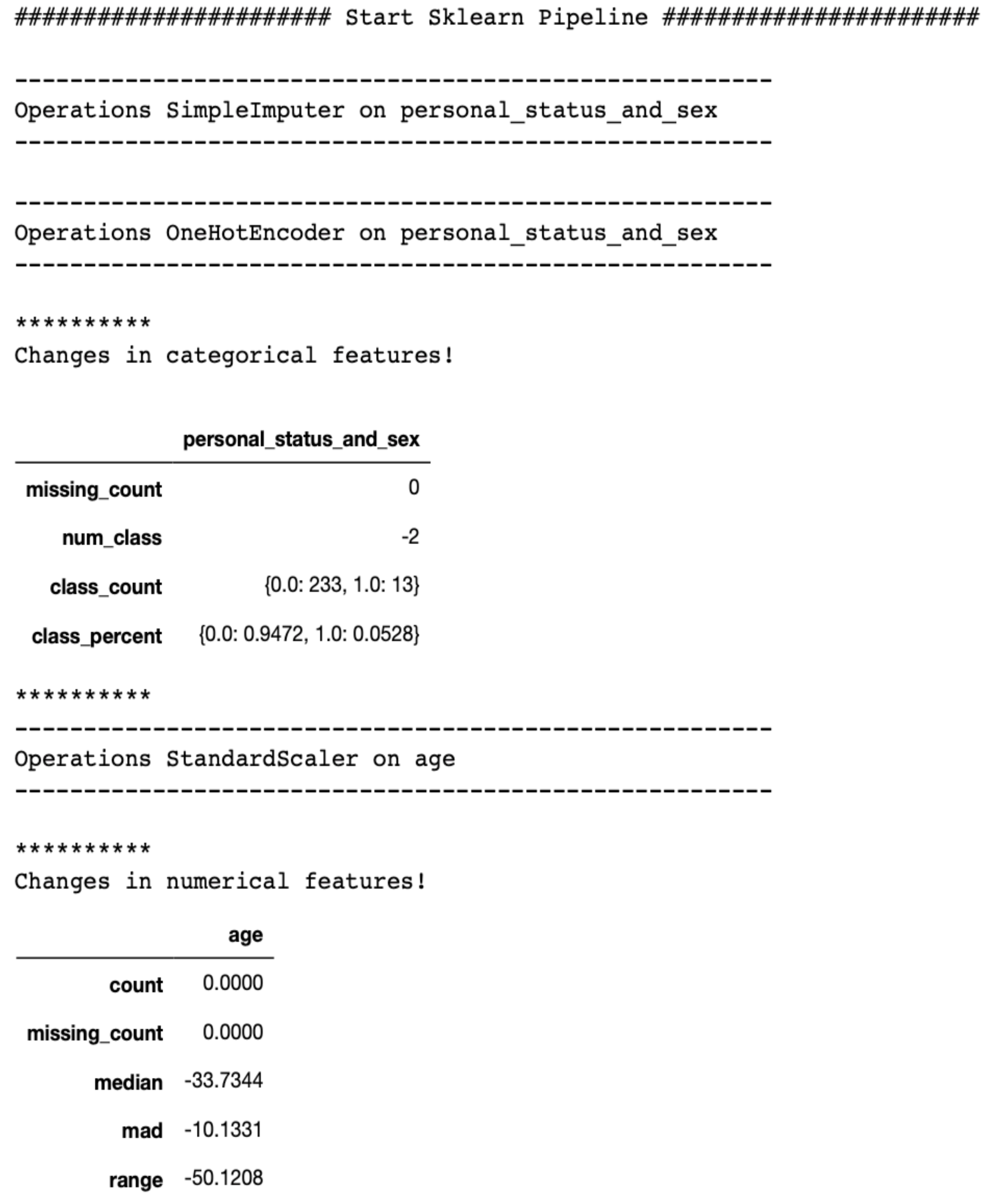
```
**********

--------------------------------------------------------
Inpected dataSplit2 = pd.read_csv(f_path_2, index_col = 0)
--------------------------------------------------------

--------------------------------------------------------
Inpected data = dataSplit1.merge(dataSplit2, on='identifier')
--------------------------------------------------------

--------------------------------------------------------
Inpected data.drop(data.columns[0], axis=1, inplace = True)
--------------------------------------------------------

--------------------------------------------------------
Inpected data = data[['duration_in_month', 'credit_his',  'credit_amt', 'preset_emp', 'personal_status_and_s
ex', 'guarantors', 'present_residence','property', 'age','label']]
--------------------------------------------------------

**********
Changes in numerical features!
```

	count	missing_count	median	mad	range
age	-754.0	0.0	0.5	0.7413	-1.0

```
**********

**********
Changes in categorical features!
```

	missing_count	num_class	class_count	class_percent
personal_status_and_sex	0.0	0.0	{'A93': -384, 'A92': -251, 'A91': -37, 'A94': -82}	{'A93': 0.1187, 'A92': -0.0702, 'A91': 0.0028, 'A94': -0.0513}

```
**********
--------------------------------------------------------
Inpected data = data.loc[(data.credit_amt>=4000)]
--------------------------------------------------------
```

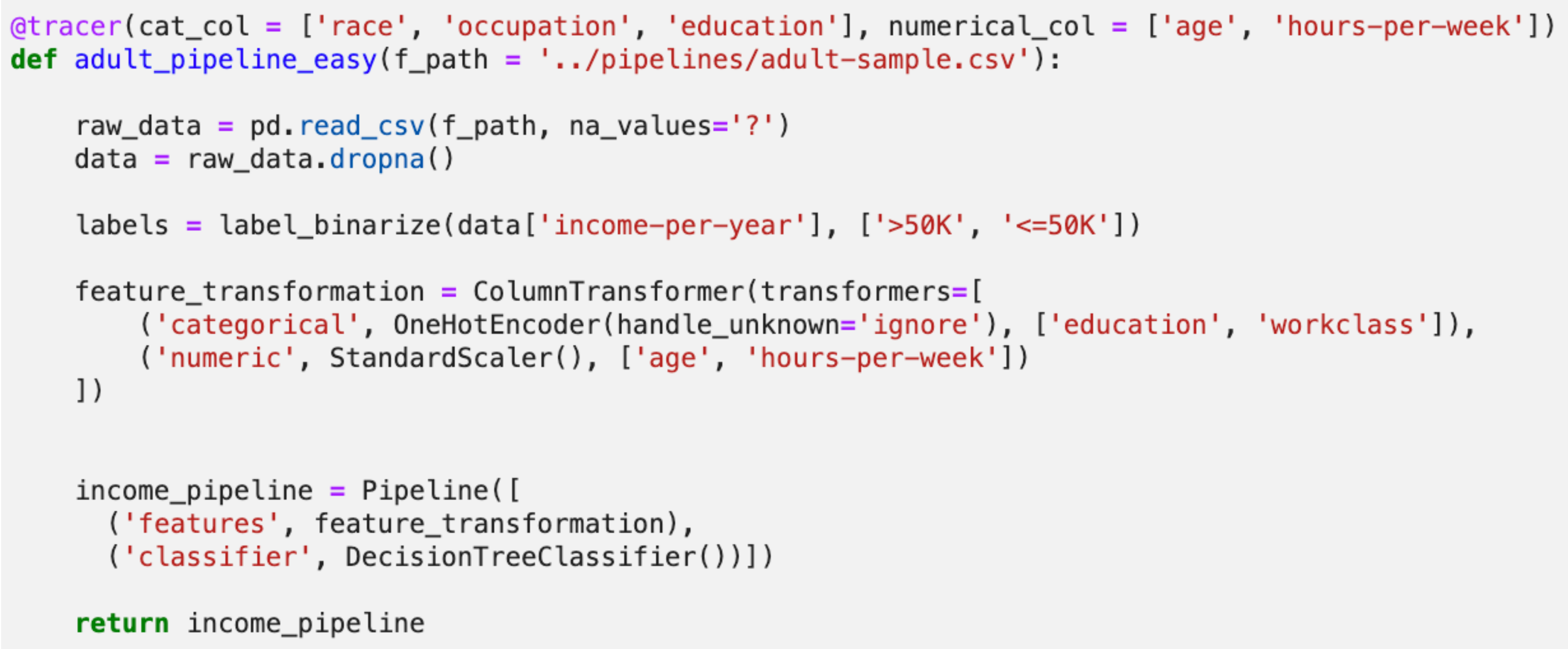


```
###################### Start Sklearn Pipeline ######################

--------------------------------------------------------
Operations SimpleImputer on personal_status_and_sex
--------------------------------------------------------

--------------------------------------------------------
Operations OneHotEncoder on personal_status_and_sex
--------------------------------------------------------

**********
Changes in categorical features!
```

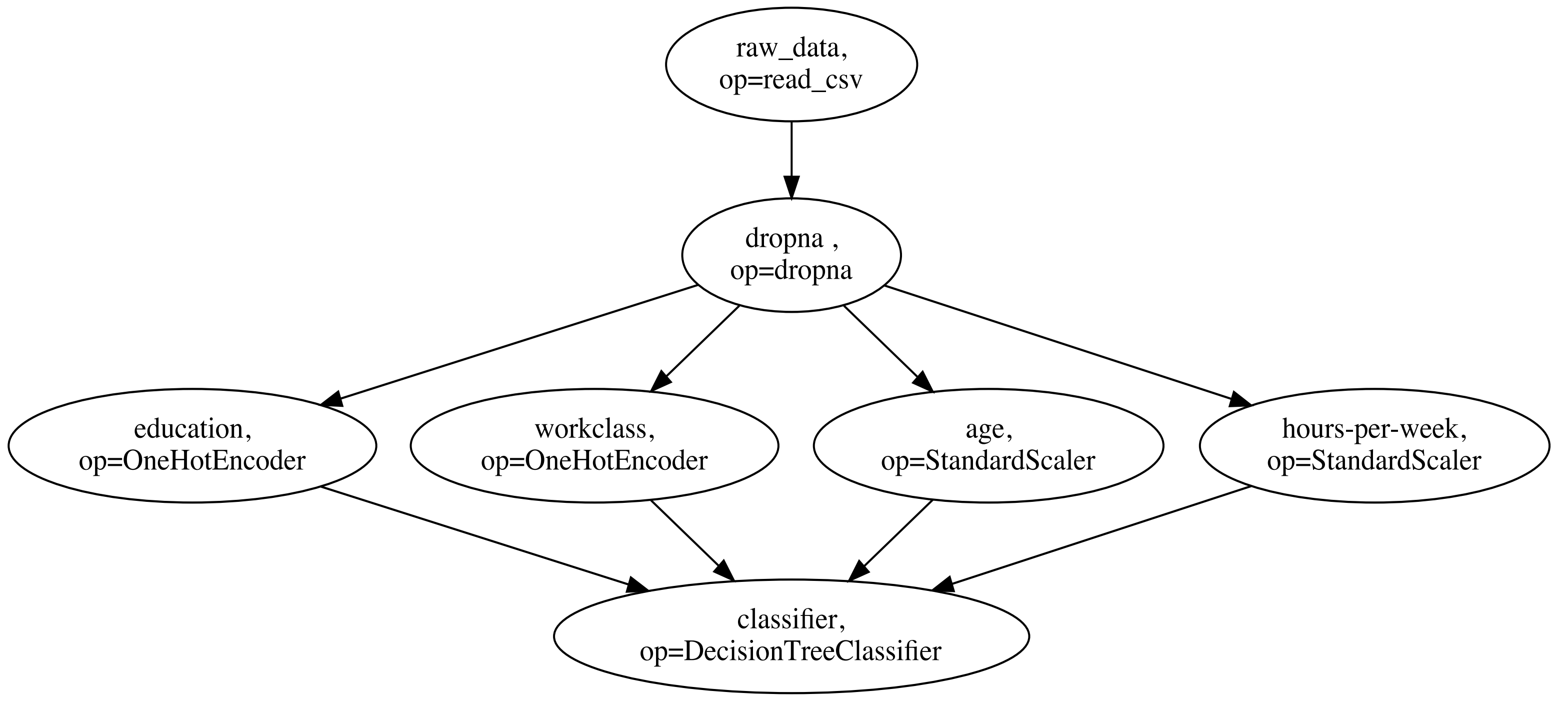
	personal_status_and_sex
missing_count	0
num_class	-2
class_count	{0.0: 233, 1.0: 13}
class_percent	{0.0: 0.9472, 1.0: 0.0528}

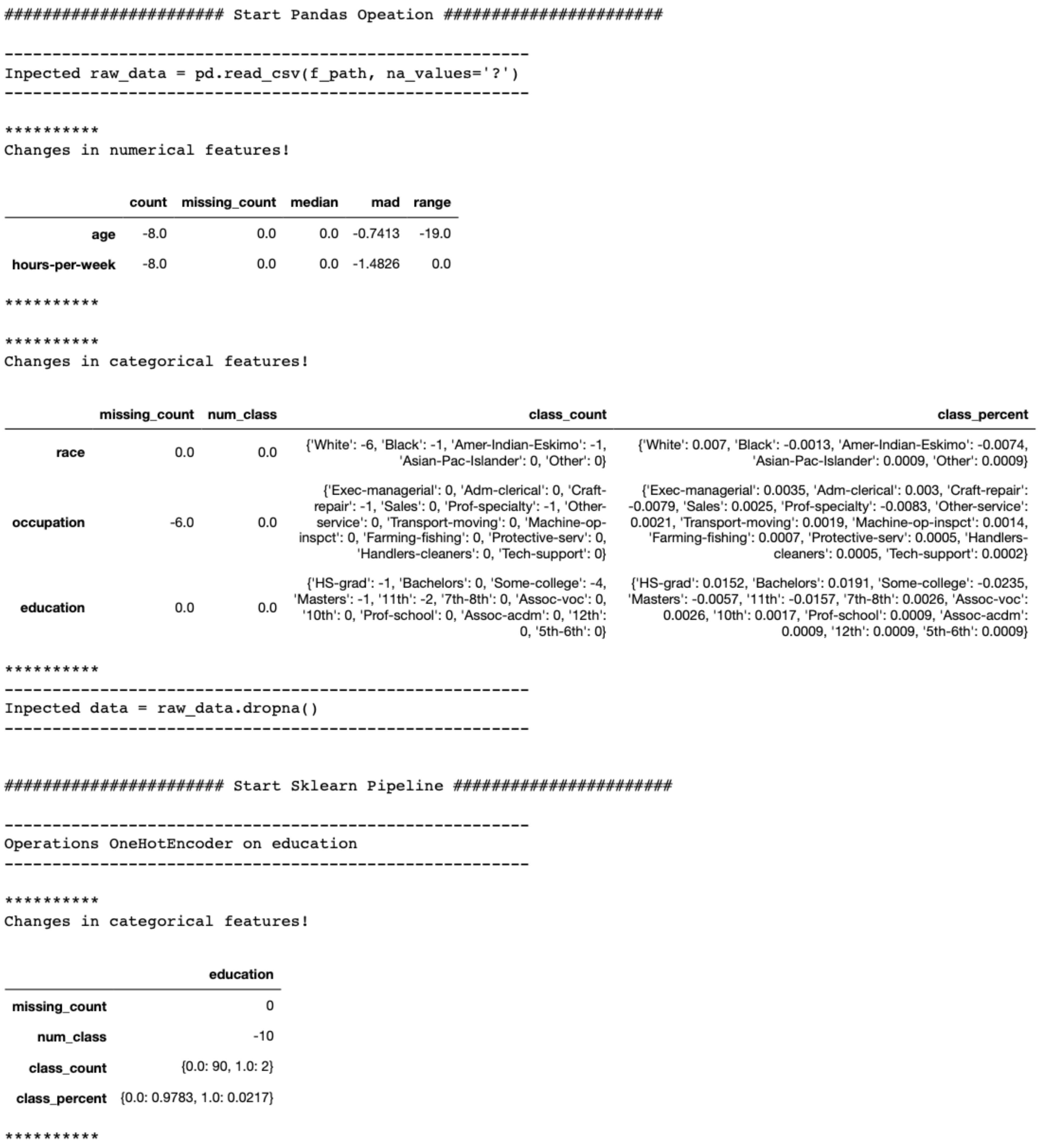
```
**********
--------------------------------------------------------
Operations StandardScaler on age
--------------------------------------------------------

**********
Changes in numerical features!
```

	age
count	0.0000
missing_count	0.0000
median	-33.7344
mad	-10.1331
range	-50.1208



```python
@tracer(cat_col = ['race', 'occupation', 'education'], numerical_col = ['age', 'hours-per-week'])
def adult_pipeline_easy(f_path = '../pipelines/adult-sample.csv'):

    raw_data = pd.read_csv(f_path, na_values='?')
    data = raw_data.dropna()

    labels = label_binarize(data['income-per-year'], ['>50K', '<=50K'])

    feature_transformation = ColumnTransformer(transformers=[
        ('categorical', OneHotEncoder(handle_unknown='ignore'), ['education', 'workclass']),
        ('numeric', StandardScaler(), ['age', 'hours-per-week'])
    ])


    income_pipeline = Pipeline([
      ('features', feature_transformation),
      ('classifier', DecisionTreeClassifier())])

    return income_pipeline
```





```
###################### Start Pandas Opeation ######################

-------------------------------------------------------
Inpected raw_data = pd.read_csv(f_path, na_values='?')
-------------------------------------------------------

**********
Changes in numerical features!
```

	count	missing_count	median	mad	range
age	-8.0	0.0	0.0	-0.7413	-19.0
hours-per-week	-8.0	0.0	0.0	-1.4826	0.0

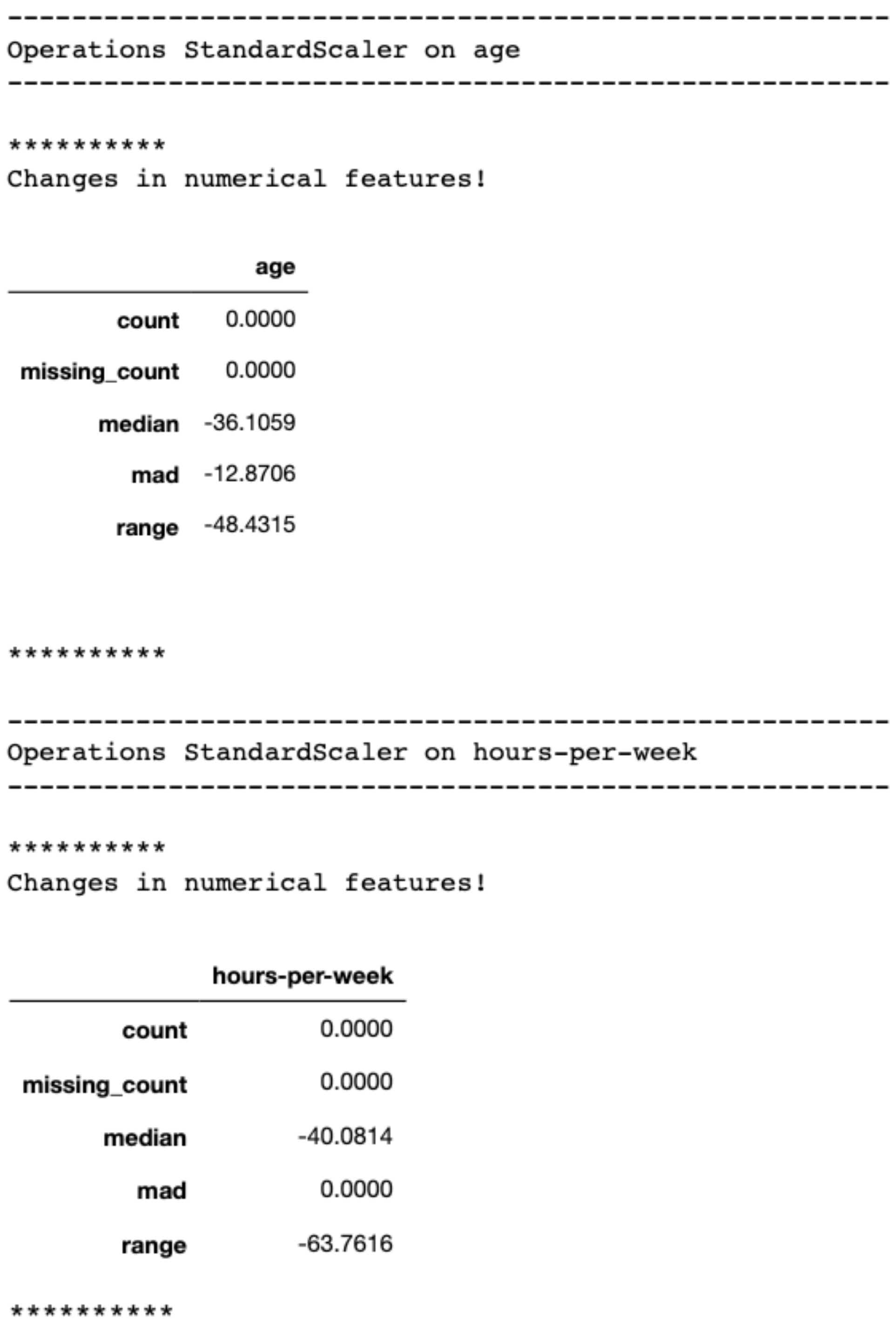
```
**********

**********
Changes in categorical features!
```

	missing_count	num_class	class_count	class_percent
race	0.0	0.0	{'White': -6, 'Black': -1, 'Amer-Indian-Eskimo': -1, 'Asian-Pac-Islander': 0, 'Other': 0}	{'White': 0.007, 'Black': -0.0013, 'Amer-Indian-Eskimo': -0.0074, 'Asian-Pac-Islander': 0.0009, 'Other': 0.0009}
occupation	-6.0	0.0	{'Exec-managerial': 0, 'Adm-clerical': 0, 'Craft-repair': -1, 'Sales': 0, 'Prof-specialty': -1, 'Other-service': 0, 'Transport-moving': 0, 'Machine-op-inspct': 0, 'Farming-fishing': 0, 'Protective-serv': 0, 'Handlers-cleaners': 0, 'Tech-support': 0}	{'Exec-managerial': 0.0035, 'Adm-clerical': 0.003, 'Craft-repair': -0.0079, 'Sales': 0.0025, 'Prof-specialty': -0.0083, 'Other-service': 0.0021, 'Transport-moving': 0.0019, 'Machine-op-inspct': 0.0014, 'Farming-fishing': 0.0007, 'Protective-serv': 0.0005, 'Handlers-cleaners': 0.0005, 'Tech-support': 0.0002}
education	0.0	0.0	{'HS-grad': -1, 'Bachelors': 0, 'Some-college': -4, 'Masters': -1, '11th': -2, '7th-8th': 0, 'Assoc-voc': 0, '10th': 0, 'Prof-school': 0, 'Assoc-acdm': 0, '12th': 0, '5th-6th': 0}	{'HS-grad': 0.0152, 'Bachelors': 0.0191, 'Some-college': -0.0235, 'Masters': -0.0057, '11th': -0.0157, '7th-8th': 0.0026, 'Assoc-voc': 0.0026, '10th': 0.0017, 'Prof-school': 0.0009, 'Assoc-acdm': 0.0009, '12th': 0.0009, '5th-6th': 0.0009}

```
**********
-------------------------------------------------------
Inpected data = raw_data.dropna()
-------------------------------------------------------

###################### Start Sklearn Pipeline ######################

-------------------------------------------------------
Operations OneHotEncoder on education
-------------------------------------------------------

**********
Changes in categorical features!
```

	education
missing_count	0
num_class	-10
class_count	{0.0: 90, 1.0: 2}
class_percent	{0.0: 0.9783, 1.0: 0.0217}

```
**********
```



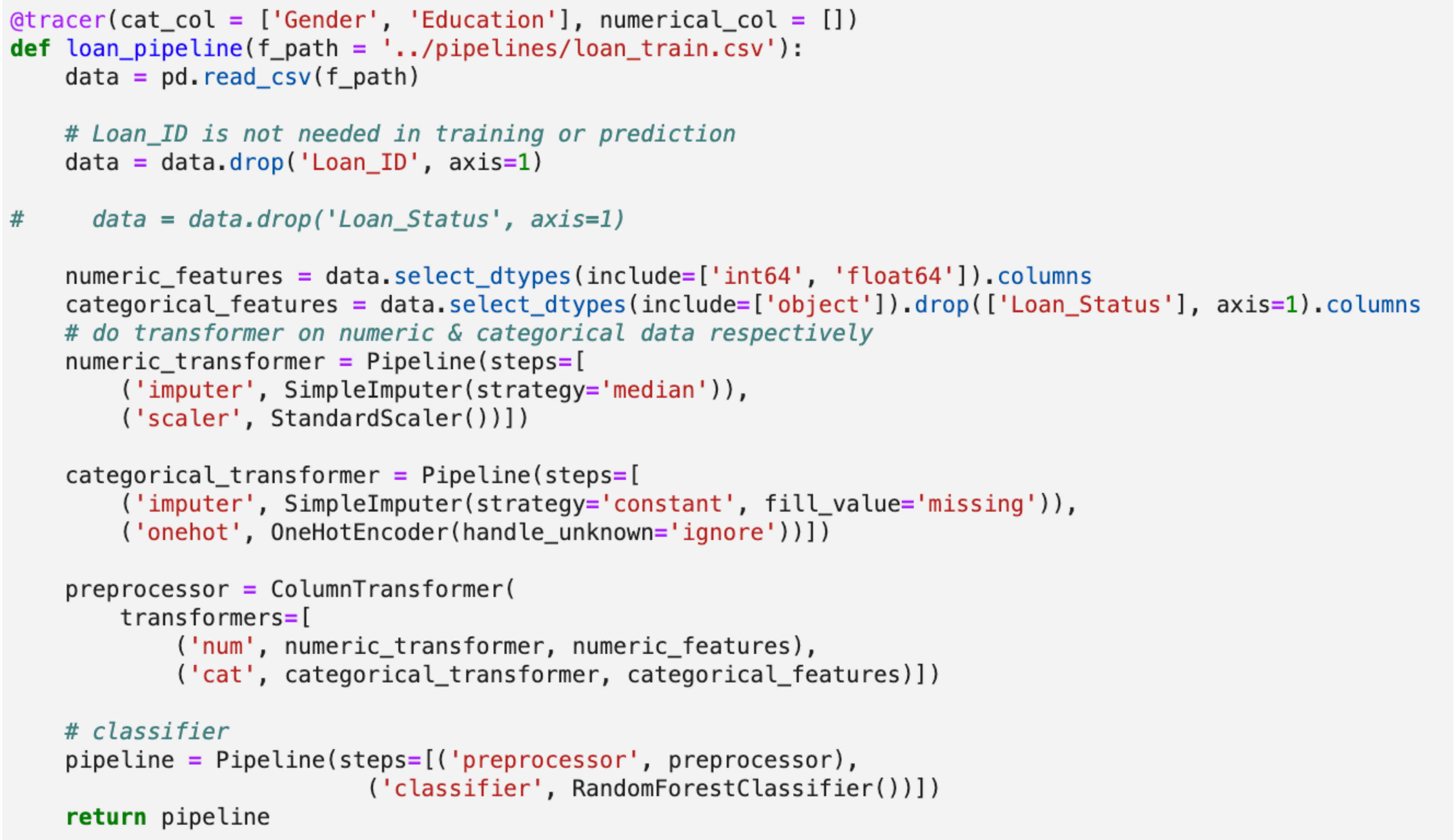
```
-------------------------------------------------------
Operations StandardScaler on age
-------------------------------------------------------

**********
Changes in numerical features!
```

	age
count	0.0000
missing_count	0.0000
median	-36.1059
mad	-12.8706
range	-48.4315

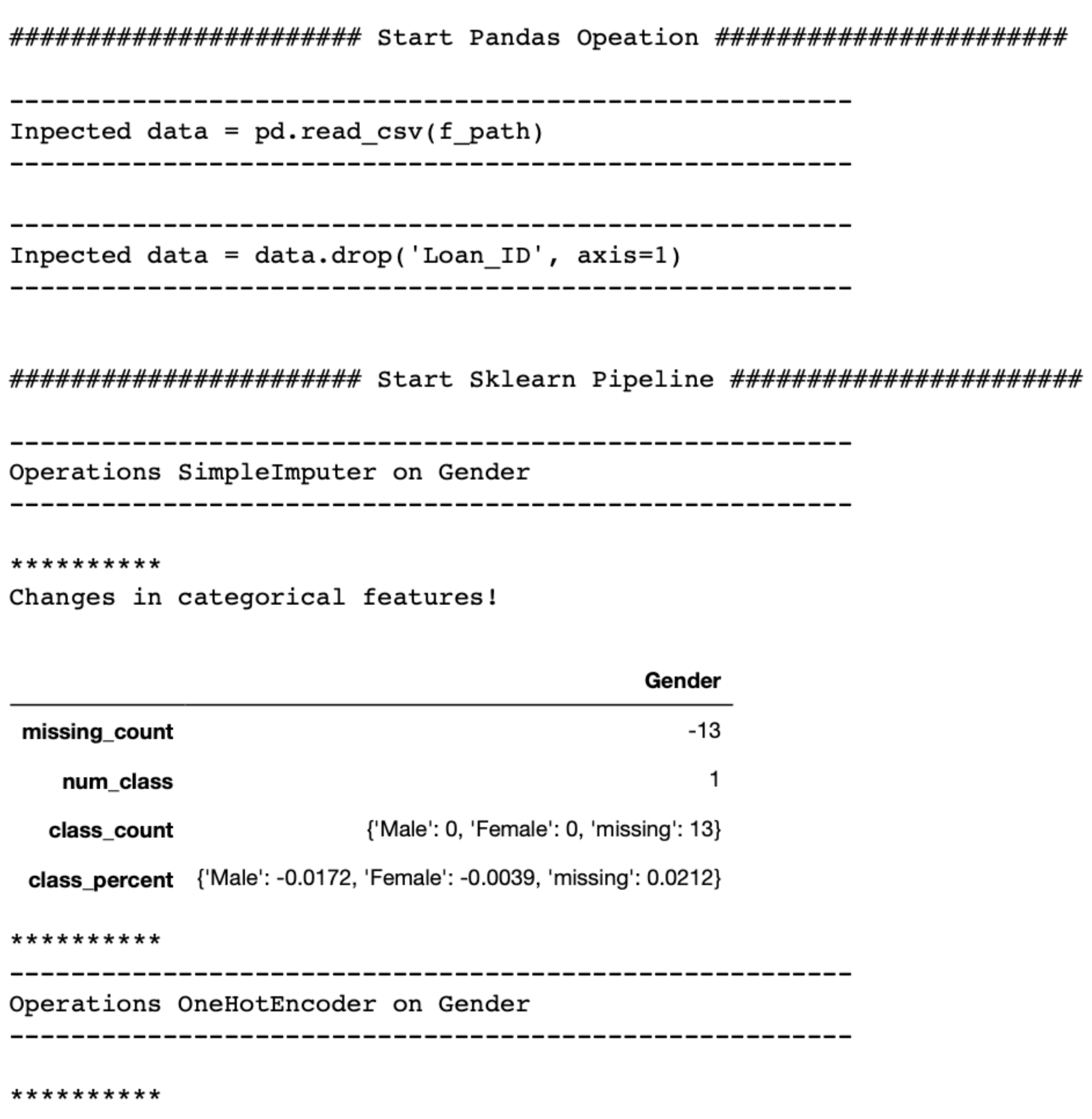
```
**********

-------------------------------------------------------
Operations StandardScaler on hours-per-week
-------------------------------------------------------

**********
Changes in numerical features!
```

	hours-per-week
count	0.0000
missing_count	0.0000
median	-40.0814
mad	0.0000
range	-63.7616

```
**********
```



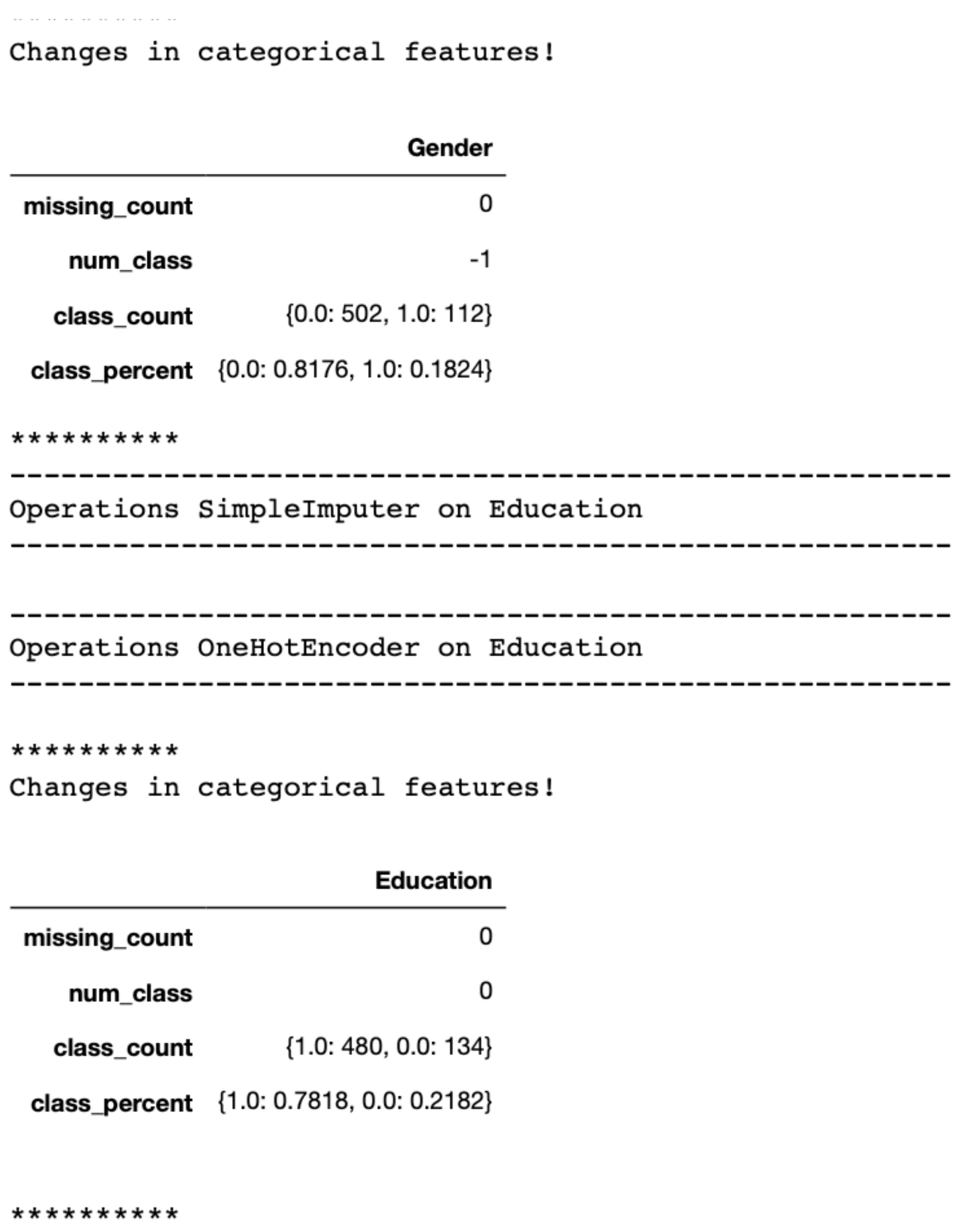
```python
@tracer(cat_col = ['Gender', 'Education'], numerical_col = [])
def loan_pipeline(f_path = '../pipelines/loan_train.csv'):
    data = pd.read_csv(f_path)

    # 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
```



```
###################### Start Pandas Opeation ######################

--------------------------------------------------------
Inpected data = pd.read_csv(f_path)
--------------------------------------------------------

--------------------------------------------------------
Inpected data = data.drop('Loan_ID', axis=1)
--------------------------------------------------------


###################### Start Sklearn Pipeline ######################

--------------------------------------------------------
Operations SimpleImputer on Gender
--------------------------------------------------------

**********
Changes in categorical features!
```

	Gender
missing_count	-13
num_class	1
class_count	{'Male': 0, 'Female': 0, 'missing': 13}
class_percent	{'Male': -0.0172, 'Female': -0.0039, 'missing': 0.0212}

```
**********
--------------------------------------------------------
Operations OneHotEncoder on Gender
--------------------------------------------------------

**********
```



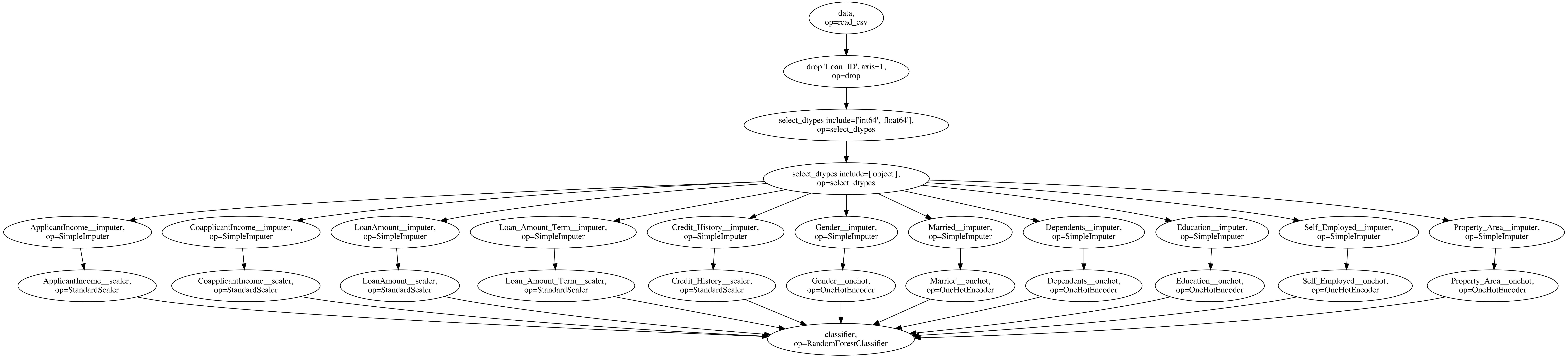
```
Changes in categorical features!
```

	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}

```
**********
--------------------------------------------------------
Operations SimpleImputer on Education
--------------------------------------------------------

--------------------------------------------------------
Operations OneHotEncoder on Education
--------------------------------------------------------

**********
Changes in categorical features!
```

	Education
missing_count	0
num_class	0
class_count	{1.0: 480, 0.0: 134}
class_percent	{1.0: 0.7818, 0.0: 0.2182}

```
**********
```


data,
op=read_csv
drop 'Loan_ID', axis=1,
op=drop
select_dtypes include=['int64', 'float64'],
op=select_dtypes
select_dtypes include=['object'],
op=select_dtypes
ApplicantIncome__imputer,
op=SimpleImputer
CoapplicantIncome__imputer,
op=SimpleImputer
LoanAmount__imputer,
op=SimpleImputer
Loan_Amount_Term__imputer,
op=SimpleImputer
Credit_History__imputer,
op=SimpleImputer
Gender__imputer,
op=SimpleImputer
Married__imputer,
op=SimpleImputer
Dependents__imputer,
op=SimpleImputer
Education__imputer,
op=SimpleImputer
Self_Employed__imputer,
op=SimpleImputer
Property_Area__imputer,
op=SimpleImputer
ApplicantIncome__scaler,
op=StandardScaler
CoapplicantIncome__scaler,
op=StandardScaler
LoanAmount__scaler,
op=StandardScaler
Loan_Amount_Term__scaler,
op=StandardScaler
Credit_History__scaler,
op=StandardScaler
Gender__onehot,
op=OneHotEncoder
Married__onehot,
op=OneHotEncoder
Dependents__onehot,
op=OneHotEncoder
Education__onehot,
op=OneHotEncoder
Self_Employed__onehot,
op=OneHotEncoder
Property_Area__onehot,
op=OneHotEncoder
classifier,
op=RandomForestClassifier