Fairness-Aware Instrumentation of ML Pipelines

Team 6

Biao Huang
Chenqin Yang
Rui Jiang
Zhengyuan Ding

Contents

Problem Statement

When implementing an end-to-end ML system in the real world, the data preparation stage is crucial to how the final model is produced and performs.

However, unintentional biases may occur in any operations within this stage if approached without care.

Goal: track potential fairness problems raised in data preparation operations

Approach

- End-to-End machine learning pipeline as DAGs
  - help understand the pipeline structure and dependencies between operations
  - common tools for data pipelines: Pandas and Scikit-Learn
  - DAG visualization
- Logs of changes
  - Categorical features: # missing values, # classes/levels, # records for each class/level and proportion for each class/level
  - Numerical features: # records, # missing values, Median and Median Absolute Deviation and Range/Scaling

Implementation -- Environment Setup

- MacOS Mojave/Catalina Version
- Python 3.6.5
- Pandas==0.23.0, Numpy==1.14.3, Scikit-learn==0.21.2, Inspect==Python3.6
- Pandas==0.25.1, Numpy==1.17, Scikit-learn==0.21.3

Implementation -- Datasets

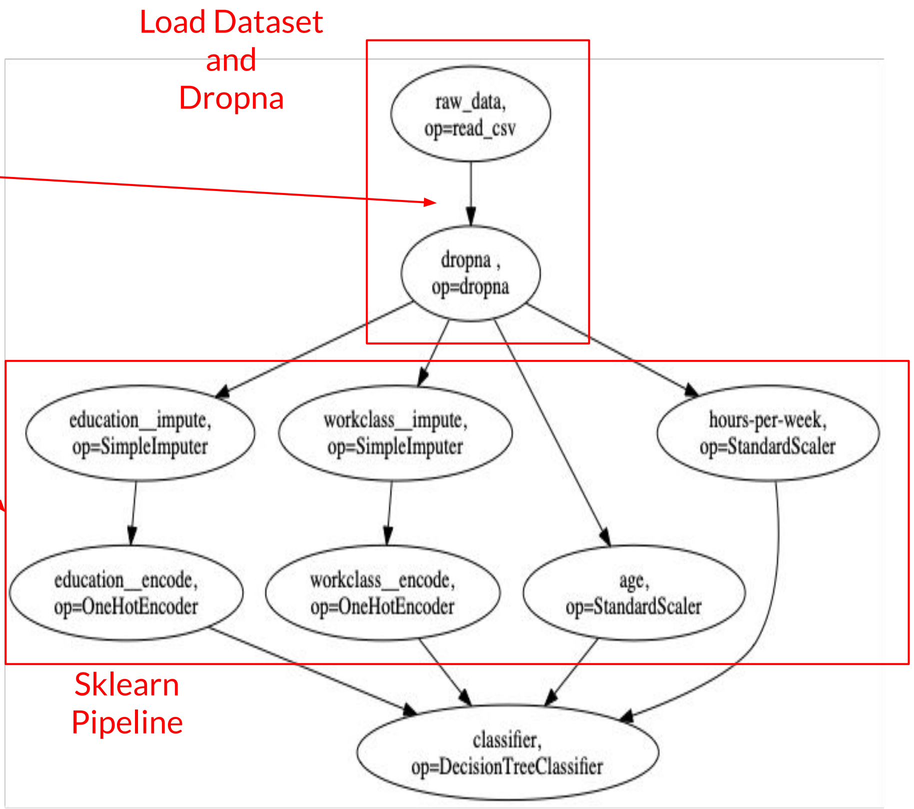
- Adult-sample.csv[1]
  - From Assignment3
  - Pipelines: Adult Pipeline Easy & **Adult Pipeline Hard**
- Loan dataset[2]
  - Loan eligibility prediction contest held by Analytics Vidhya
  - Pipeline : Loan Pipeline
- Statlog (German Credit - SCHUFA)[3]
  - 1,000 credit applicants in Germany with their credit score.
  - Pipeline: GermanCredit Pipeline
- COMPAS Recidivism Racial Bias[4]
  - Widely used in fairness-related researches
  - Pipeline: **Compas Pipeline**

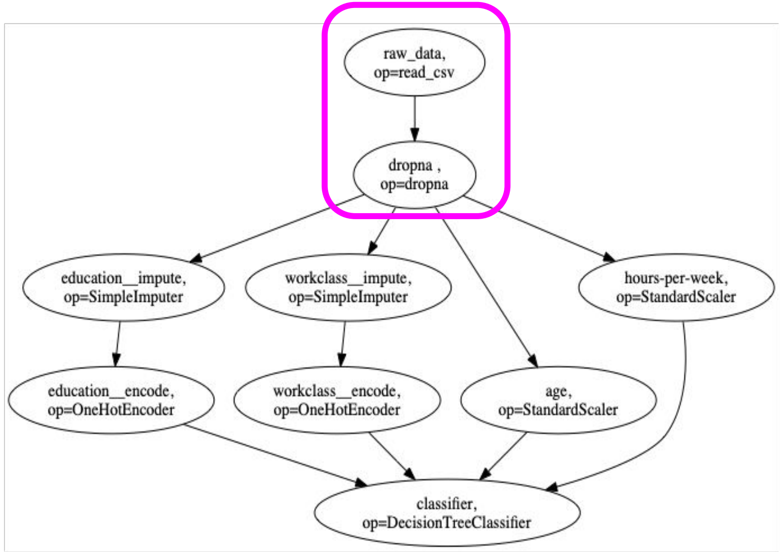
Case 1: Adult Sample Dataset

Targeted numerical features: ["age" , "hours-per-week"]

Targeted categorical features: ["race", "occupation", "education"]

```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
```





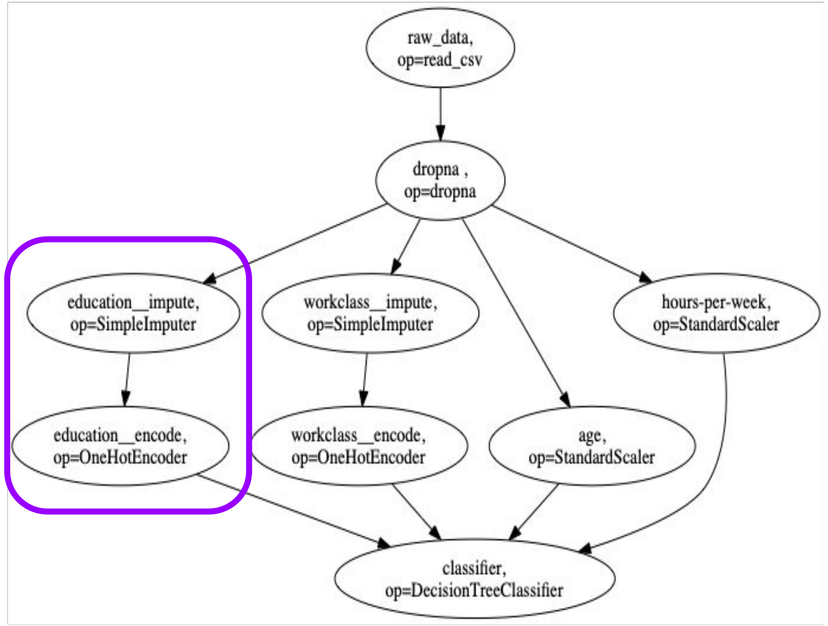

```
###################### Start Pandas Opeation ######################

-----------------------------------------------------
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

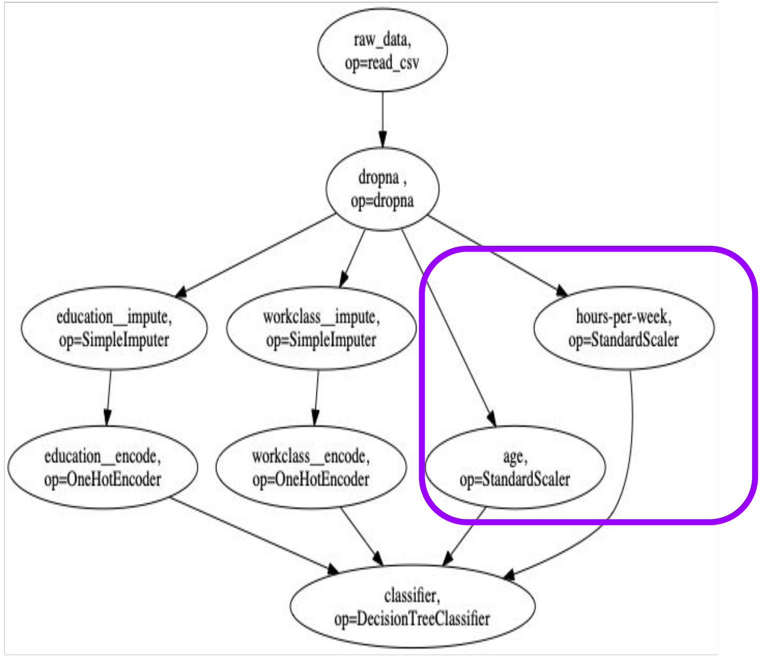
```
**********

**********
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
-----------------------------------------------------
```




```
------------------------------------------------------
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

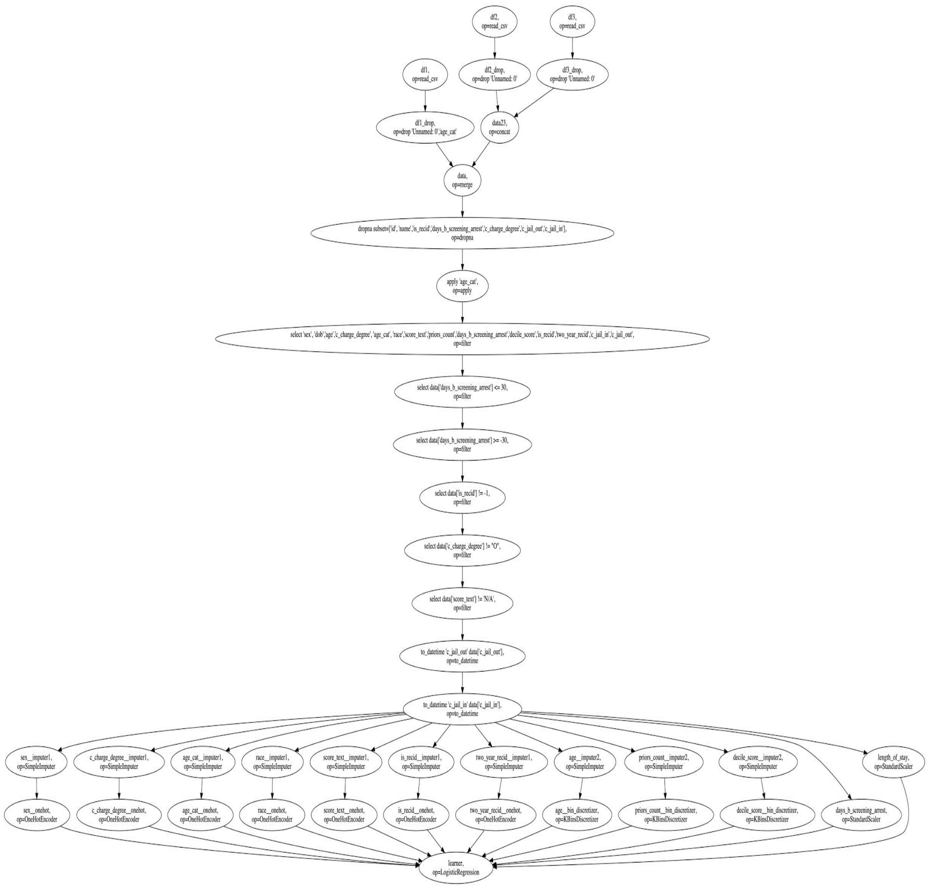
```
**********
```

Case2: COMPAS Recidivism Racial Bias

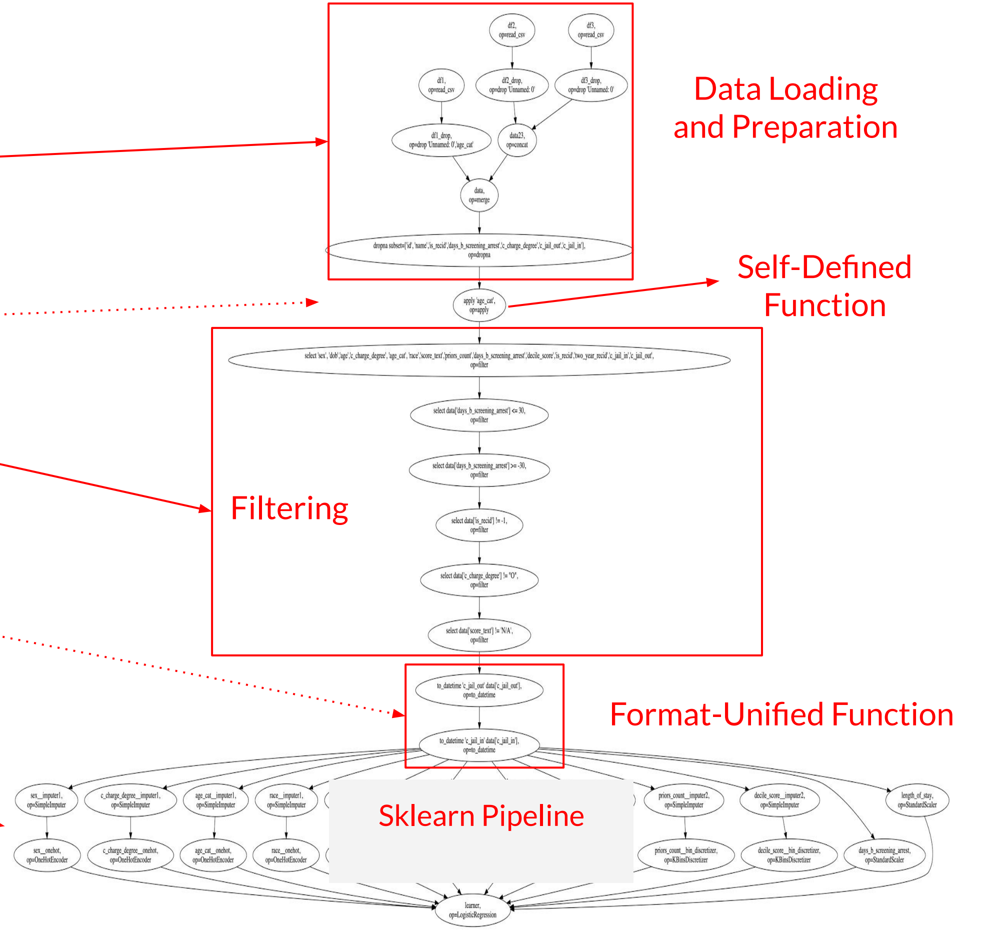
Targeted numerical features: ["age"]

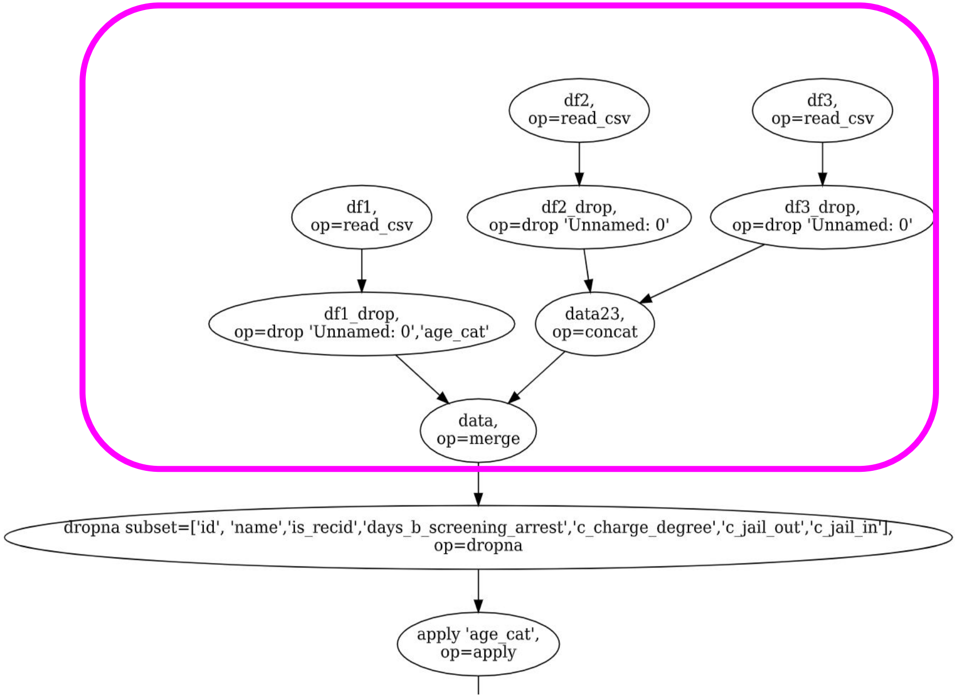
Targeted categorical features: ["race"]

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

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

    #JOIN dataframes column-wise and row-wise
    data = pd.concat([df1,df2],ignore_index=True)
    data = pd.merge(df, data, 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
```

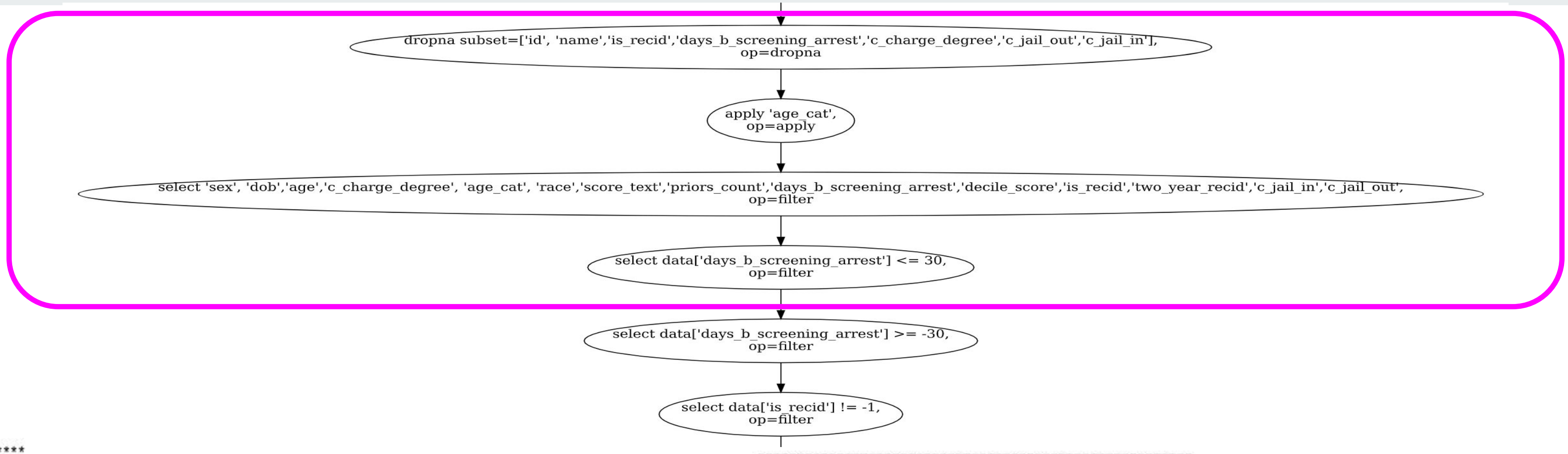


```python
@tracer(cat_col = ['race'], numerical_col = ['age'])
def compass_pipeline(f1_path = '../data/compass/demographic.csv',f2_path = '../data/compass/jailrecord1.csv',f3_path = '../data/compass/jailrecord2.csv'):
    #read csv files
    df = pd.read_csv(f1_path)
    df1 = pd.read_csv(f2_path)
 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

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

    #JOIN dataframes column-wise and row-wise
    data = pd.concat([df1,df2],ignore_index=True)
    data = pd.merge(df, data, 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_percei
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': -0.003 'Other': -0.0001, 'Asian': 0.0002, 'Native American': -0.000

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

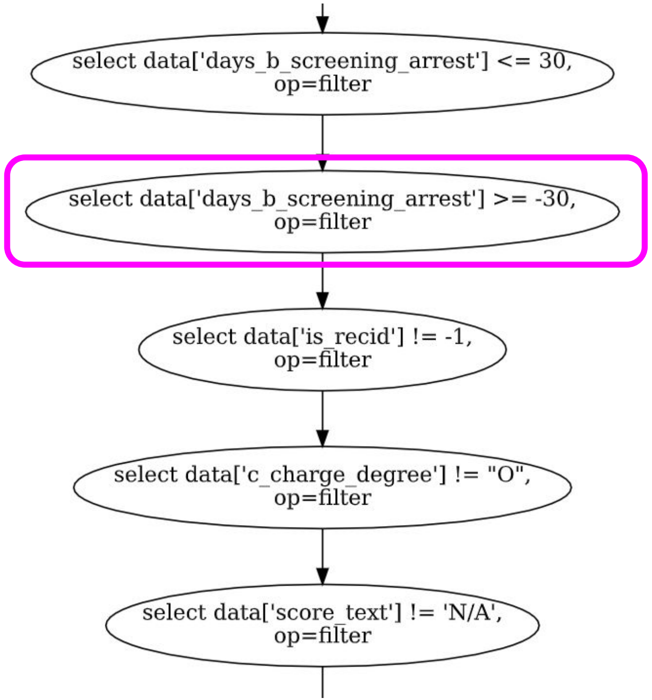
```
------------------------------------------------------------
Inpected data = data[['sex', 'dob','age','c_charge_degree', 'age_cat', 'race','score_text','priors_count','d
ays_b_screening_arrest','decile_score','is_recid','two_year_recid','c_jail_in','c_jail_out']]
------------------------------------------------------------

**********
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

****

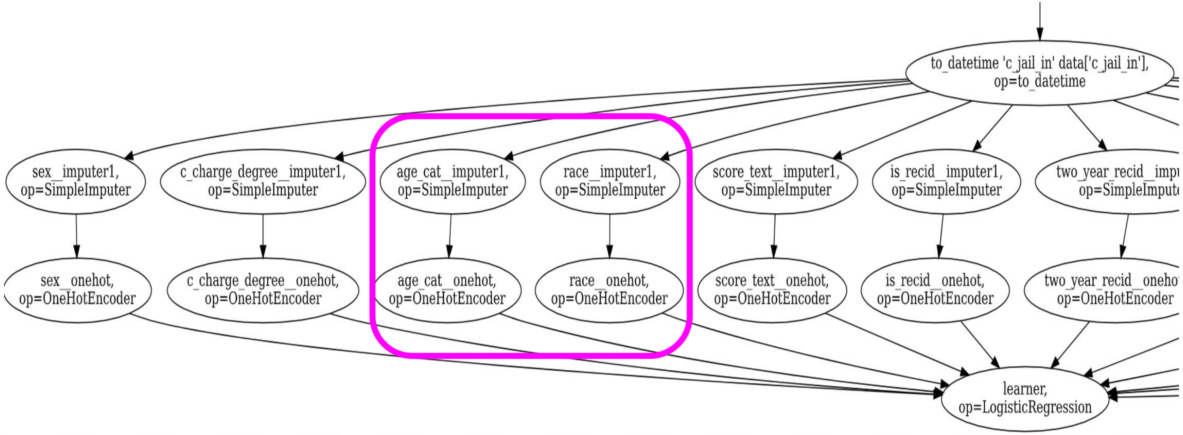
****

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
------------------------------------------------------------
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

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

Conclusion

- Our Data Preprocessing Function Tracer is robust to a variety of Pandas and Sklearn operations
    - Easy to trace operations through DAG — both big-pictured and detailed-oriented
    - More customized evaluation metrices can be added by users

- Future Work:
    - Infer data schema
    - User-Defined start and stop point
    - Interactive GUI + Visualization

Reference

[1] https://github.com/schelterlabs/deml-lab/blob/master/assignment3/adult-sample.csv
[2] https://datahack.analyticsvidhya.com/contest/practice-problem-loan-prediction-iii/
[3] http://www.fairness-measures.org/Pages/Datasets/Schufa.html
[4] https://www.kaggle.com/danofer/compass
[5] Hajian, Sara, Francesco Bonchi, and Carlos Castillo. "Algorithmic bias: From discrimination discovery to fairness-aware data mining." Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. ACM, 2016
[6] S. Barocas and A. D. Selbst. Big data's disparate impact. Calif. L. Rev., 104:671, 2016.
[7] C. O'Neil. Weapons of math destruction: How big data increases inequality and threatens democracy. Broadway Books, 2017.