In [1]:
```
import fairdetect as fd
```

In [2]:
```
import dalex as dx
import pandas as pd
data = dx.datasets.load_german()
```

In [3]:
```
data
```

Out[3]:

	risk	sex	job	housing	saving_accounts	checking_account	credit_amount	duration	purpose	age
0	1	male	2	own	not_known	little	1169	6	radio/TV	67
1	0	female	2	own	little	moderate	5951	48	radio/TV	22
2	1	male	1	own	little	not_known	2096	12	education	49
3	1	male	2	free	little	little	7882	42	furniture/equipment	45
4	0	male	2	free	little	little	4870	24	car	53
...	...	...	...	...	...	...	...	...	...	...
995	1	female	1	own	little	not_known	1736	12	furniture/equipment	31
996	1	male	3	own	little	little	3857	30	car	40
997	1	male	2	own	little	not_known	804	12	radio/TV	38
998	0	male	2	free	little	little	1845	45	radio/TV	23
999	1	male	2	own	moderate	moderate	4576	45	car	27

1000 rows × 10 columns

In [4]:
```
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
data['sex']=le.fit_transform(data.sex)
data['housing']=le.fit_transform(data.housing)
data['saving_accounts']=le.fit_transform(data.saving_accounts)
data['checking_account']=le.fit_transform(data.checking_account)
data['purpose']=le.fit_transform(data.purpose)
```

In [5]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer

X = data.drop(["risk"], axis=1)
y = data["risk"]

X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.8, test_size=0.2, random_state=0)

print("Data successfully loaded!\n")
```

```
Data successfully loaded!
```

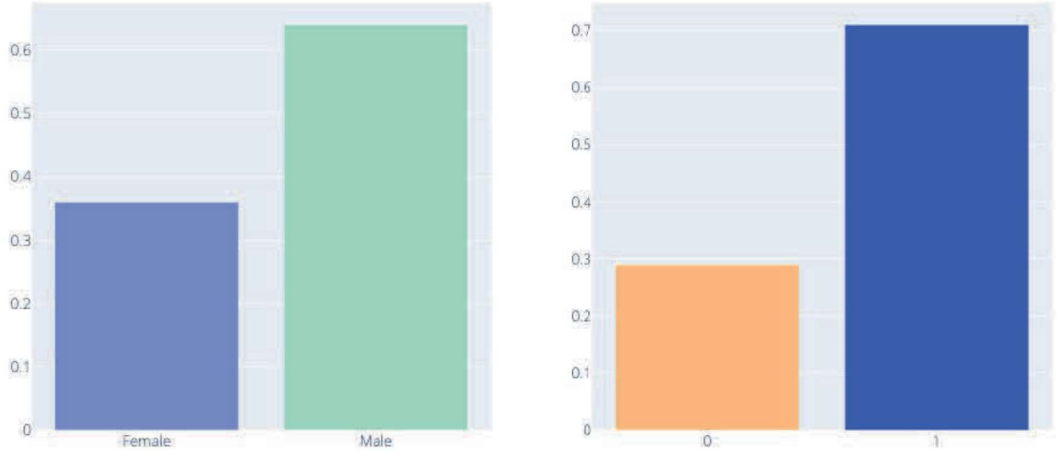
In [6]:
```
import xgboost

model = xgboost.XGBClassifier().fit(X_train, y_train)

predict = model.predict(X_test)
```

```
[10:03:55] WARNING: ../src/learner.cc:1115: Starting in XGBoost 1.3.0, the default evaluation metric used with the ob
jective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore
the old behavior.
```

```
/opt/anaconda3/lib/python3.8/site-packages/xgboost/sklearn.py:1224: UserWarning: The use of label encoder in XGBClass
ifier is deprecated and will be removed in a future release. To remove this warning, do the following: 1) Pass option
use_label_encoder=False when constructing XGBClassifier object; and 2) Encode your labels (y) as integers starting wi
th 0, i.e. 0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)
```

In [8]:
```
fd.identify_bias(model,X_test,y_test,'sex',labels)
```

```
REPRESENTATION
```




```
Demographic Parity
```

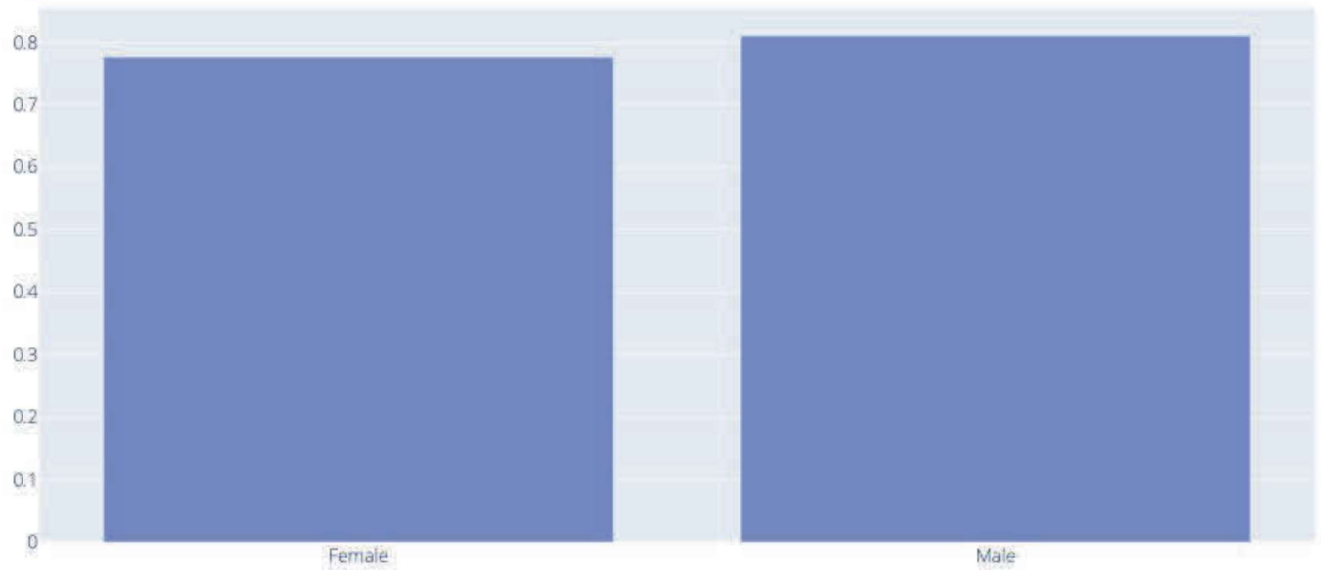
	Female	Male
0	0.347222	0.257812
1	0.652778	0.742188

```
Accept H0: No Significant Relation Between sex and Target Detected. p= 0.23990131169226664
```

ABILITY


```
Equalized Odds
Accept H0: True Positive Disparity Not Detected. p= 0.43593930807521697
Accept H0: False Positive Disparity Not Detected. p= 0.5942718834339058
Accept H0: True Negative Disparity Not Detected. p= 0.5576362423438422
** Reject H0: Significant False Negative Disparity with p= 0.016167046868271105
```

PREDICTIVE

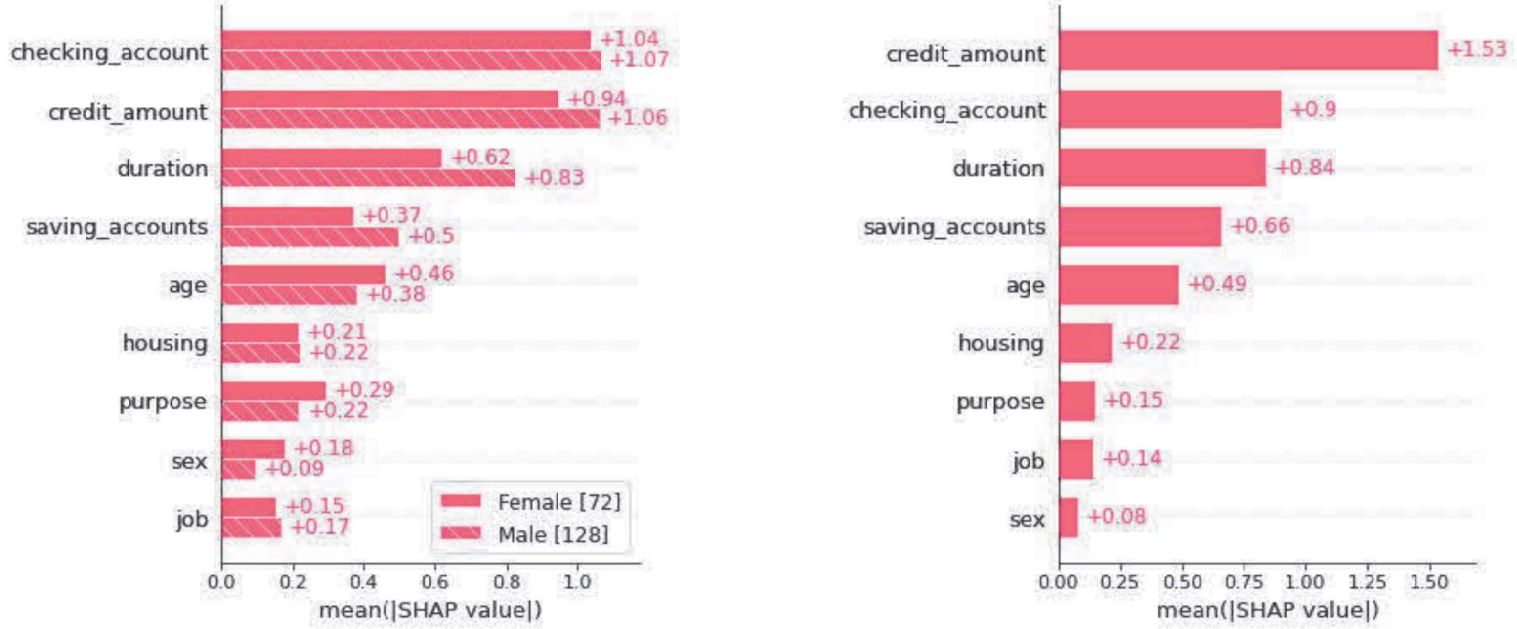


Predictive Parity
Accept H0: No Significant Predictive Disparity. p= 0.786326136425535

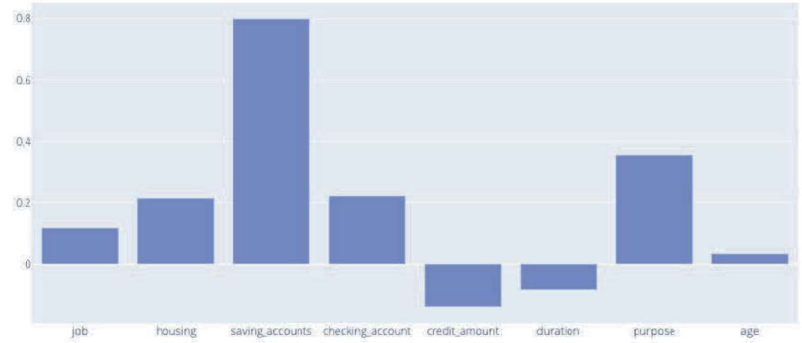
```
In [9]: fd.understand_shap(X_test,y_test,model,labels,'sex',1,0)
```

```
ntree_limit is deprecated, use `iteration_range` or model slicing instead.
```

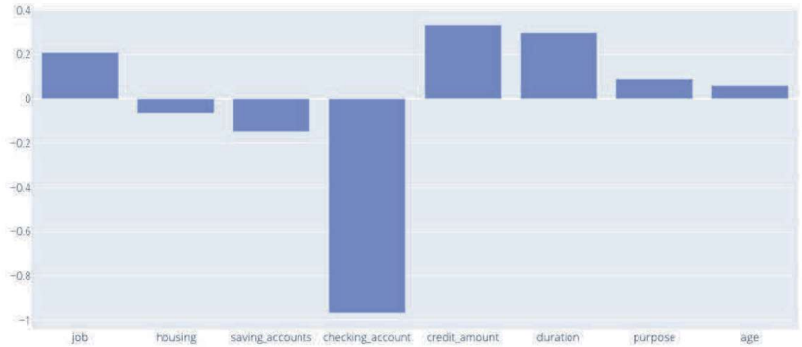
```
Model Importance Comparison
```

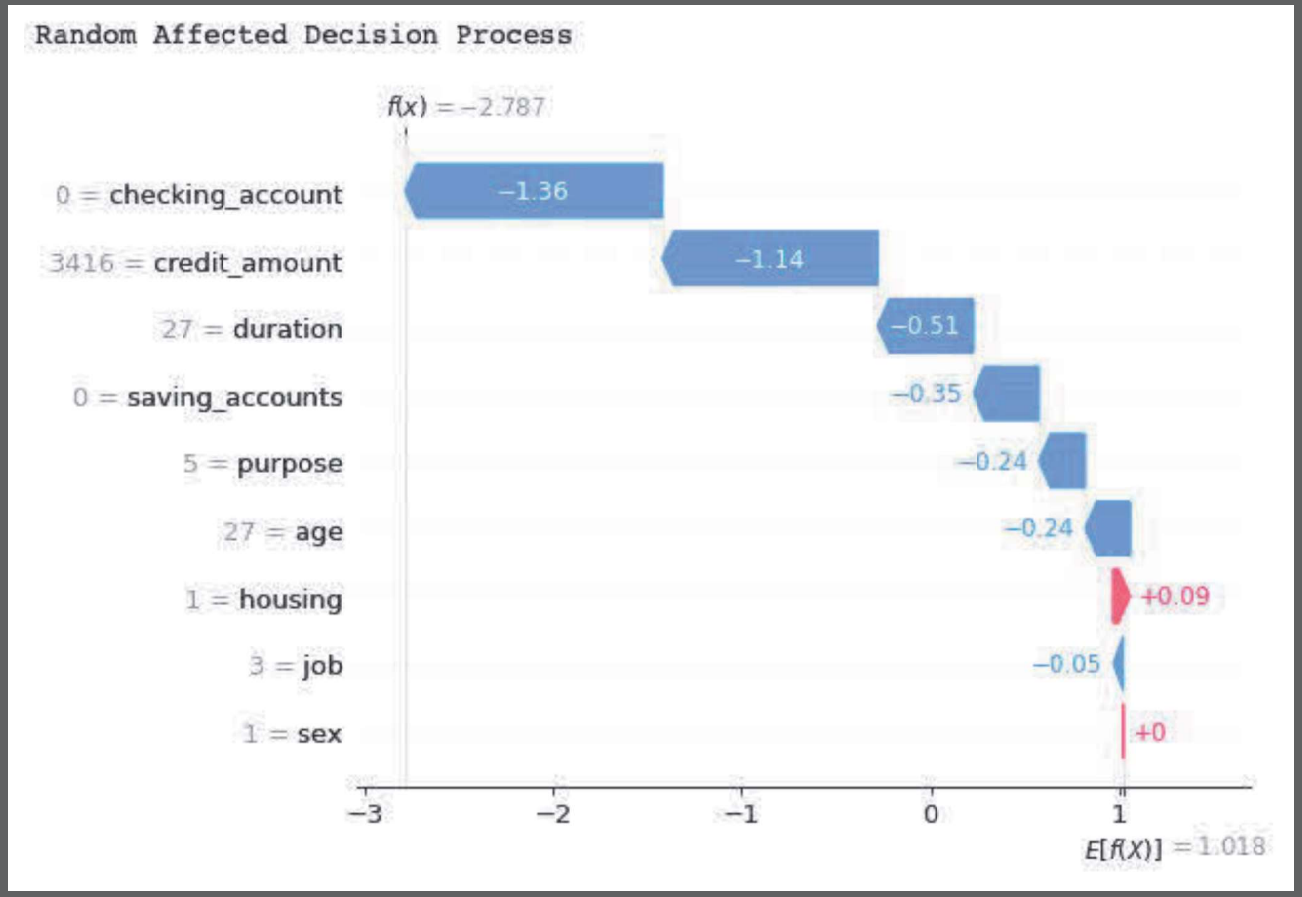


Affected Attribute Comparison
Average Comparison to True Class Members



Average Comparison to All Members




Random Affected Decision Process
f(x) = −2.787
0 = checking_account
3416 = credit_amount
27 = duration
0 = saving_accounts
5 = purpose
27 = age
1 = housing
3 = job
1 = sex
−1.36
−1.14
−0.51
−0.35
−0.24
−0.24
+0.09
−0.05
+0
−3
−2
−1
0
1
E[f(X)] = 1.018