CS 584- Data Mining and Applications
Kaustubh Prashant Karanjkar
Identifier: **Kaustubh**
#G01314326

HW4: Fairness and Classification

On Miner:
Rank: 71
Accuracy: 86%

Aim:
- **Part1- Prediction:** Explore different classes of classifiers and check the best accuracy and f-1 score.
- **Part2- Fairness Diagnosis:** For each of the classifiers presented in part 1, report their demographic disparity, inequality of odds, and opportunity for two sensitive attributes: gender and race.
- **Part3- Fairness Mitigation:** Remove the sensitive and proxy attributes from the features and report the new accuracy, F1 score, demographic disparity, equality of odds, and opportunity.

Approach:

- For part 1, I have tried and tested multiple classifiers- XGBoost classifier, Decision Tree classifier, KNN classifier, Random Forest classifier, Logistic Regression classifier, and SVM classifier. Initially, as a part of the pre-processing, I removed the null values from the training data and then converted the values from the income column into 1 and 0, the field with data having '>50k' was converted to 1, and the rest to 0. Then, I have encoded the data by using the LabelEncoder library this has helped me in eliminating the string values. Then, for validation purpose I used Train Test split library and divided the training data into an 80:20 train-test data ratio.

- Below is the accuracy and f1 score I got from the classifiers:

1. XGBoost Classifier

Accuracy	F1 Score
• 86.2%	• 68.3%

2. Decision Tree Classifier

Accuracy	F1 Score
• 80.59%	• 61.1%

3. KNN Classifier

Accuracy	F1 Score
• 81.8%	• 62.5%

4. Random Forest Classifier

Accuracy	F1 Score
• 84.4%	• 66.8%

5. Logistic Regression Classifier

Accuracy	F1 Score
• 81.9%	• 54.5%

6. SVM Classifier

Accuracy	F1 Score
• 50.5%	• 48.4%

- From the above data, we could see that XGBoost and Random Forest classifier performed well with 86% and 84% accuracy and an F1 score of 68.3% and 66.8% respectively.
- Later, for part 2, where we have to check the demographic disparity, inequality of odds, and opportunity for two sensitive attributes: gender and race. These two attributes are called sensitive because the data in these columns is very one-sided, which means there are 5208 White values for the race attribute and the remaining 825 are non-white records. Similarly, for the sex attribute, there are almost double records with the value Male as compared to Females, so these attributes play a big part in making a biased judgment while making a prediction.
- As a part of pre-processing, I have converted the values from these attributes into the form of 1 and 0, I have kept 1 for White and 0 for non-white records, similarly for sex attribute, I have kept 1 for Male and 0 for female.
- I have created functions to calculate Demographic Parity, inequality of odds, and opportunity.
- Below are the results I got for the classifiers:

1. *Demographic Parity:*

```
Demographic Parity(Race): XGBoost-  0.0808930316994833
Demographic Parity(Sex): XGBoost-  0.17004629532128607


Demographic Parity(Race): Decision Tree-  0.09649211004049715
Demographic Parity(Sex): Decsion Tree-  0.1968020673704084


Demographic Parity(Race): KNN-  0.11086999022482893
Demographic Parity(Sex): KNN-  0.1986312095132735


Demographic Parity(Race): Random Forest-  0.11240678676162548
Demographic Parity(Sex): Random Forest-  0.19400576554523052


Demographic Parity(Race): Logistic Regression-  0.004891076665270222
Demographic Parity(Sex): Logistic Regression-  0.1784786839591729


Demographic Parity(Race): SVM-  0.21753107107945813
Demographic Parity(Sex): SVM-  0.3871296033617068
```

From the above, we can see that XGBoost and Linear Regression performed well for both the attributes sex and race.

2. *Equality of Opportunity*

```
Demographic Parity(Race): XGBoost-  0.0808930316994833
Demographic Parity(Sex): XGBoost-  0.17004629532128607


Demographic Parity(Race): Decision Tree-  0.09649211004049715
Demographic Parity(Sex): Decsion Tree-  0.1968020673704084


Demographic Parity(Race): KNN-  0.11086999022482893
Demographic Parity(Sex): KNN-  0.1986312095132735


Demographic Parity(Race): Random Forest-  0.11240678676162548
Demographic Parity(Sex): Random Forest-  0.19400576554523052


Demographic Parity(Race): Logistic Regression-  0.004891076665270222
Demographic Parity(Sex): Logistic Regression-  0.1784786839591729


Demographic Parity(Race): SVM-  0.21753107107945813
Demographic Parity(Sex): SVM-  0.3871296033617068
```

This time also, we can see that XGBoost and Linear Regression classifier performed well (the lower the better score) for both the sensitive attributes of race and sex.

3. *Equality of Odds*

```
Equality of Odds(Race): XGBoost-  0.02267840630582912
Equality of Odds(Sex): XGBoost-  0.054750525408793674


Equality of Odds(Race): Decision Tree-  0.056994706239694534
Equality of Odds(Sex): Decsion Tree-  0.11019569697724164


Equality of Odds(Race): KNN-  0.054835232084556526
Equality of Odds(Sex): KNN-  0.09123521308855752


Equality of Odds(Race): Random Forest-  0.05872213593801902
Equality of Odds(Sex): Random Forest-  0.089175073614186


Equality of Odds(Race): Logistic Regression-  0.01154400344737811
Equality of Odds(Sex): Logistic Regression-  0.07307677599563238


Equality of Odds(Race): SVM-  0.2158546475265361
Equality of Odds(Sex): SVM-  0.3536184250673084
```

Here also XGBoost and Linear Regression classifiers performed well for both the sensitive attributes.

- Hence, we can say that the Sex and Race attribute plays an important role in biased prediction.
- Later, in part 3 to decrease the bias and increase the fairness of classification, I removed the sensitive attributes (race and sex) and then tried to classify the modified training dataset with XGBoost and Random Forest classifier, below are the results I got:
- *XGBoost Classifier:*

```
The accuracy of the XG Boost Classifier Model is 0.863583623404608
accuracy 0.863583623404608
f1 0.6852772466539196
```

```
After removing sensitive attributes

Demographic Parity(Race): XGBoost-  0.08006493506493506
Demographic Parity(Sex): XGBoost-  0.168548541951341


Equality of Opportunity(Race): XGBoost-  0.027673341158991227
Equality of Opportunity(Sex): XGBoost-  0.05546240541210434


Equality of Odds(Race): XGBoost-  0.020720926961668863
Equality of Odds(Sex): XGBoost-  0.055309496901247555
```

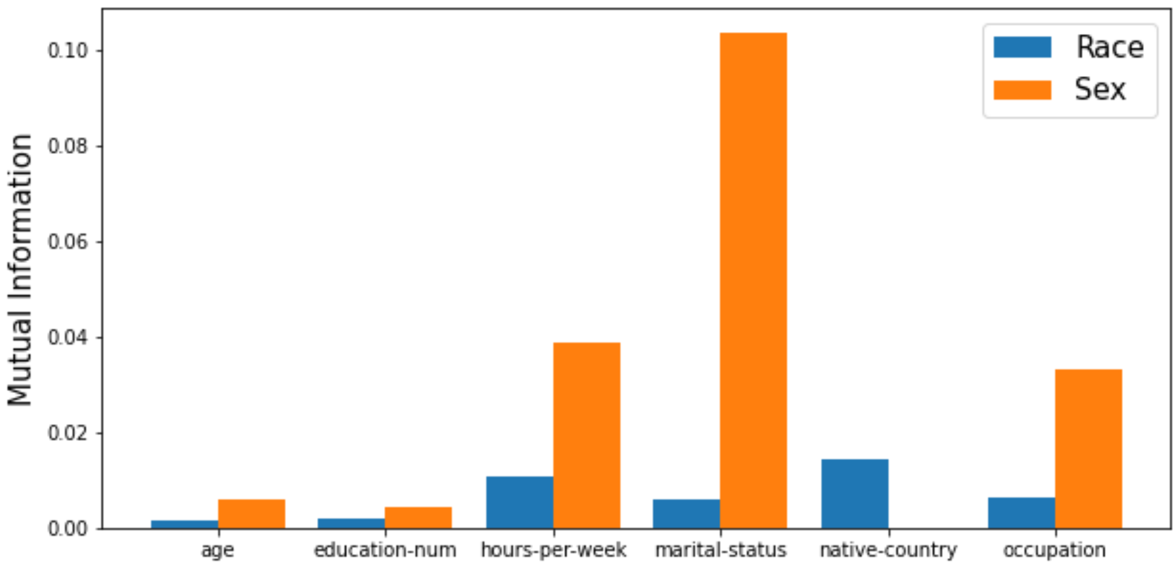
Random Forest Classifier

```
The accuracy of the Random Forest Model is 0.8428642466434609
accuracy 0.8428642466434609
f1 0.667601683029453
```

```
After removing sensitive attributes

Demographic Parity(Race): Random Forest-  0.0890643764837313
Demographic Parity(Sex): Random Forest-  0.18774580065385793


Equality of Opportunity(Race): Random Forest-  0.01773815802158296
Equality of Opportunity(Sex): Random Forest-  0.0446148687068979


Equality of Odds(Race): Random Forest-  0.03390146244243148
Equality of Odds(Sex): Random Forest-  0.08149206333708423
```

The accuracy and fairness didn’t change much, later I removed the proxy attributes along with the sensitive attributes. I used mutual_info_classif library from sklearn to find the mutual information related to sensitive attributes and found that marital status, occupation and hours-per-week are closely related. So to improve the fairness I removed these attributes and tried to classify them.

Below is the bar chart visualizing the results:



Then, I tried on XGBoost and Random Forest Classifier, and the fairness got improved but the accuracy got reduced to 85% from 86% which is the trade-off we got for removing the extra attributes (proxy and sensitive attributes).

XGBoost:

```
The accuracy of the XG Boost Classifier Model is 0.8567876678269518
accuracy 0.8567876678269518
f1 0.6601101494885916
```

```
After removing proxy and sensitive attributes

Demographic Parity(Race): XGBoost-  0.07166457198715265
Demographic Parity(Sex): XGBoost-  0.15192917819300825


Equality of Opportunity(Race): XGBoost-  0.02839352852112087
Equality of Opportunity(Sex): XGBoost-  0.03281234053100446


Equality of Odds(Race): XGBoost-  0.014858214605272216
Equality of Odds(Sex): XGBoost-  0.04764133557572843
```

Random Forest classifier:

```
The accuracy of the Random Forest Model is 0.8402121664180342
accuracy 0.8402121664180342
f1 0.6527377521613832
```

```
After removing sensitive attributes

Demographic Parity(Race): Random Forest-  0.09553344504957409
Demographic Parity(Sex): Random Forest-  0.15842377872397362


Equality of Opportunity(Race): Random Forest-  0.06043903075256751
Equality of Opportunity(Sex): Random Forest-  0.0021724233455319153


Equality of Odds(Race): Random Forest-  0.03627377927802566
Equality of Odds(Sex): Random Forest-  0.05950056263287777
```

The above results prove that the mitigation strategies have reduced the unfair outcomes since the Demographic Parity, inequality of Odds and opportunity is more tending toward zero after removing proxy and sensitive attributes.