Projekt 1

Gabriela Matuszewska, Mateusz Wojtulewicz

1 Opis datasetu

Do analizy wybrano dataset 'Adult income', który zawiera klasyfikację przychodu rocznego ($< 50k$, $>= 50k$) w zależności, od między innymi: pochodzenia, wykształcenia, miejsca zamieszkania.

Liczba cech i ich typy oraz rozmiar:

```
df.shape
```

```
(48842, 15)
```

```
df.head()
```

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hours-per-week	native-country	income
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Male	0	0	40	United-States	0
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	0	0	50	United-States	0
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	Male	0	0	40	United-States	1
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	7688	0	40	United-States	1
4	18	Private	103497	Some-college	10	Never-married	Prof-specialty	Own-child	White	Female	0	0	30	United-States	0

```
df.info()
```

```
Data columns (total 14 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   age              48842 non-null  int64
 1   workclass        48842 non-null  object
 2   fnlwgt           48842 non-null  int64
 3   education        48842 non-null  object
 4   educational-num  48842 non-null  int64
 5   marital-status   48842 non-null  object
 6   occupation       48842 non-null  object
 7   relationship     48842 non-null  object
 8   race             48842 non-null  object
 9   gender           48842 non-null  object
 10  capital-gain     48842 non-null  int64
 11  capital-loss     48842 non-null  int64
 12  hours-per-week   48842 non-null  int64
 13  income           48842 non-null  int64
dtypes: int64(7), object(7)
memory usage: 5.2+ MB
```

2 Czyszczenie danych

Ze względu na specyfikację zestawu danych konieczna była pewna modyfikacja tych danych. Brakujące wartości zostały zastąpione przez najczęściej występujące wartości. Ze względu na przeważającą liczbę native-country="United-States", która mogłaby zakłócić wyniki postanowiono usunąć kolumnę native-country. Kolejnym spostrzeżeniem jest podobieństwo kolumn education i education-num, dlatego usunięta została jedna z nich. Po empirycznej analizie danych postanowiono również usunąć kolumny: 'fnlwgt', 'capital-gain', 'capital-loss'. Po usunięciu zbędnych kolumn oraz wypełnieniu brakujących wartości, a także zmapowaniu kategorii wypełnionych typem object na int, nasze dane wyglądają nastepująco:

```
df.head()
```

	age	workclass	education	marital-status	occupation	relationship	race	gender	hours-per-week
0	25	3.0	1.0	4.0	6.0	3.0	2.0	1.0	40
1	38	3.0	11.0	2.0	4.0	0.0	4.0	1.0	50
2	28	1.0	7.0	2.0	10.0	0.0	4.0	1.0	40
3	44	3.0	15.0	2.0	6.0	0.0	2.0	1.0	40
4	18	3.0	15.0	4.0	9.0	3.0	4.0	0.0	30

3 Random Forest Classification

Las losowy to zbiór drzew klasyfikacyjnych o podziałach binarnych. Dla konkretnej obserwacji (wyrażonej jako wektor wejściowy), każde z drzew zwraca decyzję lub krotkę prawdopodobieństw klasyfikacji, Prawdopodobieństwa z drzew wchodzących w skład lasu są traktowane jako głosy $->$ jako wynik zwracana jest decyzja która otrzymała najwięcej głosów (której średnie prawdopodobieństwo jest najwyższe).

3.1 Parametry

```
class sklearn.ensemble.RandomForestClassifier(n_estimators=100,
*, criterion='gini', max_depth=None, min_samples_split=2,
min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='auto',
max_leaf_nodes=None, min_impurity_decrease=0.0,
min_impurity_split=None, bootstrap=True, oob_score=False,
n_jobs=None, random_state=None, verbose=0, warm_start=False,
class_weight=None, ccp_alpha=0.0, max_samples=None)
```

Opis najwazniejszych parametrów:

1. n_estimators - liczba drzew w lesie
2. criterion"gini", "entropy" - funkcja miary jakości podziału
3. max_depth - maksymalna głębokość drzewa

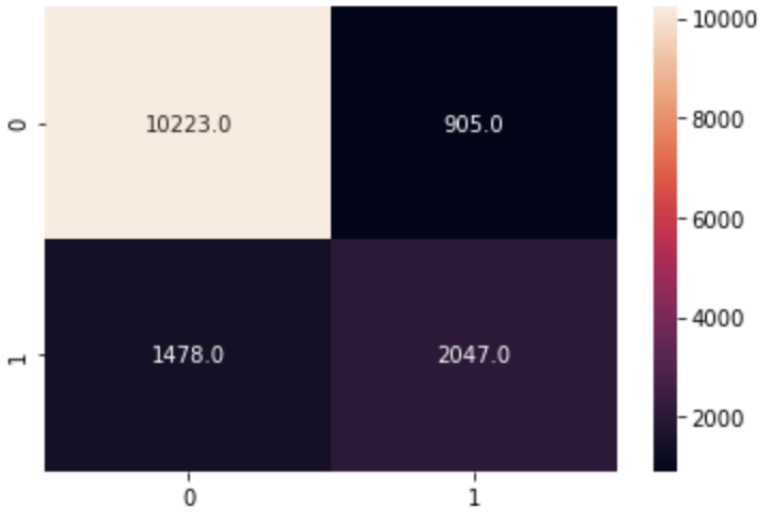
4. min_samples_split - minimalna liczba punktów danych, wymagana do podziału węzła
5. min_samples_leaf - minimalna liczba punktów danych, które muszą znajdować się w węźle liścia
6. max_features"auto", "sqrt", "log2" - liczba cech branych pod uwagę w poszukiwaniu najlepszego podziału
7. bootstrap - określa czy przy budowaniu drzew wykorzystuje się próbki bootstrap, jeśli False, wykorzystujemy cały dataset

4 Sposób wyboru zbioru testowego

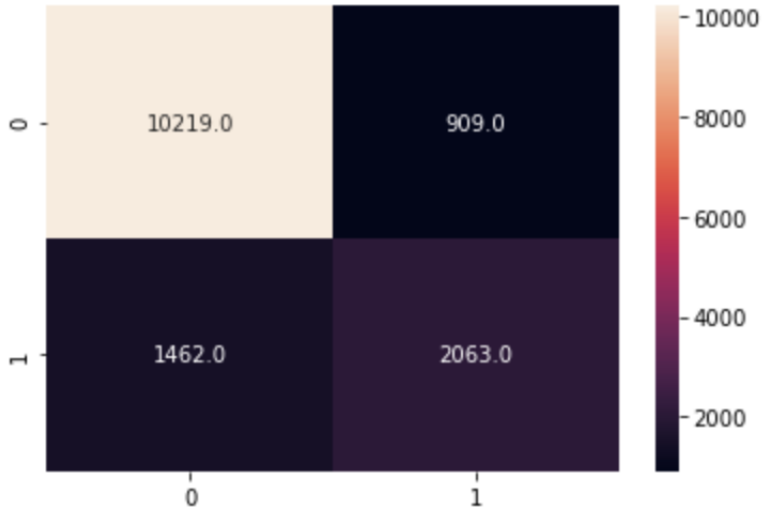
Do podziału danych na zbiór testowy i treningowy używamy funkcji train_test_split(), której jako parametry podajemy: dataframe zawierającą cechy wejściowe, kolumnę zawierającą to, co chcemy przewidzieć, rozmiar zbiorów uczącego i testowego oraz random_state, na podstawie którego dzielone są dane.

5 Macierz pomyłek

Macierz pomyłek (confusion matrix) to narzędzie do oceny jakosci klasyfikacji. Składa się z następujących klas: true-positive, true-negative, false-positive, false-negative. W naszym przypadku wygląda tak (dla parametrów testowanych przez nas):



Oraz dla parametrów wyznaczonych przez GridSearchCV():



6 Grid Search

Decydujący wpływ na efektywność w procesie kwalifikacji ma dobór parametrów. Grid Search to technika pozwalająca otrzymać optymalne wartości hiperparametrów. Grid Search buduje model dla każdej możliwej kombinacji parametrów, stąd potrafi być kosztowny pod względem złożoności obliczeniowej.

```
model = RandomForestClassifier()

kf = KFold(n_splits=5)

params = {'max_features': range(1,5),
          'max_depth': [15,20,25], 'criterion': ["entropy","gini"]}

grid=GridSearchCV(estimator=model, param_grid=params, cv=kf, scoring = 'accuracy')
gres=grid.fit(X_train,y_train)
print("Best",gres.best_score_)
print("params",gres.best_params_)
```

```
Best 0.8371114271942964
params {'criterion': 'entropy', 'max_depth': 15, 'max_features': 3}
```

Parametr 'estymator' to model, którego używamy, 'param_grid' przyjmuje listę parametrów oraz ich zakresów, 'cv' określa strategię podziału w ramach walidacji krzyżowej.

1. estymator - model, dla któego poszukujemy hiperparametrów
2. params_grid - przyjmuje listę parametrów oraz ich zakresów
3. scoring - metryka, której chcemy użyć
4. cv - określa strategię podziału w ramach walidacji krzyżowej