SPRAWOZDANIE

Zajęcia: Uczenie maszynowe

Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium Nr 2 Data 9.11.2024 Temat: "Praktyczne zastosowanie Drzew Decyzyjnych i metod Ensemble w Analizie Danych” Wariant 8	Tomasz Pietrzyk Informatyka II stopień, niestacjonarne, 1semestr, gr.1a

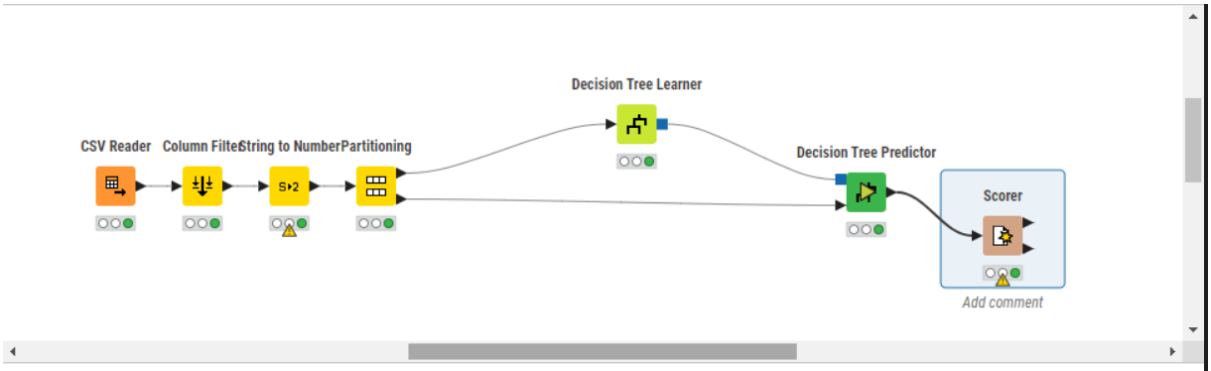
1. Polecenie: wariant 8 zadania

Opracować przepływ pracy uczenia maszynowego zagadnienia klasyfikacji (pojedyncze drzewo decyzyjne) oraz klasyfikacji ensemble (używając wszystkie modele wymienione w tutorialu) na podstawie zbioru danych według wariantu zadania.

Prostate Cancer https://www.kaggle.com/datasets/ashrafalsinglawi/prostate-cancer-survival-data

2. Opis programu opracowanego (kody źródłowe, rzuty ekranu)

GitHub: https://github.com/TomekPietrzyk/UM_I_2024_NS.git



▶ 1: Confusion matrix ▶ 2: Accuracy statistics Flow Variables

Rows: 8 | Columns: 11 Table Statistics

	#	RowID	TruePosit... Number (inte...	FalsePosi... Number (inte...	TrueNega... Number (inte...	FalseNeg... Number (inte...	Recall Number (dou...	Precision Number (dou...	Sensitivity Number (dou...	Specificity Number (dou...	
☐	1	t3a	27	71	199	101	0.211	0.276	0.211	0.737	0
☐	2	t3b	35	59	238	66	0.347	0.372	0.347	0.801	0
☐	3	t2c	84	122	138	54	0.609	0.408	0.609	0.531	0
☐	4	t2b	0	0	390	8	0	?	0	1	?
☐	5	t4	0	0	392	6	0	?	0	1	?
☐	6	NA	0	0	392	6	0	?	0	1	?
☐	7	t2a	0	0	387	11	0	?	0	1	?
☐	8	Overall	?	?	?	?	?	?	?	?	?

[61]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score

# Wczytanie danych
df = pd.read_csv("CancerProstateSurvival.csv")

# Wyświetlenie pierwszych kilku wierszy danych
print(df.head())

df = df[['times','patient.days_to_birth','patient.stage_event.tnm_categories.pathologic_categories.pathologic_t']]
df = df.dropna()

# Zakodowanie zmiennych kategorycznych w X
X = df.drop(columns=["patient.stage_event.tnm_categories.pathologic_categories.pathologic_t"])
X = pd.get_dummies(X, drop_first=True)  # One-hot encoding dla zmiennych kategorycznych

# Target (y)
y = df["patient.stage_event.tnm_categories.pathologic_categories.pathologic_t"]

# Podział na dane treningowe i testowe
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Trenowanie modelu
tree = DecisionTreeClassifier(max_depth=2, random_state=42)
tree.fit(X_train, y_train)

# Predykcja i ocena modelu
y_pred = tree.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
```

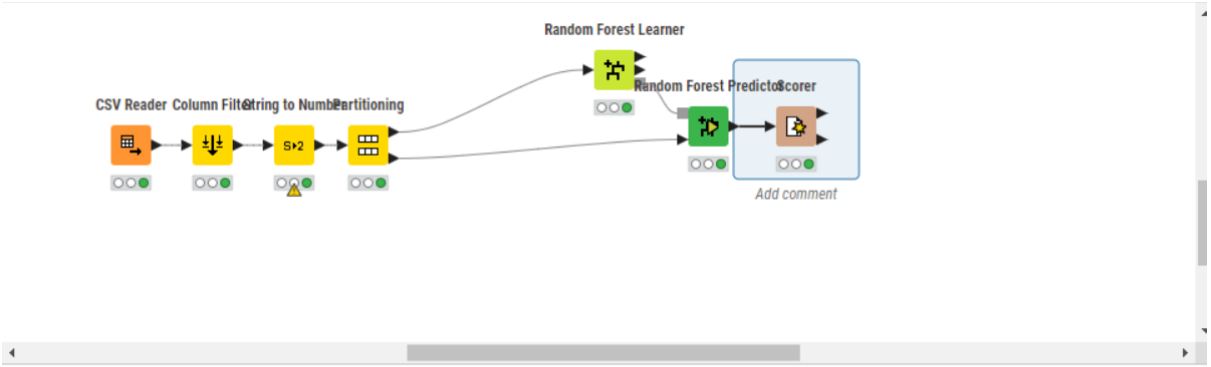
```
   times  patient.vital_status patient.gender patient.race patient.ethnicity  \
0    621                     0          male          NaN               NaN
1   1332                     0          male          NaN               NaN
2    995                     0          male          NaN               NaN
3    671                     0          male          NaN               NaN
4   1033                     0          male          NaN               NaN

   patient.days_to_birth patient.drugs.drug.therapy_types.therapy_type  \
0               -18658.0                                           NaN
1               -20958.0                                           NaN
2               -17365.0                                hormone therapy
3               -19065.0                                           NaN
4               -25904.0                                           NaN

   patient.stage_event.pathologic_stage  \
0                                   NaN
1                                   NaN
2                                   NaN
3                                   NaN
4                                   NaN

  patient.stage_event.tnm_categories.pathologic_categories.pathologic_t  \
0                                                 t2b
1                                                 t3a
2                                                  t4
3                                                 t2b
4                                                 t3b

   patient.stage_event.tnm_categories.pathologic_categories.pathologic_m
0                                                 NaN
1                                                 NaN
2                                                 NaN
3                                                 NaN
4                                                 NaN
Accuracy: 0.3917525773195876
```

[85]:
```
from sklearn.ensemble import RandomForestClassifier

# Inicjalizacja modelu Random Forest
rf_model = RandomForestClassifier(n_estimators=100, max_depth=5, random_state=42)
rf_model.fit(X_train, y_train)

# Predykcja i ewaluacja modelu
y_pred_rf = rf_model.predict(X_test)
print("Random Forest Accuracy:", accuracy_score(y_test, y_pred_rf))
```

```
Random Forest Accuracy: 0.4329896907216495
```

[113]:
```
from xgboost import XGBClassifier

y = pd.Categorical(y).codes

X_train,X_test,y_train,y_test = train_test_split(X,y,test_size = 0.2, random_state=42)

# Inicjalizacja modelu
xgb = XGBClassifier(n_estimators=100 ,max_depth=1, learning_rate=0.1)
xgb.fit(X_train, y_train)



# Predykcja i ocena modelu
y_pred_xgb = xgb.predict(X_test)
print("XGBoost Accuracy:", accuracy_score(y_test, y_pred_xgb))
```

```
XGBoost Accuracy: 0.3917525773195876
```



▶ 1: Confusion matrix ▶ 2: Accuracy statistics Flow Variables

Rows: 8 | Columns: 11 Table Statistics

#	RowID	TruePosit... Number (inte...	FalsePosi... Number (inte...	TrueNega... Number (inte...	FalseNeg... Number (inte...	Recall Number (dou...	Precision Number (dou...	Sensitivity Number (dou...	Specificity Number (dou...	
1	t3a	53	100	173	74	0.417	0.346	0.417	0.634	0
2	t4	0	1	391	8	0	0	0	0.997	
3	t3b	32	18	272	78	0.291	0.64	0.291	0.938	0
4	t2c	80	116	154	50	0.615	0.408	0.615	0.57	0
5	t2b	0	0	391	9	0	?	0	1	
6	NA	0	0	396	4	0	?	0	1	
7	t2a	0	0	388	12	0	?	0	1	
8	Overall	?	?	?	?	?	?	?	?	

3. Wnioski

Random forest najlepiej zachował się w predykcji metody leczenia raka prostaty. Jednak wynik jest z gatunku bardzo słabego, można powiedzieć, że algorytm na podstawie danych dokonał predykcji 2 na 5 sztuk(40%).