

Fakultet inženjerskih nauka  
Univerziteta u Kragujevcu

**Tema:**

*Primena Regresionih algoritama sa nadgledanim učenjem na data set-u  
Echocardiogram*

**student:**

*Nikola Mitrevski  
400/2021*

**predmetni profesor:**

*dr Nenad Filipović  
predmetni asistent:  
Tijana Šušteršić*

Kragujevac 2022.

## Sadržaj:

<b>1</b>	<b>Uvod .....</b>	<b>2</b>
<b>2</b>	<b>Preprocesiranje podataka .....</b>	<b>3</b>
2.1	Uvoz Echocardiogram skupa podataka.....	4
2.2	Prebrojavanje null vrednosti za svaku kolonu .....	5
2.3	Popunjavanje polja sa null vrednostima .....	5
2.4	Podela podataka na nezavisne i zavisnu promenljivu .....	6
2.5	Normalizacija podataka .....	7
2.6	Podela podataka na skup za treniranje i na skup za testiranje modela .....	8
<b>3</b>	<b>Kreiranje, treniranje i testiranje modela.....</b>	<b>9</b>
3.1	Treniranje modela .....	9
3.2	Testiranje modela .....	9
3.3	Grid Search .....	10
3.4	Linear Regression .....	10
3.5	Polynomial Regression .....	10
3.6	Decision Tree Regression .....	11
3.7	Random Forest Regression .....	11
3.8	Support Vector Regression .....	11
<b>4</b>	<b>Srednja kvadratna greška i koeficijent determinacije .....</b>	<b>13</b>
<b>5</b>	<b>Rezultati predikcije.....</b>	<b>14</b>
5.1	Linear Regression .....	14
5.2	Polynomial Regression .....	14
5.3	Decision Tree Regression .....	15
5.4	Random Forest Regression .....	16
5.5	Support Vector Regression .....	17
<b>6</b>	<b>Zaključak.....</b>	<b>19</b>
<b>7</b>	<b>Literatura.....</b>	<b>20</b>

# 1 Uvod

U ovom radu biće reči o nekoliko regresionih algoritama (Linear Regression, Polynomial Regression, Decision Tree Regression, Random Forest Regression i Support Vector Regression) koji su primenjeni na Echocardiogram skupu podataka. Cilj primene regresionih algoritama nad ovim skupom podataka je predviđanje broja meseci koje će pacijent preživeti nakon srčanog udara. [1]

Echocardiogram skup podataka sadrži 132 instance i 13 karakteristika:

1. survival – Broj meseci koje je pacijent preživeo. Pošto su svi pacijenti imali srčani udar u različito vreme, moguće je da su neki pacijenti preživeli manje od godinu dana, ali su još uvek živi (proveriti karakteristiku still\_alive). Takvi pacijenti se ne mogu koristiti.
2. still\_alive – Binarna karakteristika. Vrednost 0 znači da je pacijent preminuo, a vrednost 1 znači da je pacijent još uvek živ.
3. age\_at\_heart\_attack – Starost pacijenta u godinama kada je doživeo srčani udar.
4. pericardial\_effusion – Binarna karakteristika. Ova karakteristika predstavlja tečnost oko srca. Vrednost 0 znači da nema tečnosti, a vrednost 1 znači da ima.
5. fractional\_shortening – Mera kontraktilnosti oko srca.
6. epss – Odvajanje septuma E-tačke, još jedna mera kontraktilnosti.
7. lvdd – Krajnja dijastolna dimenzija leve komore srca. Ovo je mera veličine srca na kraju dijastole. Velika srca su obično bolesna srca.
8. wall\_motion\_score – Mera kako se kreću segmenti leve komore.
9. wall\_motion\_index – Jednako wall\_motion\_score podeljeno brojem većih segmenata.
10. mult – Karakteristika koja se može zanemariti.
11. name – Ime pacijenta.
12. group – Karakteristika koja se zanemaruje.
13. alive\_at\_1 – Binarna karakteristika izvedena iz prve dve karakteristike. Vrednost 0 znači da pacijent nije doživeo jednu godinu ili je praćen manje od jedne godine, a vrednost 1 znači da je pacijent bio živ više od jedne godine.

## 2 Preprocesiranje podataka

Prvobitni zadatak je bio klasifikacija podataka. Ciljna promenljiva je trebala da bude `alive_at_1`, međutim nakon vizuelizacije podataka, uočeno je da postoji slaba veza između ciljne promenljive `alive_at_1` i ostalih karakteristika (slika 1), tako da je rađena predikcija, a ciljna promenljiva je postala `survival`.



Slika 1 Vizuelizacija podataka

## 2.1 Uvoz Echocardiogram skupa podataka

Funkcija `loadData` se koristi za uvoz Echocardiogram skupa podataka.

U ovoj funkciji čitanje podataka iz fajla se vrši pomoću funkcije `pandas.read_csv`. [2]

Parametri koji su prosleđeni ovoj funkciji su:

- `dataFile` – naziv csv fajla;
- `na_values` – koristi se kako bi se nagovestilo koje vrednosti u kolonama su null;
- `error_bad_lines` – vrednost `False` označava da se vrste koje su neodgovarajuće dužine zanemaruju;
- `names` – nazivi kolona.

Pored čitanja ova funkcija služi i za uklanjanje određenih vrsta i kolona. Vrste koje se eliminišu su one čije kolone `survival` i `still_alive` sadrže null vrednosti. Kolone koje se eliminišu su: `name`, `group`, `mult`, `still_alive` i `alive_at_1`.

Sledeće linije koda predstavljaju defeniciju funkcije `loadData`:

```
def loadData():
    # Loading dataset from file
    ecg = pandas.read_csv(dataFile, na_values=["?"], error_bad_lines=False,
names=nameOfColumns)

    # Rows elimination
    ecg = ecg[ecg.survival.notnull()]
    ecg = ecg[ecg.still_alive.notnull()]
    ecg = ecg[ecg.still_alive == 0]

    # Delete columns
    del ecg['name']
    del ecg['group']
    del ecg['mult']
    del ecg['still_alive']
    del ecg['alive_at_1']
    return ecg
```

(88, 8)

Slika 2 Ukupan broj instanci i ukupan broj kolona

## 2.2 Prebrojavanje null vrednosti za svaku kolonu

Potrebno je izvršiti proveru da li preostale kolone sadrže null vrednosti.

Sledeća linija koda prebrojava null vrednosti za svaku kolonu:

```
print(ecg.isnull().sum())
```

Funkcija `pandas.DataFrame.isnull` [3] ispituje da li je vrednost polja null (vraća vrednost `True` or `False`), a funkcija `pandas.DataFrame.sum` [4] služi za prebrojavanje istih (po kolonama).

```
survival      0
age_at_heart_attack  3
pericardial_effusion  0
fractional_shortening  3
epss          8
lvdd          3
wall_motion_score  1
wall_motion_index  0
```

Slika 3 Kolone sa ukupnim brojem null vrednosti

## 2.3 Popunjavanje polja sa null vrednostima

Zbog malog skupa podataka (slika 2) ne smeju biti odbačene vrste sa null vrednostima. Na osnovu te činjenice, null vrednosti će biti popunjene izračunavanjem medijana za svaku kolonu.

Medijana se u teoriji verovatnoće i statistici opisuje kao broj koji razdvaja gornju polovinu uzorka, populacije ili raspodele verovatnoće od donje polovine. Medijana konačnog niza brojeva se može naći tako što se brojevi poređaju po veličini, a zatim se za nju uzimima srednji član niza. Ukoliko postoji paran broj članova niza, medijana nije jedinstvena, pa se često uzima aritmetička sredina dve vrednosti koje su kandidati za medijanu.

```
1, 3, 3, 6, 7, 8, 9
Median = 6

1, 2, 3, 4, 5, 6, 8, 9
Median = (4 + 5) ÷ 2
        = 4.5
```

Slika 4 Primeri računanja medijane

Sledeće linije koda popunjavaju polja sa null vrednostima pomoću medijana:

```
# Replace missing values
for columnName in ecg.keys():
    median = ecg[columnName].median()
    ecg[columnName] = ecg[columnName].fillna(median)
```

Funkcija `pandas.DataFrame.median` [5] računa medijanu za svaku kolonu, a funkcija `pandas.DataFrame.fillna` [6] popunjava svako polje koje sadrži null vrednost odgovarajućom vrednošću medijane.

```
survival      0
age_at_heart_attack  0
pericardial_effusion  0
fractional_shortening  0
epss          0
lvdd          0
wall_motion_score  0
wall_motion_index  0
```

*Slika 5 Kolone sa ukupnim brojem null vrednosti*

## 2.4 Podela podataka na nezavisne i zavisnu promenljivu

Skup podataka je potrebno podeliti na nezavisne promenljive i zavisnu promenljivu, kako bi mogla da se vrši predikcija.

Sledeće linije koda dele Echocardiogram skup podataka na nezavisne promenljive i zavisnu promenljivu (survival):

```
# Make a copy of the data
features = ecg.copy()
# pop off the regression target
target = features.pop('survival')
```

Funkcija `pandas.DataFrame.copy` [7] pravi kopiju objekta, a funkcija `pandas.DataFrame.pop` [8] izbacuje kolonu.

```
(88, 7)
(88,)
```

*Slika 6 Ukupan broj instanci i ukupan broj kolona nezavisnih i zavisne promenljive*

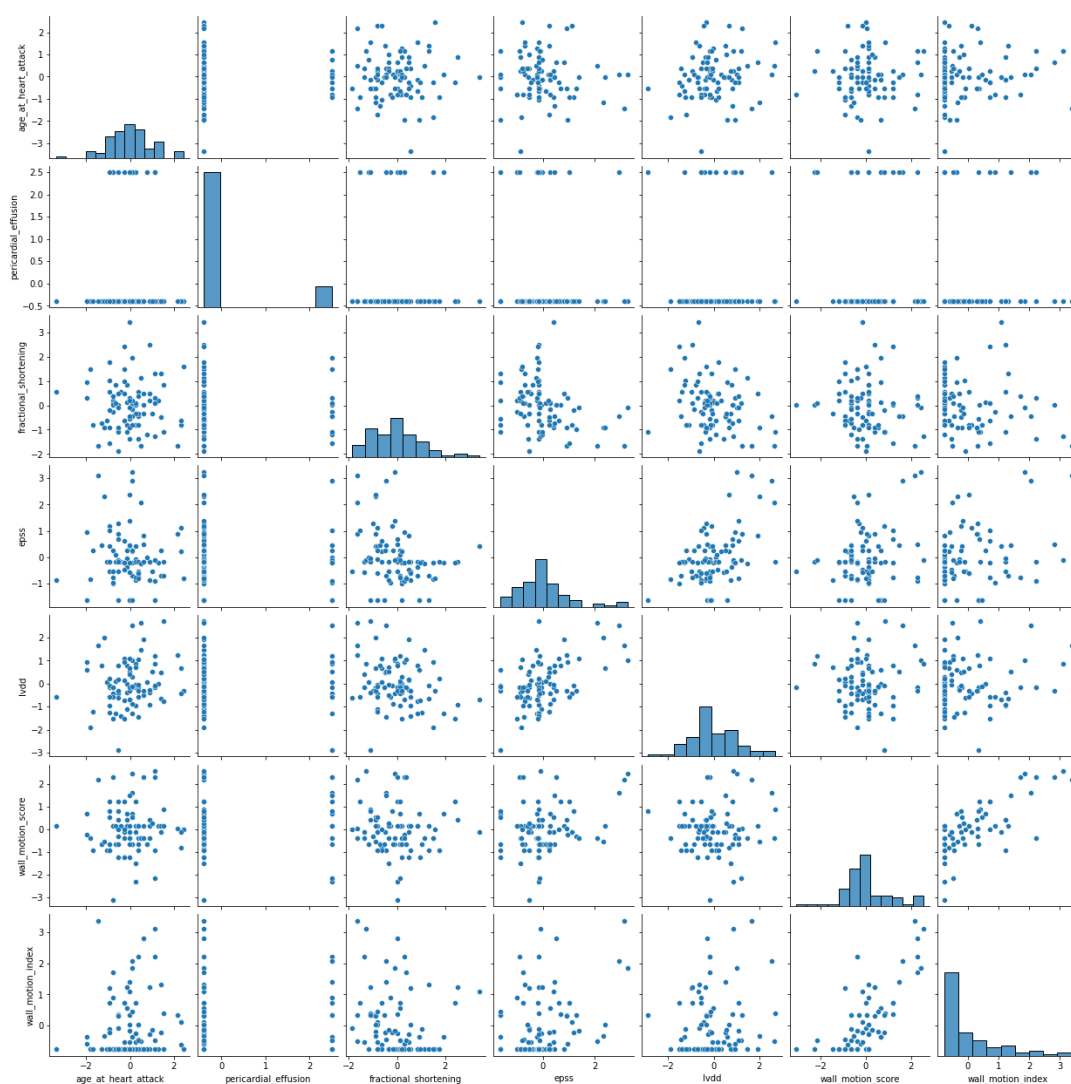
## 2.5 Normalizacija podataka

Da bi se dobili najtačniji rezultati, potrebno je primeniti normalizaciju podataka (skaliranje karakteristika).

Sledeće linije koda normalizuju podatke:

```
# Normalize features by mean and standard deviation
for columnName in features.keys():
    mean = features[columnName].mean()
    std = features[columnName].std()
    features[columnName] = (features[columnName] - mean) / std
```

Funkcija `pandas.DataFrame.keys` [9] vraća nazive kolona, a funkcije `pandas.DataFrame.mean` [10] i `pandas.DataFrame.std` [11] računaju srednju vrednost i standardnu devijaciju, respektivno.



Slika 7 Vizuelizacija podataka nakon njihove normalizacije



## 2.6 Podela podataka na skup za treniranje i na skup za testiranje modela

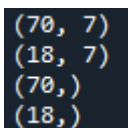
Da bi se modeli mogli naučiti i istestirati, potrebno je podeliti podatke na skup podataka za treniranje i na skup podataka za testiranje modela.

Sledeća linija koda deli podatke na trening i na test skup:

```
X_train, X_test, Y_train, Y_test = train_test_split(features, target, test_size=0.20)
```

Za podelu podataka na trening i test skup, korišćena je funkcija `sklearn.model_selection.train_test_split`. [12] Parametri koji su prosleđeni ovoj funkciji su:

- `features` – nezavisne promenljive;
- `target` – zavisna promenljiva;
- `test_size` – veličina test skupa (vrednost između 0.0 i 1.0).



```
(70, 7)
(18, 7)
(70,)
(18,)
```

*Slika 8 Ukupan broj instanci i ukupan broj kolona trening i test skupove*

## 3 Kreiranje, treniranje i testiranje modela

### 3.1 Treniranje modela

Funkcija `trainShow` se koristi za treniranje modela.

U ovoj funkciji za treniranje modela se koristi funkcija `fit`. Parametri koji su prosleđeni ovoj funkciji su:

- `trainX` – skup nezavisnih promenljivih za treniranje modela;
- `trainY` – skup zavisne promenljive za treniranje modela.

Sledeće linije koda predstavljaju defeniciju funkcije `trainShow`:

```
def trainShow(model, trainX, trainY):  
    model.fit(trainX, trainY)  
  
    try:  
        # In the case of GridSearchCV, we can show the best parameters  
        print(model.best_params_)  
        print(model.best_score_)  
    except:  
        print("Best params not implemented for model: %s" % type(model))
```

### 3.2 Testiranje modela

Funkcija `testShow` služi za testiranje modela.

U ovoj funkciji za testiranje modela se koristi funkcija `predict`. Parametri koji su prosleđeni ovoj funkciji su:

- `testX` – skup nezavisnih promenljivih za testiranje modela.

Sledeće linije koda predstavljaju defeniciju funkcije `testShow`:

```
def testShow(name, model, testX, testY):  
    predY = model.predict(testX)  
  
    plt.title(name)  
    plt.scatter(testY, predY)  
    plt.xlabel("True Survival")  
    plt.ylabel("Predicted Survival")  
    plt.show()  
  
    print('Predicted Value :', predY[3])  
    print('Actual Value :', testY.values[3])
```

### 3.3 Grid Search

sklearn.model\_selection.GridSearchCv funkcija se koristi kod modela klasifikacije i regresije za određivanje najboljih hiperparametara. Neki od parametara koji se prosleđuju ovoj funkciji su sam model za koji se traže najbolji hiperparametri, parametri koji će biti isprobani kao najbolji hiperparametri, broj unakrsnih provera, itd. [13]

### 3.4 Linear Regression

Sledeće linije koda kreiraju, treniraju i testiraju Linear Regression model:

```
lr = LinearRegression()

trainShow(lr, X_train, Y_train)

testShow("Linear Regression", lr, X_test, Y_test)
```

### 3.5 Polynomial Regression

Sledeće linije koda kreiraju, treniraju i testiraju Polynomial Regression model:

```
def PolynomialRegression(degree=2, **kwargs):
    return make_pipeline(PolynomialFeatures(degree), LinearRegression(**kwargs))

param_grid = {
    'polynomialfeatures__degree': numpy.arange(10),
    'linearregression__fit_intercept': [True, False],
    'linearregression__normalize': [True, False]}

pr = GridSearchCV(PolynomialRegression(), param_grid, cv=10,
    scoring='neg_mean_squared_error')

trainShow(pr, X_train, Y_train)

testShow("Polynomial Regression", pr, X_test, Y_test)
```

Parametri koji su prosleđeni sklearn.model\_selection.GridSearchCv funkciji su:

- model – model za koji se traže najbolji hiperparametri;
- param\_grid – skup parametara koji se isprobavaju kao najbolji hiperparametri;
- cv – broj unakrsnih provera;
- scoring – koristi se za procenu performansi unakrsno validiranog modela na testnom skupu.

### 3.6 Decision Tree Regression

Sledeće linije koda kreiraju, treniraju i testiraju Decision Tree Regression model:

```
parameters={ "splitter":["best","random"],
              "max_depth" : [1,3,5,7,9,11,12],
              "min_samples_leaf": [1,2,3,4,5,6,7,8,9,10],
              "min_weight_fraction_leaf": [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9],
              "max_features":["auto","log2","sqrt"],
              "max_leaf_nodes": [None,10,20,30,40,50,60,70,80,90] }

dtr = GridSearchCV(DecisionTreeRegressor(), param_grid=parameters,
scoring='neg_mean_squared_error', cv=3, verbose=3)

trainShow(dtr, X_train, Y_train)

testShow("Decision Tree Regressor", dtr, X_test, Y_test)
```

### 3.7 Random Forest Regression

Sledeće linije koda kreiraju, treniraju i testiraju Random Forest Regression model:

```
rfr = GridSearchCV(
    RandomForestRegressor(), cv=5, error_score=numpy.nan,
    param_grid={
        'max_depth' : [ 2, 5, 7, 10],
        'n_estimators': [20, 30, 50, 75]
    }
)

trainShow(rfr, X_train, Y_train)

testShow("Random Forest Regressor", rfr, X_test, Y_test)
```

### 3.8 Support Vector Regression

Sledeće linije koda kreiraju, treniraju i testiraju Support Vector Regression model:

```
svr = GridSearchCV(
    SVR(),
    cv=5,
    param_grid={
        "kernel":["rbf", "linear"],
        "C": [0.2, 0.5, 1],
        "gamma": [0.1, 0.3, 0.5],
        "epsilon": numpy.logspace(-6, 0, 10)
    }
)
```

```
trainShow(svr, X_train, Y_train)
```

```
testShow("Support Vector Regression ", svr, X_test, Y_test)
```

## 4 Srednja kvadratna greška i koeficijent determinacije

Srednje kvadratna greška se može dobiti iz sledeće jednačine:

$$MSE = \frac{1}{q} \sum_{i=n+1}^{n+q} (y_i - \hat{y}_i)^2,$$

gde je  $y_i$  stvaran izlaz, a  $\hat{y}_i$  željeni izlaz.

Koeficijent determinacije ( $R^2$ ) se može dobiti iz sledeće jednačine:

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2},$$

gde je  $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$ , a  $\sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n \epsilon_i^2$ .

Funkcija mse se koristi za izračunavanje srednje kvadratne greške i koeficijenta determinacije.

U ovoj funkciji za računanje srednje kvadratne greške koristi se funkcija `sklearn.metrics.mean_squared_error`, dok se za računanje koeficijenta determinacije koristi funkcija `sklearn.metrics.r2_score`. [14] Parametri koji su prosleđeni ovim funkciji su:

- testY – skup zavisne promenljive za testiranje modela;
- predY – skup zavisne promenljive dobijene iz predikcije modela.

Sledeće linije koda predstavljaju definiciju funkcije mse:

```
def mse(model, testX, testY):  
    predY = model.predict(testX)  
  
    r2 = r2_score(testY, predY)  
    print("r2_score: ", r2)  
  
    return mean_squared_error(testY, predY)
```

## 5 Rezultati predikcije

### 5.1 Linear Regression

Najbolji parametri i najbolji rezultat modela Linear Regression nepostoje, jer funkcija `sklearn.model_selection.GridSearchCv` nije implementirana za ovaj model (slika 9).

```
Best params not implemented for model:  
<class 'sklearn.linear_model._base.LinearRegression'>
```

*Slika 9 Najbolji parametri Linear Regression modela*

Željeni izlaz i stvarni izlaz ovog modela, prikazani su na slici 10.

```
Predicted Value: 30.49995382304612  
Actual Value: 26.0
```

*Slika 10 Željeni izlaz i stvarni izlaz Linear Regression modela*

Koeficijent determinacije i srednja kvadratna greška, prikazane su na slici 11.

```
r2_score: -0.11406615384093999  
mse: 160.6593425063706
```

*Slika 11 Koeficijent determinacije i srednja kvadratna greška Linear Regression modela*

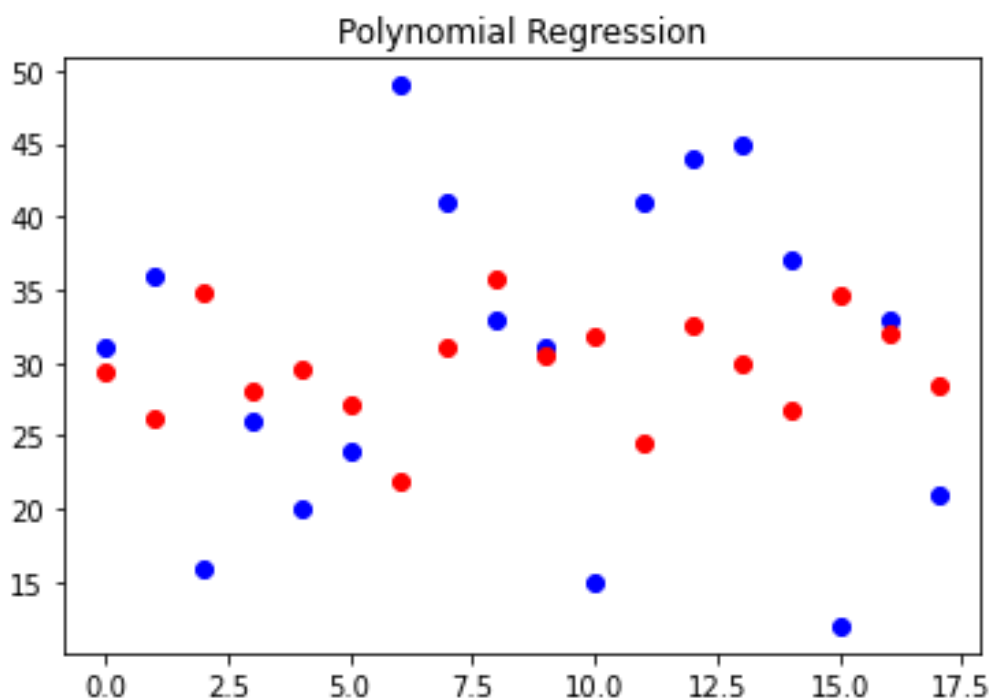
### 5.2 Polynomial Regression

Najbolji parametri i najbolji rezultat modela Polynomial Regression prikazani su na slici 12.

```
Best Parameters: {'linearregression__fit_intercept': False, 'linearregression__normalize': True,  
'polynomialfeatures__degree': 1}  
Best Score: -182.36958568541144
```

*Slika 12 Najbolji parametri i najbolji rezultat Polynomial Regression modela*

Grafički prikaz željenih i stvarnih izlaza ovog modela, prikazani su na slici 13.



Slika 13 Grafički prikaz željenih i stvarnih izlaza Polynomial Regression modela

Željeni izlaz i stvarni izlaz ovog modela, prikazani su na slici 14.

```
Predicted Value: 28.113975193920997  
Actual Value: 26.0
```

Slika 14 Željeni izlaz i stvarni izlaz Polynomial Regression modela

Koeficijent determinacije i srednja kvadratna greška, prikazane su na slici 15.

```
r2_score: -0.42144002146147774  
mse: 165.40034027505914
```

Slika 15 Koeficijent determinacije i srednja kvadratna greška Polynomial Regression modela

### 5.3 Decision Tree Regression

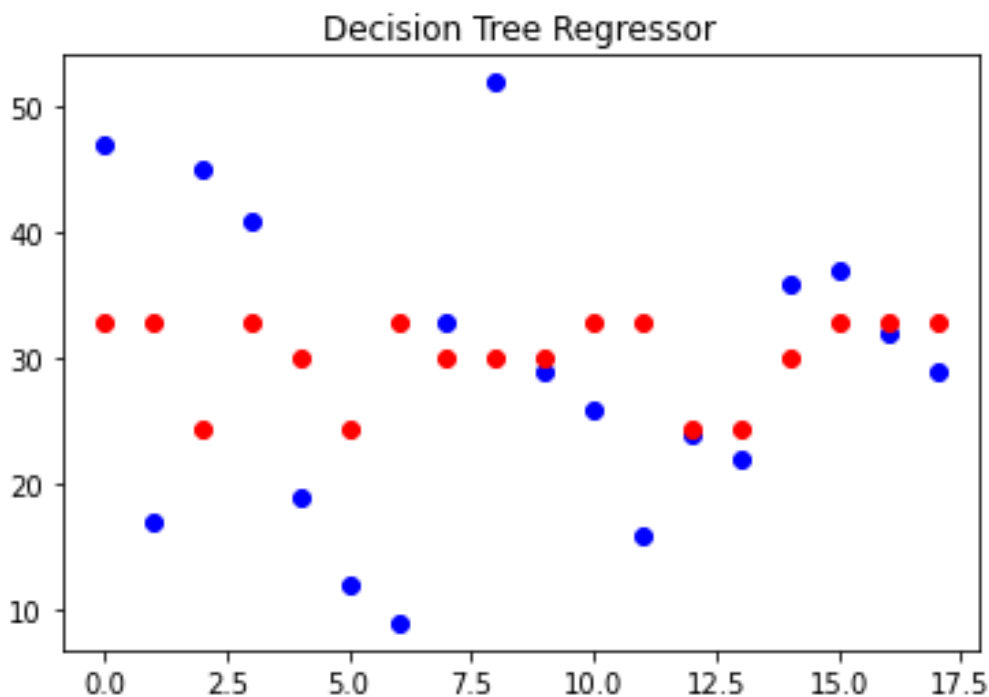
Najbolji parametri i najbolji rezultat modela Decision Tree Regression prikazani su na slici 16.

```
Best Parameters: {'max_depth': 7, 'max_features': 'sqrt', 'max_leaf_nodes': 20, 'min_samples_leaf': 3,  
'min_weight_fraction_leaf': 0.2, 'splitter': 'best'}  
Best Score: -130.82003536051892
```

Slika 16 Najbolji parametri i najbolji rezultat Decision Tree Regression modela



Grafički prikaz željenih i stvarnih izlaza ovog modela, prikazani su na slici 17.



Slika 17 Grafički prikaz željenih i stvarnih izlaza Decision Tree Regression modela

Željeni izlaz i stvarni izlaz ovog modela, prikazani su na slici 18.

```
Predicted Value: 32.86666666666667  
Actual Value: 41.0
```

Slika 18 Željeni izlaz i stvarni izlaz Decision Tree Regression modela

Koeficijent determinacije i srednja kvadratna greška, prikazane su na slici 19.

```
r2_score: -0.04194400464303483  
mse: 149.06231019510477
```

Slika 19 Koeficijent determinacije i srednja kvadratna greška Decision Tree Regression modela

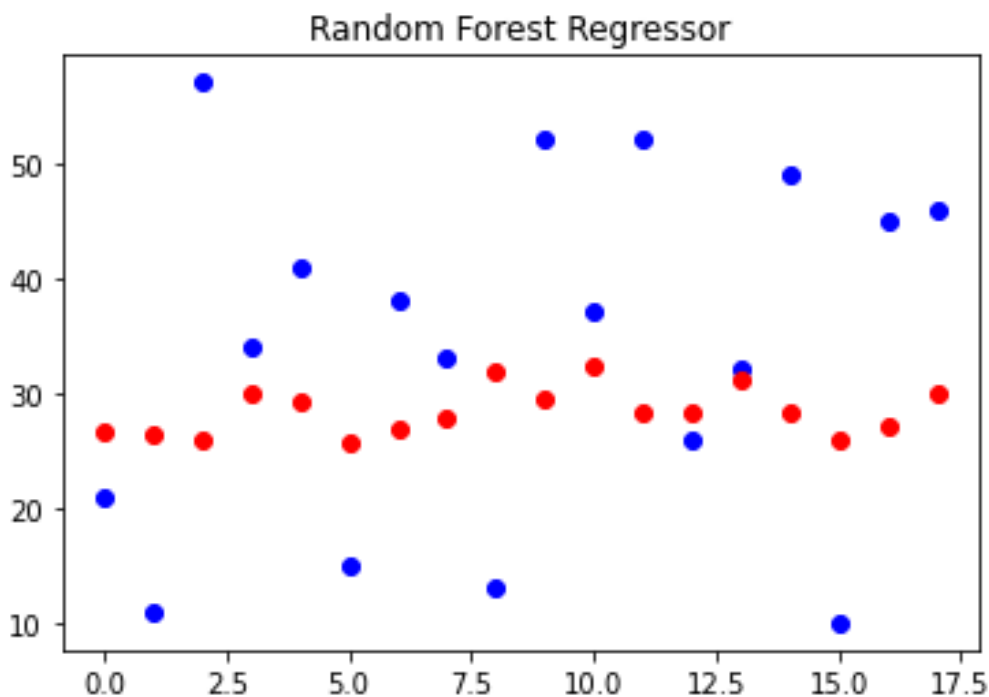
## 5.4 Random Forest Regression

Najbolji parametri i najbolji rezultat modela Random Forest Regression prikazani su na slici 20.

```
Best Parameters: {'max_depth': 2, 'n_estimators': 50}  
Best Score: -0.3943330345106798
```

Slika 20 Najbolji parametri i najbolji rezultat Random Forest Regression modela

Grafički prikaz željenih i stvarnih izlaza ovog modela, prikazani su na slici 21.



Slika 21 Grafički prikaz željenih i stvarnih izlaza Random Forest Regression modela

Željeni izlaz i stvarni izlaz ovog modela, prikazani su na slici 22.

```
Predicted Value: 29.93317314668739  
Actual Value: 34.0
```

Slika 22 Željeni izlaz i stvarni izlaz Random Forest Regression modela

Koeficijent determinacije i srednja kvadratna greška, prikazane su na slici 23.

```
r2_score: -0.12419651476843185  
mse: 242.70153646611814
```

Slika 23 Koeficijent determinacije i srednja kvadratna greška Random Forest Regression modela

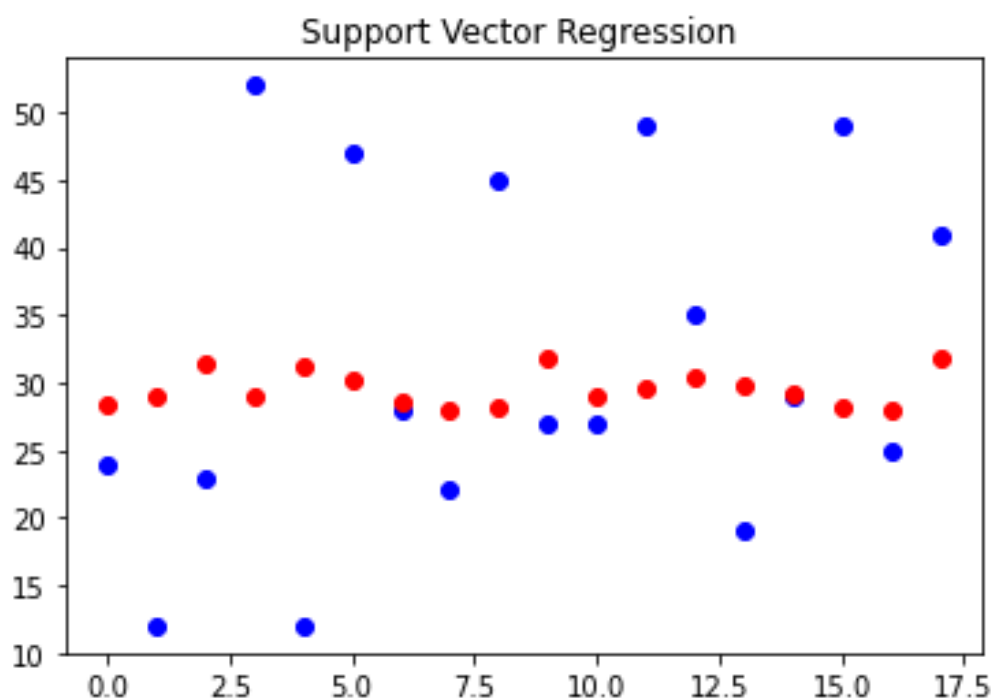
## 5.5 Support Vector Regression

Najbolji parametri i najbolji rezultat modela Support Vector Regression prikazani su na slici 24.

```
Best Parameters: {'C': 1, 'epsilon': 1.0, 'gamma': 0.3, 'kernel': 'rbf'}  
Best Score: -0.06727171219856772
```

Slika 24 Najbolji parametri i najbolji rezultat Support Vector Regression modela

Grafički prikaz željenih i stvarnih izlaza ovog modela, prikazani su na slici 25.



Slika 25 Grafički prikaz željenih i stvarnih izlaza Support Vector Regression modela

Željeni izlaz i stvarni izlaz ovog modela, prikazani su na slici 26.

```
Predicted Value: 29.003193533161042  
Actual Value: 52.0
```

Slika 26 Željeni izlaz i stvarni izlaz Support Vector Regression modela

Koeficijent determinacije i srednja kvadratna greška, prikazane su na slici 27.

```
r2_score: -0.05247283564462624  
mse: 163.39315936087868
```

Slika 27 Koeficijent determinacije i srednja kvadratna greška Support Vector Regression modela

## 6 Zaključak

U ovom radu je bilo reči o sledećim regresionim algoritmima: Linear Regression, Polynomial Regression, Decision Tree Regression, Random Forest Regression i Support Vector Regression, koji su primenjeni na Echocardiogram skupu podataka. Cilj primene regresionih algoritama nad ovim skupom podataka je bio predviđanje broja meseci koje će pacijent preživeti nakon srčanog udara.

Zbog malog skupa podataka nisu smele biti odbačene vrste sa null vrednostima, tako da su null vrednosti popunjavane medijanama. Nakon eliminacija null vrednosti, rađena je podela podataka na zavisne promenljive i nezavisnu promenljivu, a zatim normalizacija i podela podataka na trening i test skup.

Najbolji hiperparametri za svaki od modela predikcije su određeni pomoću `sklearn.model_selection.GridSearchCv` funkcije.

Na osnovu dobijenih rezultata, došli smo do zaključka da najbolju predikciju daje model Decision Tree.

## 7 Literatura

- [1] „Echocardiogram Data Set,” [Na mreži]. Available: <http://archive.ics.uci.edu/ml/datasets/Echocardiogram>. [Poslednji pristup 24 03 2022].
- [2] „pandas.read\_csv,” Pandas, [Na mreži]. Available: [https://pandas.pydata.org/docs/reference/api/pandas.read\\_csv.html](https://pandas.pydata.org/docs/reference/api/pandas.read_csv.html). [Poslednji pristup 26 03 2022].
- [3] „pandas.DataFrame.isnull,” Pandas, [Na mreži]. Available: <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.isnull.html>. [Poslednji pristup 26 03 2022].
- [4] „pandas.DataFrame.sum,” Pandas, [Na mreži]. Available: <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.sum.html>. [Poslednji pristup 26 03 2022].
- [5] „pandas.DataFrame.median,” Pandas, [Na mreži]. Available: <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.median.html>. [Poslednji pristup 26 03 2022].
- [6] „pandas.DataFrame.fillna,” Pandas, [Na mreži]. Available: <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.fillna.html>. [Poslednji pristup 26 03 2022].
- [7] „pandas.DataFrame.copy,” Pandas, [Na mreži]. Available: <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.copy.html>. [Poslednji pristup 26 03 2022].
- [8] „pandas.DataFrame.pop,” Pandas, [Na mreži]. Available: <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.pop.html>. [Poslednji pristup 26 03 2022].
- [9] „pandas.DataFrame.keys,” Pandas, [Na mreži]. Available: <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.keys.html>. [Poslednji pristup 26 03 2022].
- [10] „pandas.DataFrame.mean,” Pandas, [Na mreži]. Available: <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.mean.html>. [Poslednji pristup 26 03 2022].
- [11] „pandas.DataFrame.std,” Pandas, [Na mreži]. Available: <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.std.html>. [Poslednji pristup 26 03 2022].
- [12] „sklearn.model\_selection.train\_test\_split,” Scikit Learn, [Na mreži]. Available: [https://scikit-learn.org/stable/modules/generated/sklearn.model\\_selection.train\\_test\\_split.html](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html). [Poslednji pristup 26 03 2022].

[13] „sklearn.model\_selection.GridSearchCV,“ Scikit Learn, [Na mreži]. Available: [https://scikit-learn.org/stable/modules/generated/sklearn.model\\_selection.GridSearchCV.html](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html). [Poslednji pristup 26 03 2022].

[14] „sklearn.metrics.r2\_score,“ Scikit Learn, [Na mreži]. Available: [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2\\_score.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html). [Poslednji pristup 26 03 2022].