Applied and Practical Data science

Team name - WSU Final

Apoorva Palle (Z248H929) , Areeba Mohiuddin (D655Y483) , Sri Manikanta Reddy Tangirala (G332B644) , Lakshmi Vasanthi Godi (V567R326)

Importing all the requiredlibraryes

In [2]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from tabulate import tabulate
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
from sklearn.metrics import accuracy_score
from sklearn.svm import SVC
import seaborn as sns
```

In [3]:
```
df = pd.read_csv("C:\\Users\\Aneesh Reddy\\OneDrive\\Desktop\\appu ads EDA\\GSS2018

AGE = df['AGE']
RELIG = df['RELIG']
CLASS = df['CLASS']
PARTYID = df['PARTYID']
PRES12 = df['PRES12']
Jew = df['Jew']
OTHER = df['OTHER']
ROWNGUN = df['ROWNGUN']
OTH16 = df['OTH16']
SEI10EDUC = df['SEI10EDUC']
PRES16 = df['PRES16']
X = pd.DataFrame({
    'AGE': AGE,
    'RELIG': RELIG,
    'CLASS': CLASS,
    'PARTYID': PARTYID,
    'PRES12': PRES12,
    'Jew': Jew,
    'OTHER': OTHER,
    'ROWNGUN': ROWNGUN,
    'OTH16': OTH16,
    'SEI10EDUC': SEI10EDUC
})
```

```
y = PRES16
X.head()
```

Out[3]:

	AGE	RELIG	CLASS	PARTYID	PRES12	Jew	OTHER	ROWNGUN	OTH16	SEI10EDUC
0	43	11	2	5	2	0	0	0	0	82.4
1	74	2	2	2	1	0	0	0	0	16.5
2	42	4	3	4	2	0	0	0	0	89.4
3	63	1	3	2	2	0	0	0	0	86.7
4	71	2	4	6	2	0	0	1	0	79.2

Splitting the data into testing and training data sets

In [4]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta
```

Model Selection

Decision Tree Regression model

In [5]:
```
decisionTree_model = DecisionTreeRegressor()
decisionTree_model.fit(X_train, y_train)
D_pred = decisionTree_model.predict(X_test)
```

calculating R-squared score and mean square error for the Decision Tree Regression

In [6]:
```
a_DTR = accuracy_score(y_test, D_pred)
print("Accuracy:", a_DTR)
print("Accuracy percent :", a_DTR*100)
r2_D = r2_score(y_test, D_pred)
print("decisionTree_model R-squared score is " , r2_D)
```

```
Accuracy: 0.6127659574468085
Accuracy percent : 61.27659574468085
decisionTree_model R-squared score is  -0.32734956717060126
```

Support Vector Machine

In [7]:
```
SVM_model = SVC(kernel='linear', random_state=42)
SVM_model.fit(X_train, y_train)
S_pred = SVM_model.predict(X_test)
```

calculating R-squared score and mean square error for SVM

In [8]:
```
a_svm = accuracy_score(y_test, S_pred)
print("Accuracy:", a_svm)
print("Accuracy percent :", a_svm*100)
r2_S = r2_score(y_test, S_pred)
print("SVM R-squared score is " , r2_S)
```

```
Accuracy: 0.7297872340425532
Accuracy percent : 72.97872340425532
SVM R-squared score is  0.2647734482683177
```

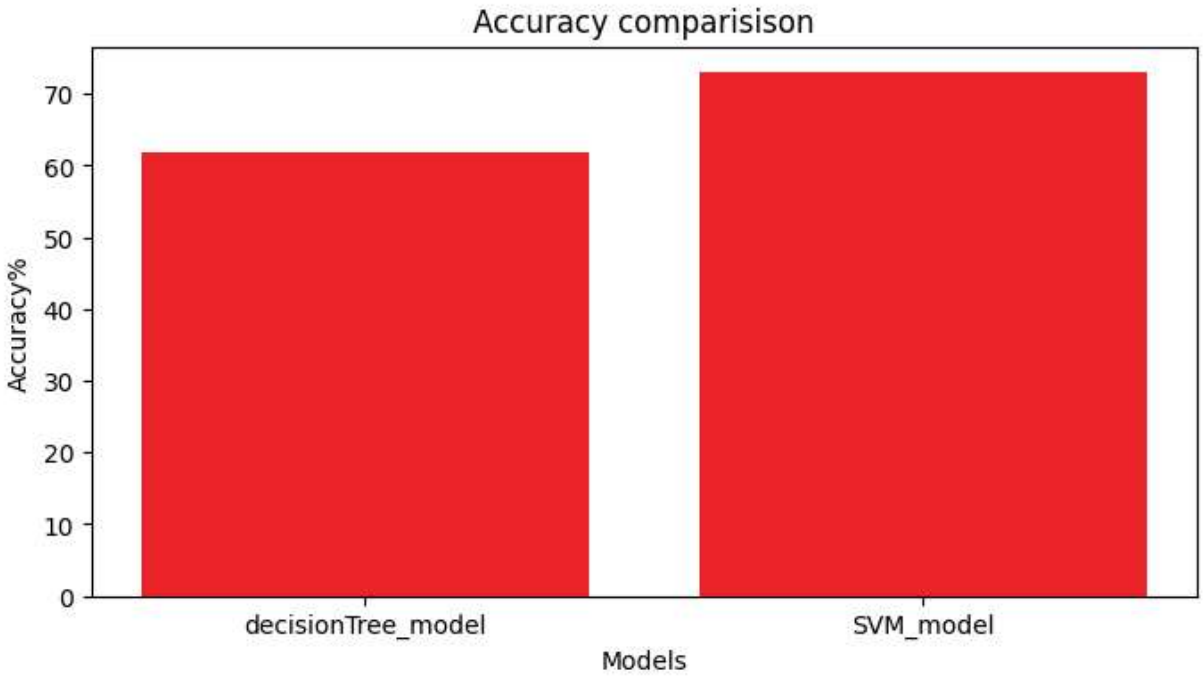
By calculating the Accuracy and R-squared score, we can say that the Support Vector Machine is the best fit for our data set as we can see that the Accuracy percent and R-squared score are high when compared to Decision Tree Regression model.

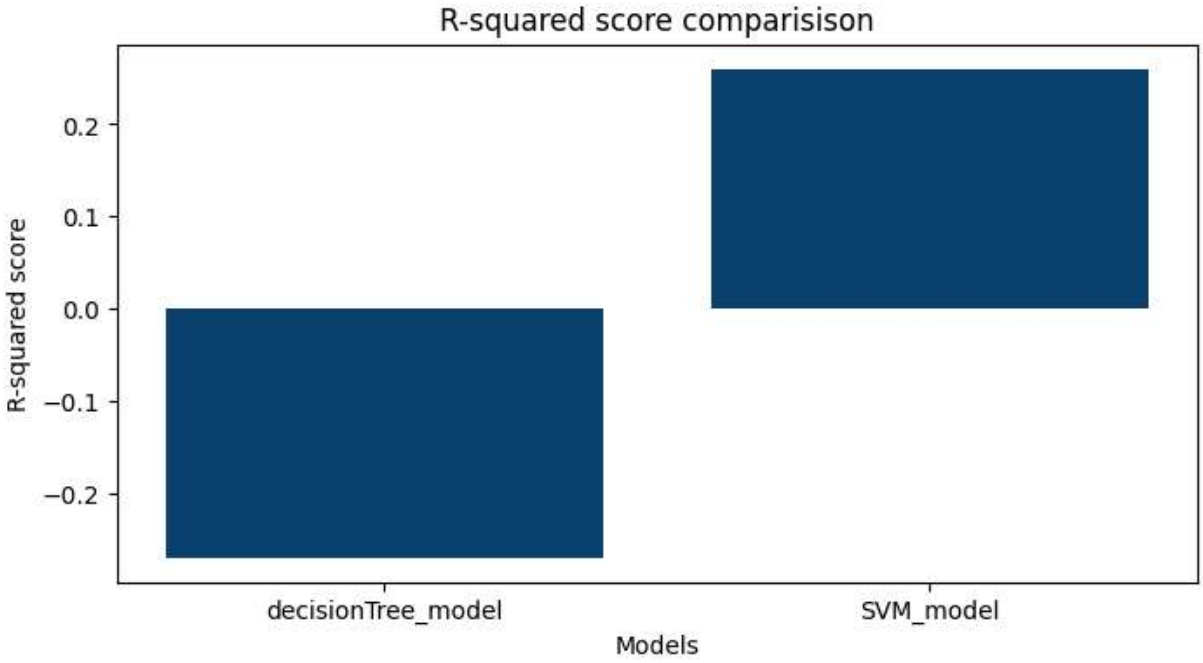
Visualization Showing and Comparing Accuracy and R-squared score of the models

In [9]:
```
models = ['decisionTree_model', 'SVM_model']
accuracy_values = [61.91,72.97]
r2_values = [-0.27,0.26]

# Plotting the MSE values
plt.figure(figsize=(8 , 4))
plt.bar(models, accuracy_values, color='#EC232B')
plt.xlabel('Models')
plt.ylabel('Accuracy%')
plt.title('Accuracy comparisison')
plt.show()

# Plotting the R-squared score
plt.figure(figsize=(8, 4))
plt.bar(models, r2_values, color='#0E4572')
plt.xlabel('Models')
plt.ylabel('R-squared score')
plt.title('R-squared score comparisison')
plt.show()
```





Accuracy and R-squared of SVM

In [10]:
```
print("Accuracy of SVM is " , a_svm*100)
print("R-squared score of SVM is " , r2_S)
```

```
Accuracy of SVM is  72.97872340425532
R-squared score of SVM is  0.2647734482683177
```