Data Analysis Part-2

```
import pandas as pd
import numpy as np
from sklearn.datasets import fetch_openml
from sklearn.preprocessing import LabelEncoder,OneHotEncoder


#Load the dataset
data=fetch_openml('titanic',version=1,as_frame=True)
print(data)             (module) preprocessing
```

```
1305                              NaN
1306                              NaN
1307                              NaN
1308                              NaN

[1309 rows x 13 columns], 'target': 0       1
1       1
2       0
3       0
4       0
       ..
1304    0
1305    0
1306    0
1307    0
1308    0
Name: survived, Length: 1309, dtype: category
Categories (2, object): ['0', '1'], 'frame':       pclass survived                                             name  \
0          1        1                    Allen, Miss. Elisabeth Walton
1          1        1                   Allison, Master. Hudson Trevor
2          1        0                     Allison, Miss. Helen Loraine
3          1        0             Allison, Mr. Hudson Joshua Creighton
4          1        0  Allison, Mrs. Hudson J C (Bessie Waldo Daniels)
...      ...      ...                                              ...
1304       3        0                          Zabour, Miss. Hileni
1305       3        0                         Zabour, Miss. Thamine
1306       3        0                     Zakarian, Mr. Mapriededer
1307       3        0                           Zakarian, Mr. Ortin
1308       3        0                         Zimmerman, Mr. Leo

         sex      age  sibsp  parch  ticket      fare    cabin embarked boat  \
0     female  29.0000      0      0   24160  211.3375       B5        S    2
1       male   0.9167      1      2  113781  151.5500  C22 C26        S   11
2     female   2.0000      1      2  113781  151.5500  C22 C26        S  NaN
3       male  30.0000      1      2  113781  151.5500  C22 C26        S  NaN
4     female  25.0000      1      2  113781  151.5500  C22 C26        S  NaN
...      ...      ...    ...    ...     ...       ...      ...      ...  ...
1304  female  14.5000      1      0    2665   14.4542      NaN        C  NaN
1305  female      NaN      1      0    2665   14.4542      NaN        C  NaN
1306    male  26.5000      0      0    2656    7.2250      NaN        C  NaN
1307    male  27.0000      0      0    2670    7.2250      NaN        C  NaN
1308    male  29.0000      0      0  315082    7.8750      NaN        S  NaN

       body                        home.dest
0       NaN                     St Louis, MO
1       NaN  Montreal, PQ / Chesterville, ON
2       NaN  Montreal, PQ / Chesterville, ON
3     135.0  Montreal, PQ / Chesterville, ON
4       NaN  Montreal, PQ / Chesterville, ON
...     ...                              ...
1304  328.0                              NaN
1305    NaN                              NaN
1306  304.0                              NaN
1307    NaN                              NaN
1308    NaN                              NaN

[1309 rows x 14 columns], 'categories': None, 'feature_names': ['pclass', 'name', 'sex', 'age', 'sibsp', 'parch', 'tick
```

```
data['feature_names']
```

```
['pclass',
 'name',
 'sex',
 'age',
 'sibsp',
 'parch',
 'ticket',
 'fare',
 'cabin',
 'embarked',
 'boat',
 'body',
 'home.dest']
```

```
data_f=data.frame.copy()


#Dropping the empty values
data=data_f[['age','sex','fare','embarked','pclass','survived']].dropna()


set(list(data['embarked']))
```

```
{'C', 'Q', 'S'}
```

```
#Lable Encoding          (module) preprocessing
le=LabelEncoder()
data['embarked_le']=le.fit_transform(data['embarked'])


data.columns
```

```
Index(['age', 'sex', 'fare', 'embarked', 'pclass', 'survived', 'embarked_le'], dtype='object')
```

```
data['embarked_le']
```

	embarked_le
0	2
1	2
2	2
3	2
4	2
...	...
1301	0
1304	0
1306	0
1307	0
1308	2

1043 rows × 1 columns

dtype: int64

```
#One-Hot Encoder
ohe=OneHotEncoder()
# df_ohe=pd.get_dummies(data['sex'])
df_ohe=pd.get_dummies(data,columns=['sex'])
df_ohe
```

	age	fare	embarked	pclass	survived	embarked_le	sex_female	sex_male
0	29.0000	211.3375	S	1	1	2	True	False
1	0.9167	151.5500	S	1	1	2	False	True
2	2.0000	151.5500	S	1	0	2	True	False
3	30.0000	151.5500	S	1	0	2	False	True
4	25.0000	151.5500	S	1	0	2	True	False
...	...	...	...	...	...	...	...	...
1301	45.5000	7.2250	C	3	0	0	False	True
1304	14.5000	14.4542	C	3	0	0	True	False
1306	26.5000	7.2250	C	3	0	0	False	True
1307	27.0000	7.2250	C	3	0	0	False	True
1308	29.0000	7.8750	S	3	0	2	False	True

1043 rows × 8 columns

Next steps: Generate code with df_ohe | View recommended plots | New interactive sheet

```
#Loadnthe dataset of diabetes
from sklearn.datasets import load_diabetes
```

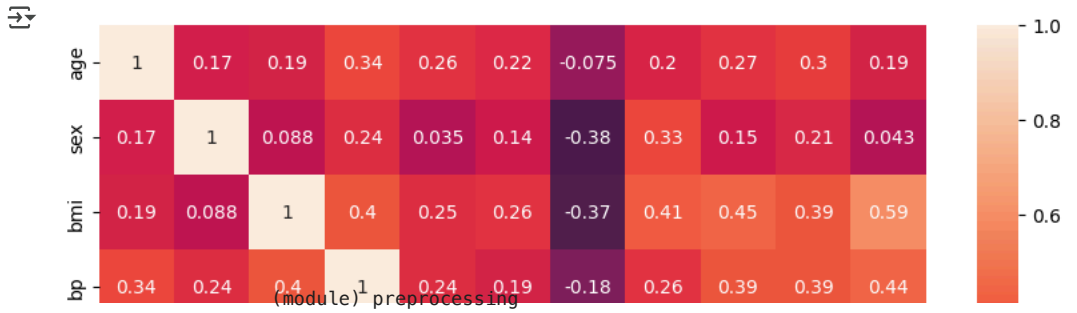
```
data=load_diabetes(as_frame=True)
data=data.frame
print(data)
```

```
          age       sex       bmi        bp        s1        s2        s3  \
0    0.038076  0.050680  0.061696  0.021872 -0.044223 -0.034821 -0.043401
1   -0.001882 -0.044642 -0.051474 -0.026328 -0.008449 -0.019163  0.074412
2    0.085299  0.050680  0.044451 -0.005670 -0.045599 -0.034194 -0.032356
3   -0.089063 -0.044642 -0.011595 -0.036656  0.012191  0.024991 -0.036038
4    0.005383 -0.044642 -0.036385  0.021872  0.003935  0.015596  0.008142
..        ...       ...       ...       ...       ...       ...       ...
437  0.041708  0.050680  0.019662  0.059744 -0.005697 -0.002566 -0.028674
438 -0.005515  0.050680 -0.015906 -0.067642  0.049341  0.079165 -0.028674
439  0.041708  0.050680 -0.015906  0.017293 -0.037344 -0.013840 -0.024993
440 -0.045472 -0.044642  0.039062  0.001215  0.016318  0.015283 -0.028674
441 -0.045472 -0.044642 -0.073030 -0.081413  0.083740  0.027809  0.173816

           s4        s5        s6  target
0   -0.002592  0.019907 -0.017646   151.0
1   -0.039493 -0.068332 -0.092204    75.0
2   -0.002592  0.002861 -0.025930   141.0
3    0.034309  0.022688 -0.009362   206.0
4   -0.002592 -0.031988 -0.046641   135.0
..        ...       ...       ...     ...
437 -0.002592  0.031193  0.007207   178.0
438  0.034309 -0.018114  0.044485   104.0
439 -0.011080 -0.046883  0.015491   132.0
440  0.026560  0.044529 -0.025930   220.0
441 -0.039493 -0.004222  0.003064    57.0

[442 rows x 11 columns]
```

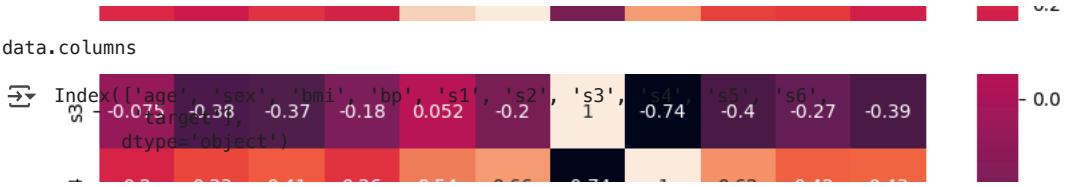
```
data.describe()
```

	age	sex	bmi	bp	s1	s2	s3	s4	s5	
count	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	4.4
mean	-2.511817e-19	1.230790e-17	-2.245564e-16	-4.797570e-17	-1.381499e-17	3.918434e-17	-5.777179e-18	-9.042540e-18	9.293722e-17	1.
std	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	4.
min	-1.072256e-01	-4.464164e-02	-9.027530e-02	-1.123988e-01	-1.267807e-01	-1.156131e-01	-1.023071e-01	-7.639450e-02	-1.260971e-01	·
25%	-3.729927e-02	-4.464164e-02	-3.422907e-02	-3.665608e-02	-3.424784e-02	-3.035840e-02	-3.511716e-02	-3.949338e-02	-3.324559e-02	·
50%	5.383060e-03	-4.464164e-02	-7.283766e-03	-5.670422e-03	-4.320866e-03	-3.819065e-03	-6.584468e-03	-2.592262e-03	-1.947171e-03	·

```
import seaborn as sns
import matplotlib.pyplot as plt


#Get coorelation in-between the features
corr=data.corr()
plt.figure(figsize=(10,8))
sns.heatmap(corr,annot=True)
plt.show()
```




```
(module) preprocessing
#Feature Importance
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import train_test_split
```



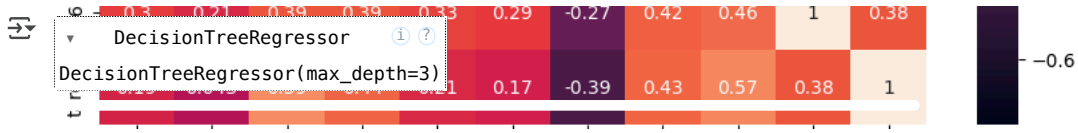

```
data.columns
```

```
Index(['age', 'sex', 'bmi', 'bp', 's1', 's2', 's3', 's4', 's5', 's6',
       'target'],
      dtype='object')
```

```
X_train,X_test,y_train,y_test=train_test_split(data.drop('target',axis=1),data['target'],test_size=0.2,random_state=42)
```

```
tree=DecisionTreeRegressor(max_depth=3)
tree.fit(X_train,y_train)
```

```
DecisionTreeRegressor(max_depth=3)
```




```
importance=pd.Series(tree.feature_importances_,index=X_train.columns)
importance
```

	0
age	0.000000
sex	0.000000
bmi	0.711521
bp	0.000000
s1	0.000000
s2	0.021410
s3	0.000000
s4	0.023996
s5	0.178897
s6	0.064176

dtype: float64