HexnBit
Skilling for Future

Artificial Intelligence and Machine Learning

Project Assignment

Summer- 2021

Car Model Acceptance Using AI-ML Models

DECLARATION

This is to certify that the 6-week AIML Summer Training project entitled *"Car Model Acceptance Using AI-ML Models"* is a bonafide work carried out by ***Satwik I Naik*** during the internship period of **Jun 21-Aug 21.** It is certified that all corrections/suggestions indicated for project have been incorporated in the report deposited in the institution. The project report has been approved as it satisfies the academic requirements in respect of internship work.

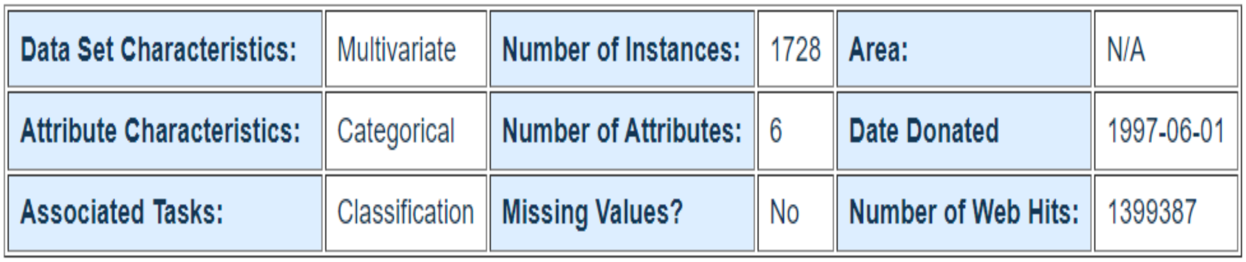
Name: Satwik I Naik (GROUP11)

Course: - 6-week AIML Summer Training

Mentor: - Gagan Singh

INTRODUCTION

Derived from simple hierarchical decision model, this database may be useful for testing constructive induction and structure discovery methods.



Data Set Characteristics:	Multivariate	**Number of Instances:**	1728	**Area:**	N/A
Attribute Characteristics:	Categorical	**Number of Attributes:**	6	**Date Donated**	1997-06-01
Associated Tasks:	Classification	**Missing Values?**	No	**Number of Web Hits:**	1399387

Source:

Creator:

Marko Bohanec

Donors:

1. Marko Bohanec (marko.bohanec '@' ijs.si)
2. Blaz Zupan (blaz.zupan '@' ijs.si)

Data Set Information:

Car Evaluation Database was derived from a simple hierarchical decision model originally developed for the demonstration of DEX, M. Bohanec, V. Rajkovic: Expert system for decision making. Sistemica 1(1), pp. 145-157, 1990.). The model evaluates cars according to the following concept structure:

CAR acceptability
. PRICE overall price
. . buying price
. . maint price of the maintenance
. TECH technical characteristics
. . COMFORT comfort
. . . doors number of doors
. . . persons capacity in terms of persons to carry
. . . lug_boot the size of luggage boot
. . safety estimated safety of the car

Input attributes are printed in lowercase. Besides the target concept (CAR), the model includes three intermediate concepts: PRICE, TECH, COMFORT. Every concept is in the original model related to its lower level descendants by a set of examples (for these examples sets see [Web Link]).

The Car Evaluation Database contains examples with the structural information removed, i.e., directly relates CAR to the six input attributes: buying, maint, doors, persons, lug_boot, safety.

Because of known underlying concept structure, this database may be particularly useful for testing constructive induction and structure discovery methods.

Attribute Information:

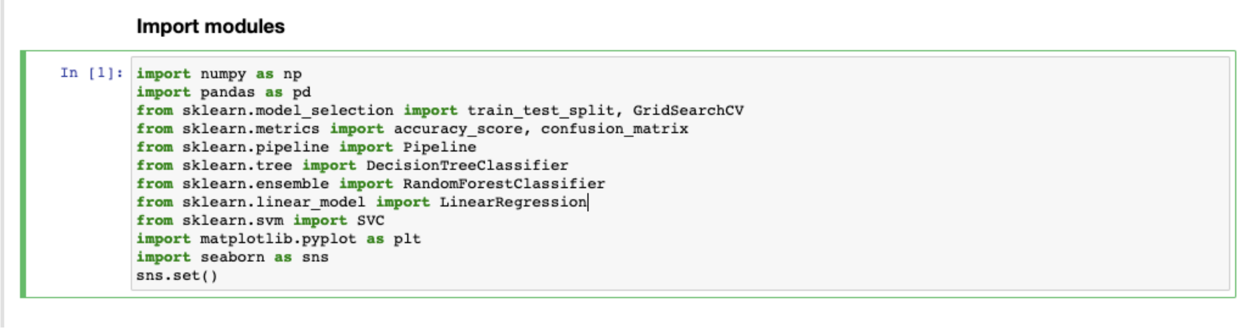
Class Values:

unacc, acc, good, vgood

Attributes:

buying: vhigh, high, med, low.
maint: vhigh, high, med, low.
doors: 2, 3, 4, 5more.
persons: 2, 4, more.
lug_boot: small, med, big.
safety: low, med, high.

Importing Modules

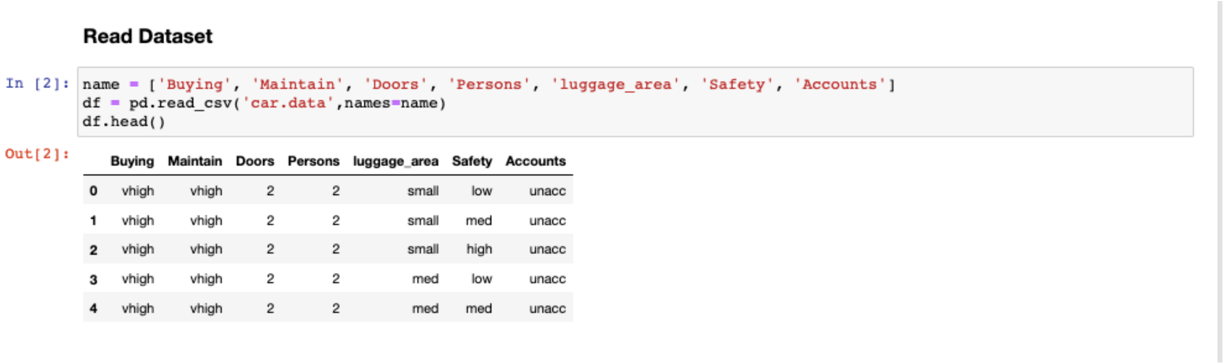


Import modules

In [1]:

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.pipeline import Pipeline
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LinearRegression
from sklearn.svm import SVC
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
```

Load the dataset



Read Dataset

In [2]:

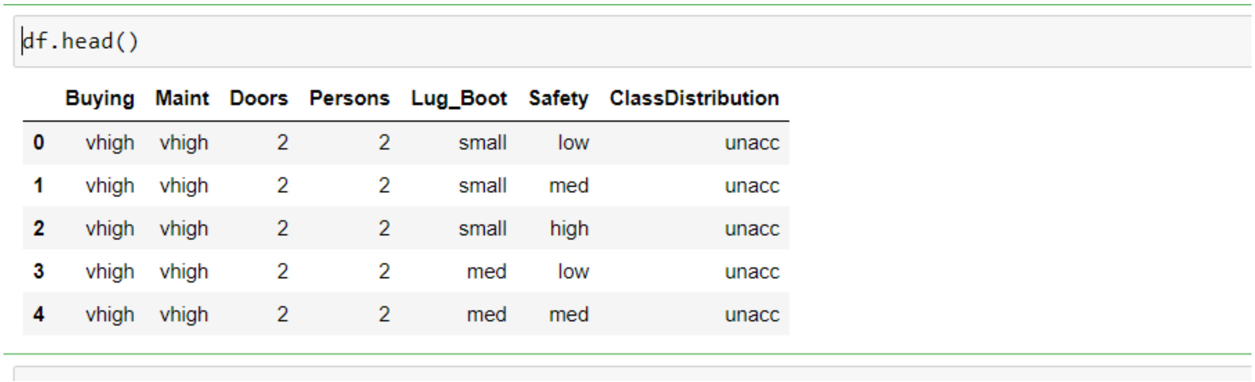
```
name = ['Buying', 'Maintain', 'Doors', 'Persons', 'luggage_area', 'Safety', 'Accounts']
df = pd.read_csv('car.data',names=name)
df.head()
```

Out[2]:

	Buying	Maintain	Doors	Persons	luggage_area	Safety	Accounts
0	vhigh	vhigh	2	2	small	low	unacc
1	vhigh	vhigh	2	2	small	med	unacc
2	vhigh	vhigh	2	2	small	high	unacc
3	vhigh	vhigh	2	2	med	low	unacc
4	vhigh	vhigh	2	2	med	med	unacc

Performing EDA

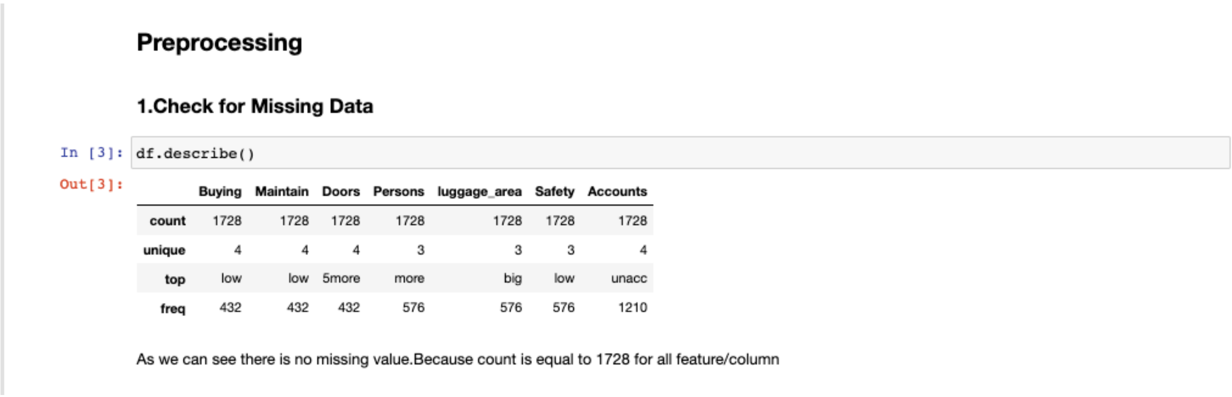
Look first five records:



```
df.head()
```

	Buying	Maint	Doors	Persons	Lug_Boot	Safety	ClassDistribution
0	vhigh	vhigh	2	2	small	low	unacc
1	vhigh	vhigh	2	2	small	med	unacc
2	vhigh	vhigh	2	2	small	high	unacc
3	vhigh	vhigh	2	2	med	low	unacc
4	vhigh	vhigh	2	2	med	med	unacc

Checking for Missing Data



Preprocessing

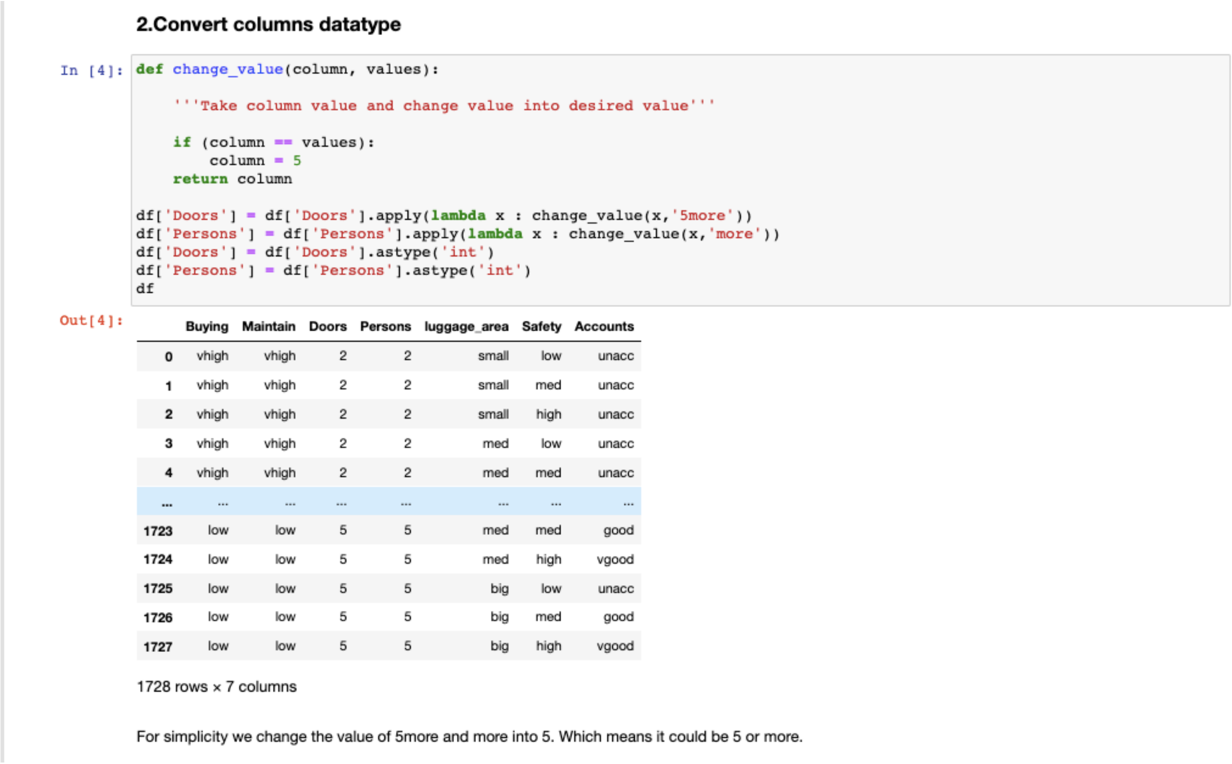
1.Check for Missing Data

```
In [3]: df.describe()
Out[3]:
```

	Buying	Maintain	Doors	Persons	luggage_area	Safety	Accounts
count	1728	1728	1728	1728	1728	1728	1728
unique	4	4	4	3	3	3	4
top	low	low	5more	more	big	low	unacc
freq	432	432	432	576	576	576	1210

As we can see there is no missing value.Because count is equal to 1728 for all feature/column

Convert columns datatype



2.Convert columns datatype

```
In [4]: def change_value(column, values):

            '''Take column value and change value into desired value'''

            if (column == values):
                column = 5
            return column

        df['Doors'] = df['Doors'].apply(lambda x : change_value(x,'5more'))
        df['Persons'] = df['Persons'].apply(lambda x : change_value(x,'more'))
        df['Doors'] = df['Doors'].astype('int')
        df['Persons'] = df['Persons'].astype('int')
        df
```

Out[4]:

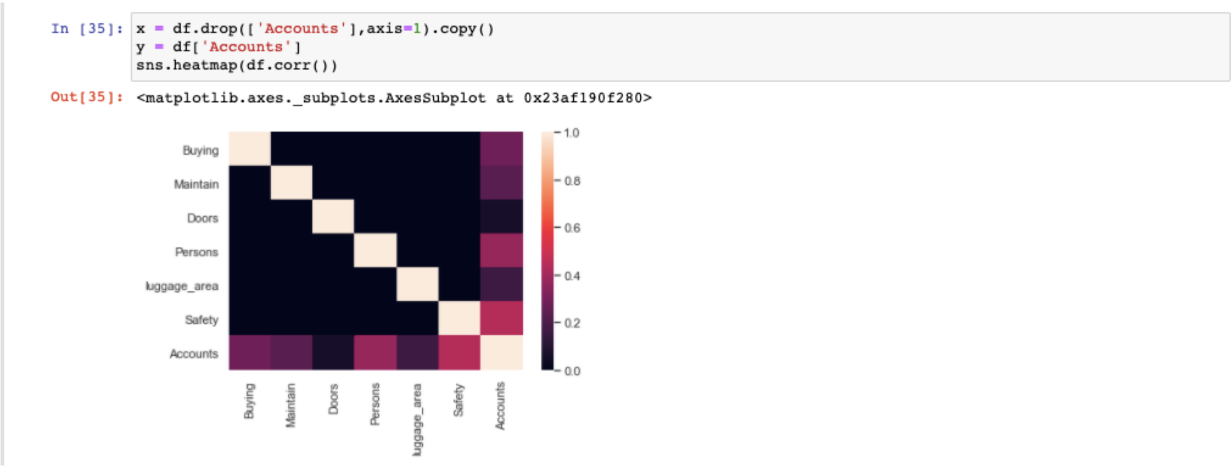
	Buying	Maintain	Doors	Persons	luggage_area	Safety	Accounts
0	vhigh	vhigh	2	2	small	low	unacc
1	vhigh	vhigh	2	2	small	med	unacc
2	vhigh	vhigh	2	2	small	high	unacc
3	vhigh	vhigh	2	2	med	low	unacc
4	vhigh	vhigh	2	2	med	med	unacc
...	...	...	...	...	...	...	...
1723	low	low	5	5	med	med	good
1724	low	low	5	5	med	high	vgood
1725	low	low	5	5	big	low	unacc
1726	low	low	5	5	big	med	good
1727	low	low	5	5	big	high	vgood

1728 rows × 7 columns

For simplicity we change the value of 5more and more into 5. Which means it could be 5 or more.

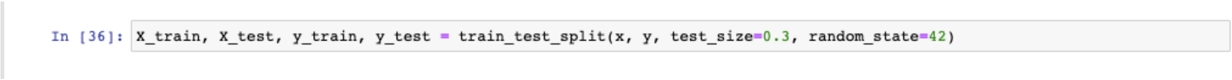
DATA VISUALIZATION

Correalation:



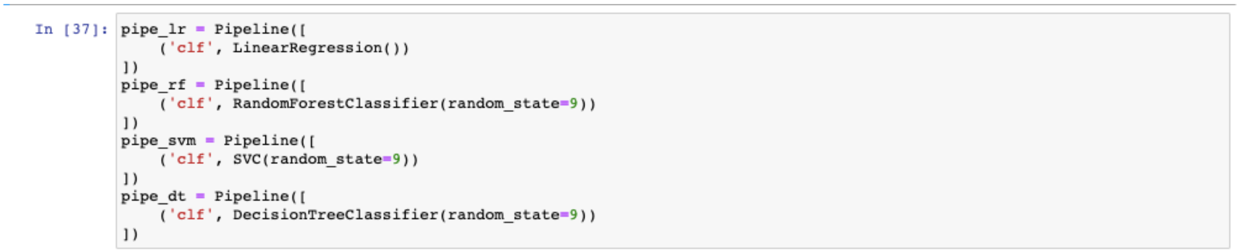
```
In [35]: x = df.drop(['Accounts'],axis=1).copy()
         y = df['Accounts']
         sns.heatmap(df.corr())

Out[35]: <matplotlib.axes._subplots.AxesSubplot at 0x23af190f280>
```

Split training and testing sets



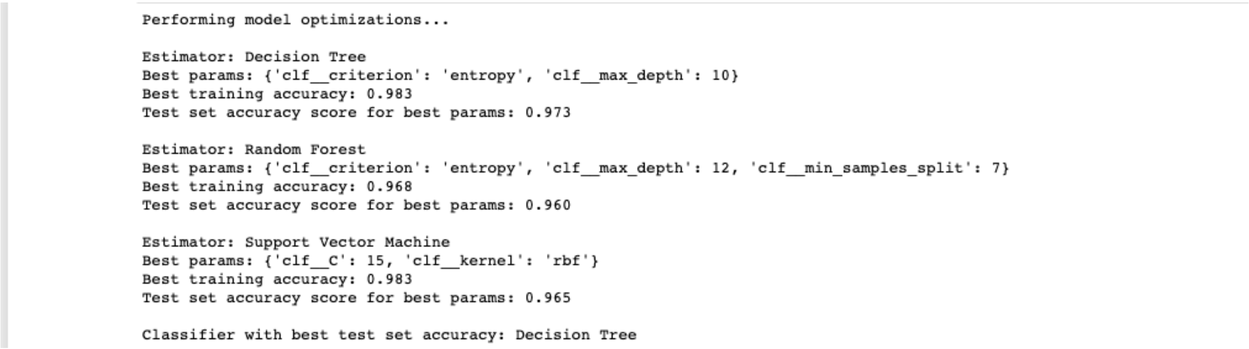
```
In [36]: X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=42)
```

Apply different Classification Algorithms and tune them



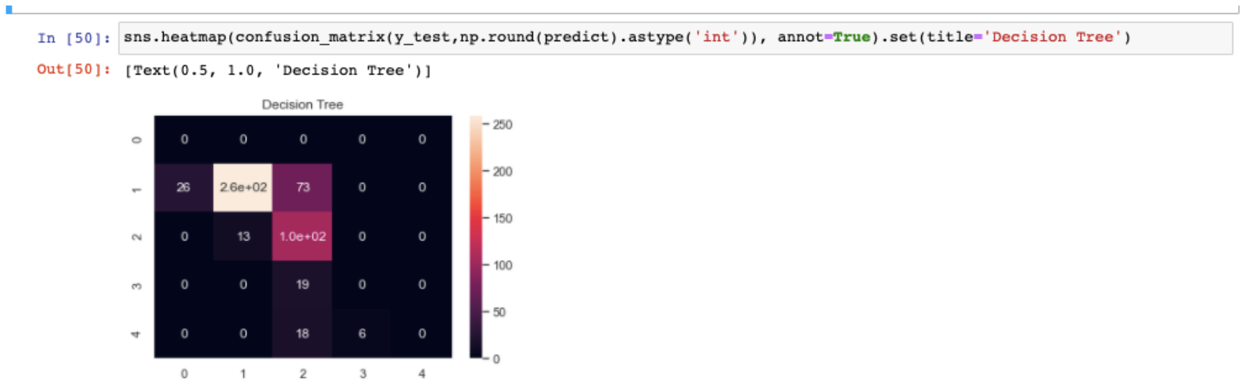
```
In [37]: pipe_lr = Pipeline([
             ('clf', LinearRegression())
         ])
         pipe_rf = Pipeline([
             ('clf', RandomForestClassifier(random_state=9))
         ])
         pipe_svm = Pipeline([
             ('clf', SVC(random_state=9))
         ])
         pipe_dt = Pipeline([
             ('clf', DecisionTreeClassifier(random_state=9))
         ])
```

records of y_test_pred and y_test: Model Optimization



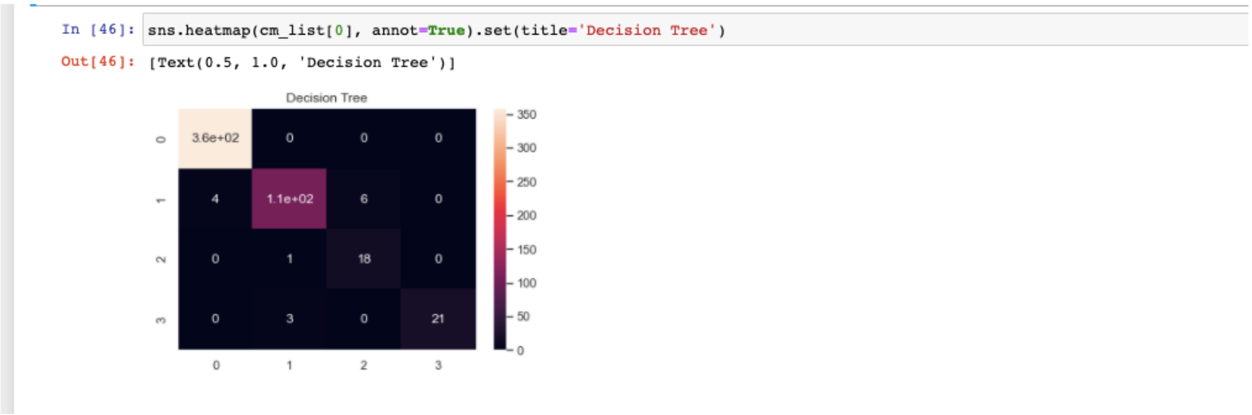
```
Performing model optimizations...

Estimator: Decision Tree
Best params: {'clf__criterion': 'entropy', 'clf__max_depth': 10}
Best training accuracy: 0.983
Test set accuracy score for best params: 0.973

Estimator: Random Forest
Best params: {'clf__criterion': 'entropy', 'clf__max_depth': 12, 'clf__min_samples_split': 7}
Best training accuracy: 0.968
Test set accuracy score for best params: 0.960

Estimator: Support Vector Machine
Best params: {'clf__C': 15, 'clf__kernel': 'rbf'}
Best training accuracy: 0.983
Test set accuracy score for best params: 0.965

Classifier with best test set accuracy: Decision Tree
```

Visually compare the performance of all classifiers



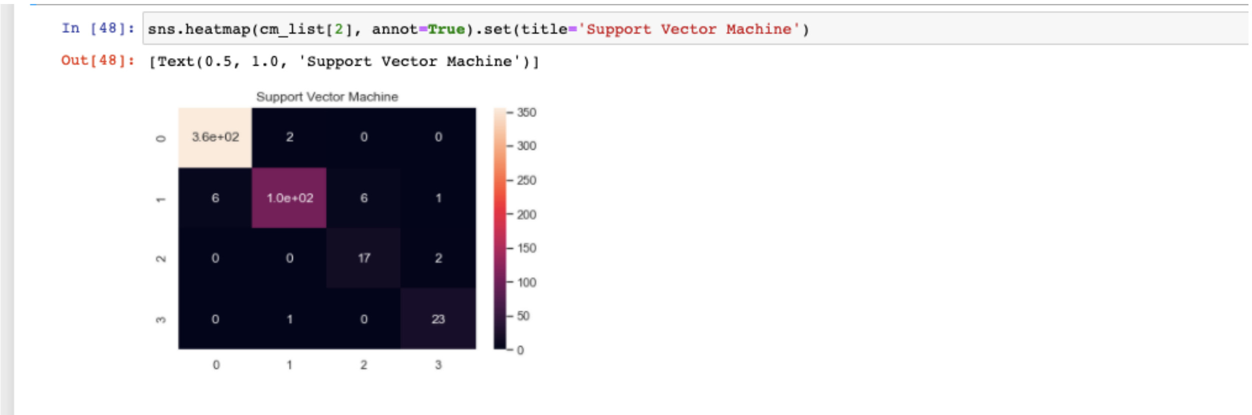
```
In [50]: sns.heatmap(confusion_matrix(y_test,np.round(predict).astype('int')), annot=True).set(title='Decision Tree')
Out[50]: [Text(0.5, 1.0, 'Decision Tree')]
```

Linear regression has Test accuracy score of = 70.0%



```
In [46]: sns.heatmap(cm_list[0], annot=True).set(title='Decision Tree')
Out[46]: [Text(0.5, 1.0, 'Decision Tree')]
```

Random Forest Tree has Test accuracy score of = 96.0%



```
In [48]: sns.heatmap(cm_list[2], annot=True).set(title='Support Vector Machine')
```

```
Out[48]: [Text(0.5, 1.0, 'Support Vector Machine')]
```