Support Vector Regression:

If Kernel='linear'

```
from sklearn.svm import SVR
regressor=SVR(kernel="linear",C=0.1)
regressor=regressor.fit(X_train,y_train)
```

```
C:\Anaconda\Lib\site-packages\sklearn\utils\validation.py:1184: DataConversionWarning: A column-vector y was passed when a 1d a
rray was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

```
y_pred=regressor.predict(X_test)
```

```
from sklearn.metrics import r2_score
r_score=r2_score(y_test,y_pred)
r_score
```

```
0.9408798822563795
```

This kernel looks better model since r_score value is near to 1.

If C value is changed to C=1.0 then r_score value decreases to 0.8, Similarly for below kernel types value is changed based on C value.

If Kernel='rbf'

```
from sklearn.svm import SVR
regressor=SVR(kernel="rbf",C=0.1)
regressor=regressor.fit(X_train,y_train)
```

```
C:\Anaconda\Lib\site-packages\sklearn\utils\validation.py:1184: DataConversionWarning: A column-vector y was passed when a 1d a
rray was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

```
y_pred=regressor.predict(X_test)
```

```
from sklearn.metrics import r2_score
r_score=r2_score(y_test,y_pred)
r_score
```

```
-0.05747583240851273
```

This model doesn't look good and it as negative value.

If Kernel='poly'

```
from sklearn.svm import SVR
regressor=SVR(kernel="poly",C=0.1)
regressor=regressor.fit(X_train,y_train)
```

```
C:\Anaconda\Lib\site-packages\sklearn\utils\validation.py:1184: DataConversionWarning: A column-vector y was passed when a 1d a
rray was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

```
y_pred=regressor.predict(X_test)
```

```
from sklearn.metrics import r2_score
r_score=r2_score(y_test,y_pred)
r_score
```

```
-0.05683096395558507
```

This model doesn't look good and it as negative value.

If Kernel='sigmoid'

```
from sklearn.svm import SVR
regressor=SVR(kernel="sigmoid",C=0.1)
regressor=regressor.fit(X_train,y_train)
```

```
C:\Anaconda\Lib\site-packages\sklearn\utils\validation.py:1184: DataConversionWarning: A column-vector y was passed when a 1d a
rray was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

```
y_pred=regressor.predict(X_test)
```

```
from sklearn.metrics import r2_score
r_score=r2_score(y_test,y_pred)
r_score
```

```
-0.05749402523392089
```

This model doesn't look good and it as negative value.

If Kernel='precomputed'

```
from sklearn.svm import SVR
regressor=SVR(kernel="precomputed",C=0.1)
regressor=regressor.fit(X_train,y_train)
```

```
C:\Anaconda\Lib\site-packages\sklearn\utils\validation.py:1184: DataConversionWarning: A column-vector y was passed when a 1d a
rray was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[9], line 3
      1 from sklearn.svm import SVR
      2 regressor=SVR(kernel="precomputed",C=0.1)
----> 3 regressor=regressor.fit(X_train,y_train)

File C:\Anaconda\Lib\site-packages\sklearn\base.py:1151, in _fit_context.<locals>.decorator.<locals>.wrapper(estimator, *args,
**kwargs)
   1144     estimator._validate_params()
   1146 with config_context(
   1147     skip_parameter_validation=(
   1148         prefer_skip_nested_validation or global_skip_validation
   1149     )
   1150 ):
-> 1151     return fit_method(estimator, *args, **kwargs)

File C:\Anaconda\Lib\site-packages\sklearn\svm\_base.py:215, in BaseLibSVM.fit(self, X, y, sample_weight)
    209     raise ValueError(
    210         "X and y have incompatible shapes.\n"
    211         + "X has %s samples, but y has %s." % (n_samples, y.shape[0])
    212     )
    214 if self.kernel == "precomputed" and n_samples != X.shape[1]:
--> 215     raise ValueError(
    216         "Precomputed matrix must be a square matrix."
```

```
217         " Input is a {}x{} matrix.".format(X.shape[0], X.shape[1])
218     )
220 if sample_weight.shape[0] > 0 and sample_weight.shape[0] != n_samples:
221     raise ValueError(
222         "sample_weight and X have incompatible shapes: "
223         "%r vs %r\n"
(...)
226         % (sample_weight.shape, X.shape)
227     )

ValueError: Precomputed matrix must be a square matrix. Input is a 35x5 matrix.
```

Since this input has 5 columns and 35 rows this cannot be done using "precomputed kernel"

Support Vector Machine:

Sl.no	C value	Linear(r_score value)	RBF (r_score value)	Poly(r_score value)	Sigmoid(r_score value)
1	C=0.1	0.9408	-0.0574	-0.0568	-0.0574
2	C=1.0	0.8774	-0.0573	-0.0508	-0.0575
3	C=10	0.5380	-0.0558	0.0253	-0.0576
4	C=100	-107.9775	-0.0302	0.4656	-0.0587

From above list , Linear, C=0.1 r_score value is 0.9408 is the best model