DAY3: ML_Linear Regression Interview

- By PARIMAL A
- PRACTICAL

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
from sklearn.linear_model import LinearRegression
```

```
!pip install scikit-learn==1.1.1
```

```
    self[name] = toklist
  File "/usr/local/lib/python3.10/dist-packages/pip/_vendor/py
    self._tokdict[k] = self._tokdict.get(k, list()) + [
KeyboardInterrupt

During handling of the above exception, another exception occu

Traceback (most recent call last):
  File "/usr/lib/python3.10/logging/__init__.py", line 1732, i
    return self._cache[level]
KeyError: 50

During handling of the above exception, another exception occu

Traceback (most recent call last):
  File "/usr/local/bin/pip3", line 8, in <module>
    sys.exit(main())
  File "/usr/local/lib/python3.10/dist-packages/pip/_internal/
    return command.main(cmd_args)
  File "/usr/local/lib/python3.10/dist-packages/pip/_internal/
    return self._main(args)
  File "/usr/local/lib/python3.10/dist-packages/pip/_internal/
    return run(options, args)
  File "/usr/local/lib/python3.10/dist-packages/pip/_internal/
    logger.critical("Operation cancelled by user")
  File "/usr/lib/python3.10/logging/__init__.py", line 1523, i
    if self.isEnabledFor(CRITICAL):
  File "/usr/lib/python3.10/logging/__init__.py", line 1734, i
    _acquireLock()
  File "/usr/lib/python3.10/logging/__init__.py", line 226, in
    _lock.acquire()
```

```
from sklearn.datasets import load_boston # housepricedataset
```

```
load_boston()
```

```
# or you canuse
#import seaborn as sns
## Load the Boston housing dataset
# boston = sns.load_dataset('boston')
```

```
df=load_boston()
df
```

```
        18.9, 21.7, 20.4, 18.2, 19.9, 23.1, 17.5, 20.2, 18.2,
13.6, 19.6,
        15.2, 14.5, 15.6, 13.9, 16.6, 14.8, 18.4, 21. , 12.7,
14.5, 13.2,
        13.1, 13.5, 18.9, 20. , 21. , 24.7, 30.8, 34.9, 26.6,
25.3, 24.7,
        21.2, 19.3, 20. , 16.6, 14.4, 19.4, 19.7, 20.5, 25. ,
23.4, 18.9,
        35.4, 24.7, 31.6, 23.3, 19.6, 18.7, 16. , 22.2, 25. ,
33. , 23.5,
        19.4, 22. , 17.4, 20.9, 24.2, 21.7, 22.8, 23.4, 24.1,
21.4, 20. ,
        20.8, 21.2, 20.3, 28. , 23.9, 24.8, 22.9, 23.9, 26.6,
22.5, 22.2,
```

```
       22.5, 24.4,
              20. , 21.7, 19.3, 22.4, 28.1, 23.7, 25. , 23.3, 28.7,
       21.5, 23. ,
              26.7, 21.7, 27.5, 30.1, 44.8, 50. , 37.6, 31.6, 46.7,
       31.5, 24.3,
              31.7, 41.7, 48.3, 29. , 24. , 25.1, 31.5, 23.7, 23.3,
       22. , 20.1,
              22.2, 23.7, 17.6, 18.5, 24.3, 20.5, 24.5, 26.2, 24.4,
       24.8, 29.6,
              42.8, 21.9, 20.9, 44. , 50. , 36. , 30.1, 33.8, 43.1,
       48.8, 31. ,
              36.5, 22.8, 30.7, 50. , 43.5, 20.7, 21.1, 25.2, 24.4,
       35.2, 32.4,
              32. , 33.2, 33.1, 29.1, 35.1, 45.4, 35.4, 46. , 50. ,
       32.2, 22. ,
              20.1, 23.2, 22.3, 24.8, 28.5, 37.3, 27.9, 23.9, 21.7,
       28.6, 27.1,
              20.3, 22.5, 29. , 24.8, 22. , 26.4, 33.1, 36.1, 28.4,
       33.4, 28.2,
              22.8, 20.3, 16.1, 22.1, 19.4, 21.6, 23.8, 16.2, 17.8,
       19.8, 23.1,
              21. , 23.8, 23.1, 20.4, 18.5, 25. , 24.6, 23. , 22.2,
       19.3, 22.6,
```

```
dataset=pd.DataFrame(df.data)
```

```
dataset
```

	0	1	2	3	4	5	6	7	8	9
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0
...	...	...	...	...	...	...	...	...	...	...
501	0.06263	0.0	11.93	0.0	0.573	6.593	69.1	2.4786	1.0	273.0
502	0.04527	0.0	11.93	0.0	0.573	6.120	76.7	2.2875	1.0	273.0
503	0.06076	0.0	11.93	0.0	0.573	6.976	91.0	2.1675	1.0	273.0
504	0.10959	0.0	11.93	0.0	0.573	6.794	89.3	2.3889	1.0	273.0
505	0.04741	0.0	11.93	0.0	0.573	6.030	80.8	2.5050	1.0	273.0

506 rows × 13 columns

Next steps: Generate code with dataset View recommended plots

```
dataset.columns=df.feature_names
```

```
dataset.head()
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TA
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.

Next steps: Generate code with dataset View recommended plots

```
df.target.shape
```

```
(506,)
```

```
df.feature_names.shape
```

```
(13,)
```

```
## Independent features =X and dependent features=y
X=dataset
y=df.target
```

```
X
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	2
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	2
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	2
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	2
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	2
...	...	...	...	...	...	...	...	...	...	
501	0.06263	0.0	11.93	0.0	0.573	6.593	69.1	2.4786	1.0	2
502	0.04527	0.0	11.93	0.0	0.573	6.120	76.7	2.2875	1.0	2
503	0.06076	0.0	11.93	0.0	0.573	6.976	91.0	2.1675	1.0	2
504	0.10959	0.0	11.93	0.0	0.573	6.794	89.3	2.3889	1.0	2
505	0.04741	0.0	11.93	0.0	0.573	6.030	80.8	2.5050	1.0	2

506 rows × 13 columns

Next steps: Generate code with dataset View recommended plots

Train test split

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.
```

```
X_train
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD
5	0.02985	0.0	2.18	0.0	0.458	6.430	58.7	6.0622	3.0
116	0.13158	0.0	10.01	0.0	0.547	6.176	72.5	2.7301	6.0
45	0.17142	0.0	6.91	0.0	0.448	5.682	33.8	5.1004	3.0
16	1.05393	0.0	8.14	0.0	0.538	5.935	29.3	4.4986	4.0
468	15.57570	0.0	18.10	0.0	0.580	5.926	71.0	2.9084	24.0
...	...	...	...	...	...	...	...	...	...
106	0.17120	0.0	8.56	0.0	0.520	5.836	91.9	2.2110	5.0
270	0.29916	20.0	6.96	0.0	0.464	5.856	42.1	4.4290	3.0
348	0.01501	80.0	2.01	0.0	0.435	6.635	29.7	8.3440	4.0
435	11.16040	0.0	18.10	0.0	0.740	6.629	94.6	2.1247	24.0
102	0.22876	0.0	8.56	0.0	0.520	6.405	85.4	2.7147	5.0

354 rows × 13 columns

Next steps: Generate code with X_train View recommended plots

```
X_train.shape
```

```
(354, 13)
```

```
X_test.shape
```

```
(152, 13)
```

Standardizing the dataset

```
## standardizing the dataset : mean=0 std deviation =1
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
```

```
X_train=scaler.fit_transform(X_train)
```

```
X_test=scaler.transform(X_test)
```

Model Training

```
from sklearn.linear_model import LinearRegression
##cross validation
from sklearn.model_selection import cross_val_score
```

```
regr=LinearRegression()
regr.fit(X_train,y_train) #training model x,y
```

```
LinearRegression()
```

```
mse=cross_val_score(regr,X_train,y_train,scoring='neg_mean_squared_e
```

```
np.mean(mse) # shouldbeless
```

```
-25.55066079166079
```

```
# Prediction
reg_pred=regr.predict(X_test)
```

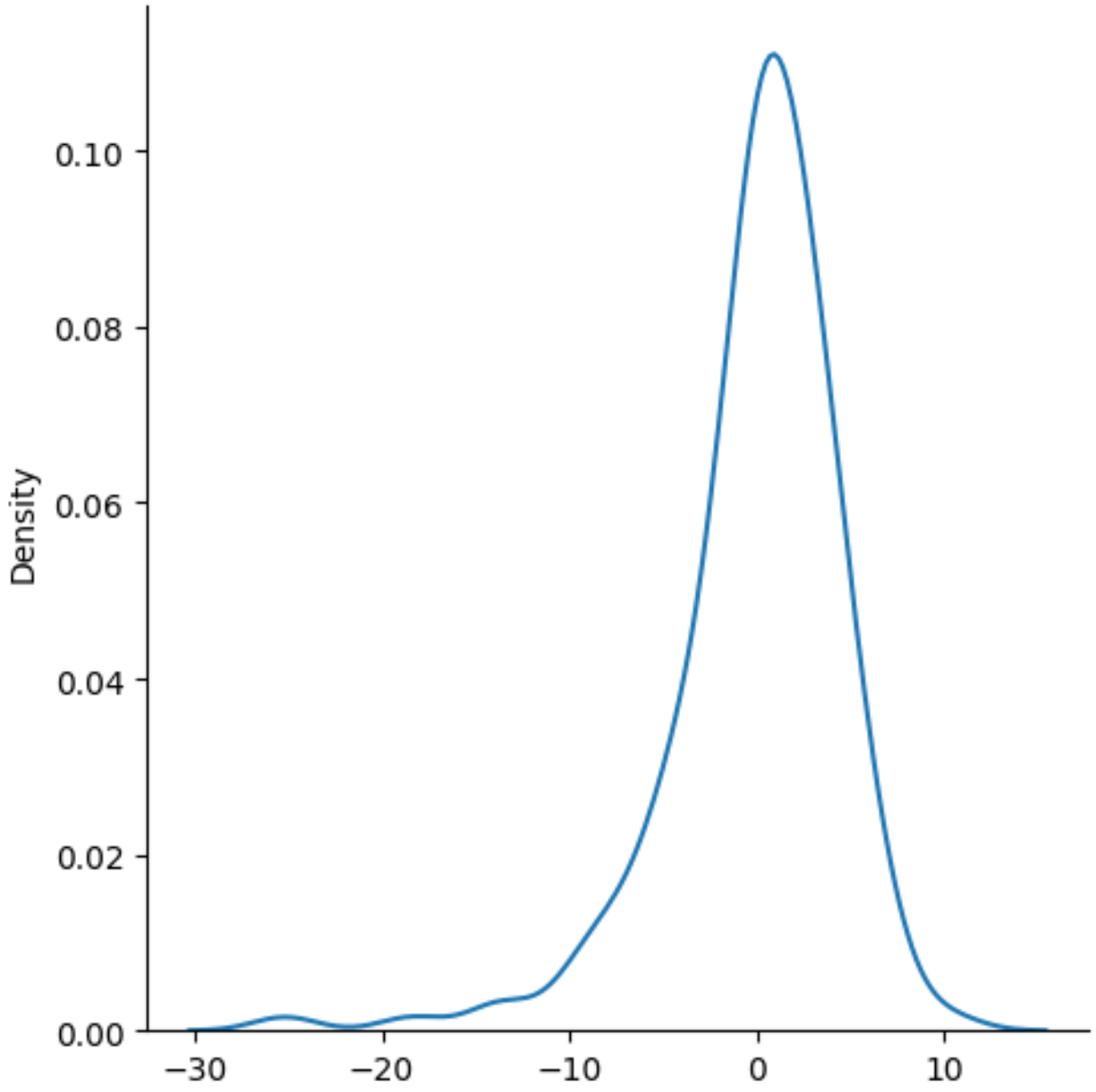
```
reg_pred # ypred=y=mx+c
```

```
array([28.64896005, 36.49501384, 15.4111932 , 25.40321303,
18.85527988,
       23.14668944, 17.3921241 , 14.07859899, 23.03692679,
20.59943345,
       24.82286159, 18.53057049, -6.86543527, 21.80172334,
19.22571177,
       26.19191985, 20.27733882,  5.61596432, 40.44887974,
17.57695918,
       27.44319095, 30.1715964 , 10.94055823, 24.02083139,
18.07693812,
       15.934748  , 23.12614028, 14.56052142, 22.33482544,
19.3257627 ,
       22.16564973, 25.19476081, 25.31372473, 18.51345025,
16.6223286 ,
       17.50268505, 30.94992991, 20.19201752, 23.90440431,
24.86975466,
       13.93767876, 31.82504715, 42.56978796, 17.62323805,
27.01963242,
       17.19006621, 13.80594006, 26.10356557, 20.31516118,
30.08649576,
       21.3124053 , 34.15739602, 15.60444981, 26.11247588,
39.31613646,
       22.99282065, 18.95764781, 33.05555669, 24.85114223,
12.91729352,
       22.68101452, 30.80336295, 31.63522027, 16.29833689,
21.07379993,
       16.57699669, 20.36362023, 26.15615896, 31.06833034,
11.98679953,
       20.42550472, 27.55676301, 10.94316981, 16.82660609,
23.92909733,
        5.28065815, 21.43504661, 41.33684993, 18.22211675,
9.48269245,
       21.19857446, 12.95001331, 21.64822797,  9.3845568 ,
23.06060014,
       31.95762512, 19.16662892, 25.59942257, 29.35043558,
20.13138581,
       25.57297369,  5.42970803, 20.23169356, 15.1949595 ,
14.03241742,
       20.91078077, 24.82249135, -0.47712079, 13.70520524,
15.69525576,
```

```
       22.06972676, 24.64152943, 10.7382866 , 19.68622564,
23.63678009,
       12.07974981, 18.47894211, 25.52713393, 20.93461307,
24.6955941 ,
        7.59054562, 19.01046053, 21.9444339 , 27.22319977,
32.18608828,
       15.27826455, 34.39190421, 12.96314168, 21.01681316,
28.57880911,
       15.86300844, 24.85124135,  3.37937111, 23.90465773,
25.81792146,
       23.11020547, 25.33489201, 33.35545176, 20.60724498,
38.4772665 ,
       13.97398533, 25.21923987, 17.80946626, 20.63437371,
9.80267398,
       21.07953576, 22.3378417 , 32.32381854, 31.48694863,
15.46621287,
       16.86242766, 28.99330526, 24.95467894, 16.73633557,
```

```
import seaborn as sns
sns.displot(reg_pred-y_test,kind='kde') # ypred- yactual
```

```
<seaborn.axisgrid.FacetGrid at 0x7bcf6b9da710>
```



Almost -10to -30 are more paameters fall so differenceisvery less

```
from sklearn.metrics import r2_score
```

```
score=r2_score(reg_pred,y_test)
```

```
score
```

```
0.6693702691495591
```

```
#save the model
```

```
import joblib

# Save the model to a file
joblib_file = "linear_regression_model.pkl"
joblib.dump(regr, joblib_file)
print(f"Model saved to {joblib_file}")
```

```
Model saved to linear_regression_model.pkl
```

```
import pickle

# Save the model to a file
pickle_file = "linear_regression_modelpk.pkl"
with open(pickle_file, 'wb') as file:
    pickle.dump(regr, file)
print(f"Model saved to {pickle_file}")
```

```
Model saved to linear_regression_modelpk.pkl
```

```
# Compare predictions with actual values
print("Predicted values:", reg_pred) #ypredictedvalues
print("Actual values:", y_test) #yactualvalues
```

Show hidden output

```
# Create a DataFrame to compare predictions with actual values
results_df = pd.DataFrame({'Actual': y_test, 'Predicted': reg_pred})
```

```
print(results_df)
```

```
      Actual  Predicted
0       23.6  28.648960
1       32.4  36.495014
2       13.6  15.411193
3       22.8  25.403213
4       16.1  18.855280
```

```
..      ...         ...
147    17.1  17.403672
```