LOAD DATA

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

```
Start coding or generate with AI.
```

Double-click (or enter) to edit

```
Start coding or generate with AI.
```

```
url = 'https://raw.githubusercontent.com/nitindig/datascience/master/re

# Importing the dataset
df = pd.read_csv(url)
```

DATA PREPERATION Data separation

```
y=df["Salary"]
x=df.drop("Salary",axis=1)
```

Data Splitting

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random
```

Model Building Linear Regression

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

LinearRegression() In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook. On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

Applying the model to make prediction

```
y_train_lr_pred=lr.predict(x_train)
y_test_lr_pred=lr.predict(x_test)

y_train_lr_pred
```

```
array([116180.82036723,  64030.39965754,  76136.74732229,
100349.44265179,
        53786.56701814,  74274.23229695,  56580.33955616,
68686.68722091,
       103143.21518981,  90105.61001239,  38886.44681538,
124562.13798128,
        54717.82453082,  47267.76442943,  81724.29239833,
82655.549911  ,
        61236.62711953,  56580.33955616, 110593.27529119,
45405.24940409,
        37023.93179003,  92899.38255041,  72411.7172716 ,
64030.39965754])
```

```
from sklearn.metrics import mean_squared_error,r2_score

lr_train_mse=mean_squared_error(y_train,y_train_lr_pred)
lr_train_r2=r2_score(y_train,y_train_lr_pred)

lr_test_mse=mean_squared_error(y_test,y_test_lr_pred)
lr_test_r2=r2_score(y_test,y_test_lr_pred)

lr_result=pd.DataFrame(["linear Regression",lr_train_mse,lr_train_r2,lr_
lr_result.columns=["method","training Mse","training r2","test Mse","te
lr_result
```

	method	training Mse	training r2	test Mse	test r2
0	linear Regression	36149670.118161	0.941195	12823412.298127	0.98817

Data Visualization

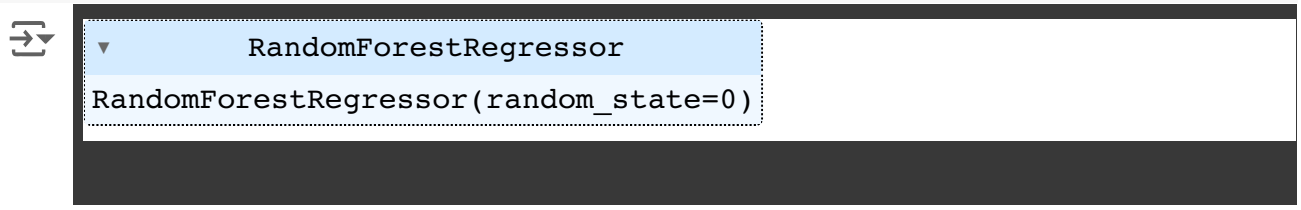
```
plt.scatter(x_train, y_train, color = 'red')
plt.plot(x_train,y_train_lr_pred,color = 'blue')
plt.title('Salary vs Experience (Training set)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()

#test set result

plt.scatter(x_test, y_test, color = 'red')
plt.plot(x_test,y_test_lr_pred,color = 'blue')
plt.title('Salary vs Experience (Test set)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()
```

Random Forest

```
from sklearn.ensemble import RandomForestRegressor
rf=RandomForestRegressor(random_state=0)
rf.fit(x_train,y_train)
```



RandomForestRegressor(random_state=0) In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook. On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

Applying the model to make prediction

Double-click (or enter) to edit

```
y_train_rf_pred=rf.predict(x_train)
y_test_rf_pred=rf.predict(x_test)
```

Evaluate Model Performence

```
from sklearn.metrics import mean_squared_error,r2_score

rf_train_mse=mean_squared_error(y_train,y_train_rf_pred)
rf_train_r2=r2_score(y_train,y_train_rf_pred)

rf_test_mse=mean_squared_error(y_test,y_test_rf_pred)
rf_test_r2=r2_score(y_test,y_test_rf_pred)

rf_result=pd.DataFrame(["RandomForest",rf_train_mse,rf_train_r2,rf_test_
rf_result.columns=["method","training Mse","training r2","test Mse","te
df=pd.concat([lr_result,rf_result]).reset_index(drop=True)
df
```

	method	training Mse	training r2	test Mse	test r2
0	linear Regression	36149670.118161	0.941195	12823412.298127	0.98817
1	RandomForest	8281022.529747	0.986529	22223542.924426	0.979497

Data Visuliztion