```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

data=pd.read_csv("/content/insurance_data.csv")
data
```

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500
1337	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

The main data mining problem is predicting insurances charges based on various factors such as age, sex, BMI (Body Mass Index), number of children, smoking status, and region. linear regression algorithm will be appropriate in predicting the insurance charges by creating a model. **The objective** would be to develop a predictive model that can generalize well to unseen data, allowing insurance companies to better estimate the charges for potential clients and adjust their pricing strategies accordingly.

```
print(data.head())
```

```
   age     sex     bmi  children smoker     region      charges
0   19  female  27.900         0    yes  southwest  16884.92400
1   18    male  33.770         1     no  southeast   1725.55230
2   28    male  33.000         3     no  southeast   4449.46200
3   33    male  22.705         0     no  northwest  21984.47061
4   32    male  28.880         0     no  northwest   3866.85520
```

```
print(data.isnull().sum())
```

```
age         0
sex         0
bmi         0
children    0
smoker      0
```

```
region      0
charges     0
dtype: int64
```

```
data['sex'] = data['sex'].map({'male': 0, 'female': 1})
data['smoker']=data['smoker'].map({'no':0,'yes':1})
data = pd.get_dummies(data, columns=['region'], drop_first=True, dtype=int)
print(data.head())
```

```
   age  sex     bmi  children  smoker      charges  region_northwest  \
0   19    1  27.900         0       1  16884.92400                 0
1   18    0  33.770         1       0   1725.55230                 0
2   28    0  33.000         3       0   4449.46200                 0
3   33    0  22.705         0       0  21984.47061                 1
4   32    0  28.880         0       0   3866.85520                 1

   region_southeast  region_southwest
0                 0                 1
1                 1                 0
2                 1                 0
3                 0                 0
4                 0                 0
```

We encode categorical variables sex ,smoker and region(one hot encoding We drop the first dummy variable to avoid multicollinearity issues

```
X = data.drop(columns=['charges'])
y = data['charges']
y.head()
```

```
0    16884.92400
1     1725.55230
2     4449.46200
3    21984.47061
4     3866.85520
Name: charges, dtype: float64
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
print(y_test.head())
```

```
764      9095.06825
887      5272.17580
890     29330.98315
1293     9301.89355
259     33750.29180
Name: charges, dtype: float64
```

```
model = LinearRegression()
model.fit(X_train, y_train)
predictions = pd.DataFrame(model.predict(X_test))
predictions.head()
```

	0
0	8969.550274
1	7068.747443
2	36858.410912
3	9454.678501
4	26973.173457

```
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print("Mean Absolute Error:", mae)
print("Mean Squared Error:", mse)
print("R-squared:", r2)
```

```
Mean Absolute Error: 4181.194473753645
Mean Squared Error: 33596915.85136143
R-squared: 0.7835929767120725
```

Mean Absolute Error (MAE): The MAE of approximately 4181.19 indicates the average absolute difference between the predicted insurance charges and the actual insurance charges across all observations in the dataset. **Mean Squared Erro**r (MSE): The MSE of approximately 33,596,915.85 represents the average of the squared differences between the predicted insurance charges and the actual charges. It penalizes larger errors more heavily than smaller ones. R-squared (R^2): The R-squared value of approximately 0.784 indicates the proportion of the variance in the insurance charges that is explained by the independent variables in the model. In other words, around 78.36% of the variability in insurance charges can be explained by the independent variables (BMI,region ,smoker,children and sex).

Double-click (or enter) to edit

```
params = {'fit_intercept': [True, False]}  # Hyperparameters to tune
grid_search = GridSearchCV(estimator=LinearRegression(), param_grid=params, cv=5, scoring='neg_mean_squared_error')
grid_search.fit(X_train, y_train)
```

```
▸ GridSearchCV
▸ estimator: LinearRegression
  ▸ LinearRegression
```

```
best_params = grid_search.best_params_
print("Best Hyperparameters:", best_params)
```

```
Best Hyperparameters: {'fit_intercept': True}
```

```
best_model = grid_search.best_estimator_
best_model
```

```
▾ LinearRegression
LinearRegression()
```

```
y_pred_best = best_model.predict(X_test)
mae_best = mean_absolute_error(y_test, y_pred_best)
mse_best = mean_squared_error(y_test, y_pred_best)
r2_best = r2_score(y_test, y_pred_best)

print("Best Model - Mean Absolute Error:", mae_best)
print("Best Model - Mean Squared Error:", mse_best)
print("Best Model - R-squared:", r2_best)
```

```
Best Model - Mean Absolute Error: 4181.194473753645
Best Model - Mean Squared Error: 33596915.85136143
Best Model - R-squared: 0.7835929767120725
```

R-squared (R^2): This metric measures the proportion of the variance in the charges from the independent variables. It ranges from 0 to 1, where 1 indicates a perfect fit. An R^2 of 0.78 suggests that your model explains about 78.36% of the variance in the charges, which is relatively good.