


```
import pandas as pd
from sklearn.linear_model import LogisticRegression
import numpy as np
```

```
insu=pd.read_csv('/content/task ins.csv')
```

```
insu.head()
```



	id	Gender	Age	Driving_License	Region_Code	Previously_Insured	Vehicle_Age	Vehicle_Damage	Annual_Premium	Policy_Sales_Channr
0	381110	Male	25	1	11	1	< 1 Year	No	35786	1
1	381111	Male	40	1	28	0	1-2 Year	Yes	33762	
2	381112	Male	47	1	28	0	1-2 Year	Yes	40050	1
3	381113	Male	24	1	27	1	< 1 Year	Yes	37356	1
4	381114	Male	27	1	28	1	< 1 Year	No	59097	1


```
insu.isnull().sum()
```



	0
id	0
Gender	0
Age	0
Driving_License	0
Region_Code	0
Previously_Insured	0
Vehicle_Age	0
Vehicle_Damage	0
Annual_Premium	0
Policy_Sales_Channel	0
Vintage	0

dtype: int64

```
insu.info()
```



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 127037 entries, 0 to 127036
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype
---  -
0   id                    127037 non-null int64
1   Gender                127037 non-null object
2   Age                  127037 non-null int64
3   Driving_License      127037 non-null int64
4   Region_Code          127037 non-null int64
5   Previously_Insured   127037 non-null int64
6   Vehicle_Age          127037 non-null object
7   Vehicle_Damage       127037 non-null object
8   Annual_Premium       127037 non-null int64
9   Policy_Sales_Channel 127037 non-null int64
10  Vintage              127037 non-null int64
dtypes: int64(8), object(3)
memory usage: 10.7+ MB
```

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
insu['Gen']=le.fit_transform(insu['Gender'])
insu.head()
```

	id	Gender	Age	Driving_License	Region_Code	Previously_Insured	Vehicle_Age	Vehicle_Damage	Annual_Premium	Policy_Sales_Channr
0	381110	Male	25	1	11	1	< 1 Year	No	35786	1
1	381111	Male	40	1	28	0	1-2 Year	Yes	33762	
2	381112	Male	47	1	28	0	1-2 Year	Yes	40050	1
3	381113	Male	24	1	27	1	< 1 Year	Yes	37356	1
4	381114	Male	27	1	28	1	< 1 Year	No	59097	1

```
le1=LabelEncoder()
insu['Veh_age']=le.fit_transform(insu['Vehicle_Age'])
insu.head()
```

	id	Gender	Age	Driving_License	Region_Code	Previously_Insured	Vehicle_Age	Vehicle_Damage	Annual_Premium	Policy_Sales_Channr
0	381110	Male	25	1	11	1	< 1 Year	No	35786	1
1	381111	Male	40	1	28	0	1-2 Year	Yes	33762	
2	381112	Male	47	1	28	0	1-2 Year	Yes	40050	1
3	381113	Male	24	1	27	1	< 1 Year	Yes	37356	1
4	381114	Male	27	1	28	1	< 1 Year	No	59097	1

```
log=LogisticRegression()
```

```
a=insu[['id','Driving_License','Veh_age','Annual_Premium']]
b=insu['Vehicle_Damage']
log.fit(a,b)
```

LogisticRegression

```
id=int(input("Enter your id:"))
Driving_License=int(input("Enter your Driving_License:"))
Veh_age=int(input("Enter your Veh_age:"))
Annual_Premium=int(input("Enter your Annual_Premium:"))
predict=log.predict([[id,Driving_License,Veh_age,Annual_Premium]])
print(predict)
```

```
Enter your id:12
Enter your Driving_License:5643
Enter your Veh_age:7
Enter your Annual_Premium:2009
['No']
/usr/local/lib/python3.11/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but Logistic
warnings.warn()
```

```
log.score(a,b)
```

```
0.6439383801569621
```

```
from sklearn.metrics import accuracy_score
preval=log.predict(a)
accuracy_score(b,preval)
```

```
0.6439383801569621
```

