

```
In [1]: #Loading Dataset
```

```
In [2]: import pandas as pd
```

```
In [3]: df = pd.read_csv('train.csv')
```

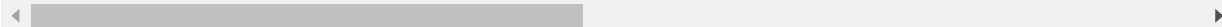
```
In [4]: #Showing dataset in short
```

```
In [5]: df.head()
```

Out[5]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utili
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	All
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	All
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	All
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	All
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	All

5 rows × 81 columns



```
In [6]: #Showing dataset in details (With all columns)
```

```
In [7]: pd.options.display.max_columns = None
df.head()
```

Out[7]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utili
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	All
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	All

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utili
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	All
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	All
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	All

```
In [8]: from sklearn.preprocessing import LabelEncoder
```

```
In [9]: #Encoding all columns which contains string value using below function
        applying LabelEncoder
```

```
In [10]: def encode(X, p):
          ldr = LabelEncoder()
          for q in p:
              encode = ldr.fit_transform(X[q])
              X[q] = encode
          return X
```

```
In [11]: X = df.drop(['Id', 'Alley', 'PoolQC', 'Fence', 'PoolArea', 'MiscFeatur
          e'], axis=1).dropna()
          y = X['SalePrice']
          X = X.drop(['SalePrice'], axis=1)
```

```
In [12]: #Encoding using LabelEncoder and the encode function mentioned above
```

```
In [13]: X = encode(X, ['MSZoning', 'Street',
          'LotShape', 'LandContour', 'Utilities', 'LotConfig', 'LandSlope',
          'Neighborhood', 'Condition1', 'Condition2', 'BldgType', 'HouseStyle',
          'RoofStyle',
          'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnrType',
          'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual', 'BsmtCond',
          'BsmtExposure', 'BsmtFinType1', 'BsmtFinType2', 'Heating', 'HeatingQC',
          'CentralAir', 'Electrical', 'KitchenQual',
          'Functional', 'FireplaceQu', 'GarageType',
          'GarageFinish', 'GarageQual', 'GarageCond',
```

```
'PavedDrive', 'SaleType',  
'SaleCondition']])
```

```
In [14]: #Splitting Dataset
```

```
In [15]: from sklearn.model_selection import train_test_split
```

```
In [16]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.  
1, random_state=42)
```

```
In [17]: # Fitting Linear Regression into the dataset
```

```
In [18]: from sklearn.linear_model import LinearRegression
```

```
In [19]: ln = LinearRegression()  
ln.fit(X_train, y_train)
```

```
Out[19]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,  
normalize=False)
```

```
In [20]: # Accuracy
```

```
In [21]: ln.score(X_test, y_test)
```

```
Out[21]: 0.8334292191988455
```

```
In [ ]:
```