

LLD (Low Level Design) :- Consumer Complaint Analysis

Import python libraries - these some libraires have to very important to execute for load the data and in others step to run the functions.

1. Pandas :- this software library used for data manipulation and analysis.
2. Numpy :- Numpy is used for working with array and numeric.
3. Matplotlib:- matplotlib is used for data visualization and graphical plotting.

_Above three types of libraries is import of python libraires.

Load the data - data is import part of the project what you would be prediction base on the data.

- Consumer Complaint Analysis data :- executed the data from CSV file.

Getting information - in initial, gain the information from data. Before it is also important what you want predict.

1.Shape method :- data shape is very huge in which 18 columns and 2785755 rows.

2.info method :- In this you have to know about Dtype (all datatype is object data) and as well as you can see columns name.

3.isna() method :- used this method then you can see how many NAN values present in particular columns data.

4. value_counts() method and unique() method :- this used two types of method you get values of the variable name in particular column.

Data Exploration and Visualize - in this you will have to explore the data whatever you have taken and visualize the that data that's why you can easily see and understand.

1. missing values data :- in this used round method and plot bar() for data visualize.

Now pick some columns (like a product wise complaints, state wise complaints to consumer disputed? Or not and Number of Complaints timely response or not in company) using some coding and plotting then you can see get idea by visualization.

Data Cleaning – this part is more carefully when you go to prepare the model. Remove an unwanted thing which is use for my prediction data.

1. `data = data[['Product','Consumer disputed?','Timely response?']]` -

I have picked some columns where come out from my prediction accuracy.

2. `data.dropna(inplace=True)` – used this method I have removed NAN values.

3. Label encoder method – used this method whenever string value convert into numeric.

4. `data.reset_index(inplace=True)`

`data.drop(['index'],axis=1,inplace=True)` – used this method that arranged in sequence order data.

5 loc method :- I used this method because this data is very huge rows that's why I want pick a certain row then I used this method.

Now your data is cleaned

Build a model – after cleaning then you would have to do build a model for training and testing.

1. `x = np.zeros(shape=(23706,48,48))`

`for i in range(len(data['Product'])):`

`x[i] = data['Product'][i]` --- I have executed x model for training data.

2. `y = pd.get_dummies(data['Consumer disputed?'])`

`y.shape` --- and this executed y model for testing data.

Split the model – in this you have to import model selection for training and testing split data.

1. `from sklearn.model_selection import train_test_split`

2. `x_train,x_test,y_train,y_test =
train_test_split(x,y,test_size=0.25,random_state=10)`

Import keras model & keras Layers – these libraries are within a deep learning libraries which is CNN algorithm part.

1. `from keras.models import Sequential`

2. `from keras.layers import
Dense,Activation,Conv2D,MaxPooling2D,Flatten,Dropout`

Build model base on CNN – when you use a CNN algorithm in deep learning so you would have to create a model.

In this first you would have to executed `model sequential()` and add some parameter like a in `conv2D` filter ,kernel size , padding , input shape, add activation “relu” , and `MaxPooling 2D()` all these are my first layer and same process create a layer as you want. As well as add flatten , Dense and activation. Last Activation is “softmax”. Like this create your CNN model.

Model compile – after build model then you have to do model compile.

1.

`model.compile(optimizer='adam',loss='categorical_crossentropy',metrics='accuracy')`

Model fit -then you have to do model fit process. These process have to important parts of CNN.

1.model.fit() method :- in this method you have to pick some important parameter like a epochs , validation data , x_train, y_train and batch size .

model.fit(x_train,y_train,batch_size=100,validation_data=(x_test,y_test),epochs=5) -- you can take any number of epochs as depend accuracy and val_accuracy and also batch size.

Evaluate model -in this you can see the history of Accuracy and loss.

You can see history of accuracy val_ accuracy and loss , val_accuracy by used plot method. plt.plot() method

Prediction – use the model evaluate method then give an accuracy.

model.evaluate(x_test,y_test)

loss : 0.4869 accuracy : 0.8104

so now **“Consumer Complaints not disputed”** accuracy prediction is **81 %**