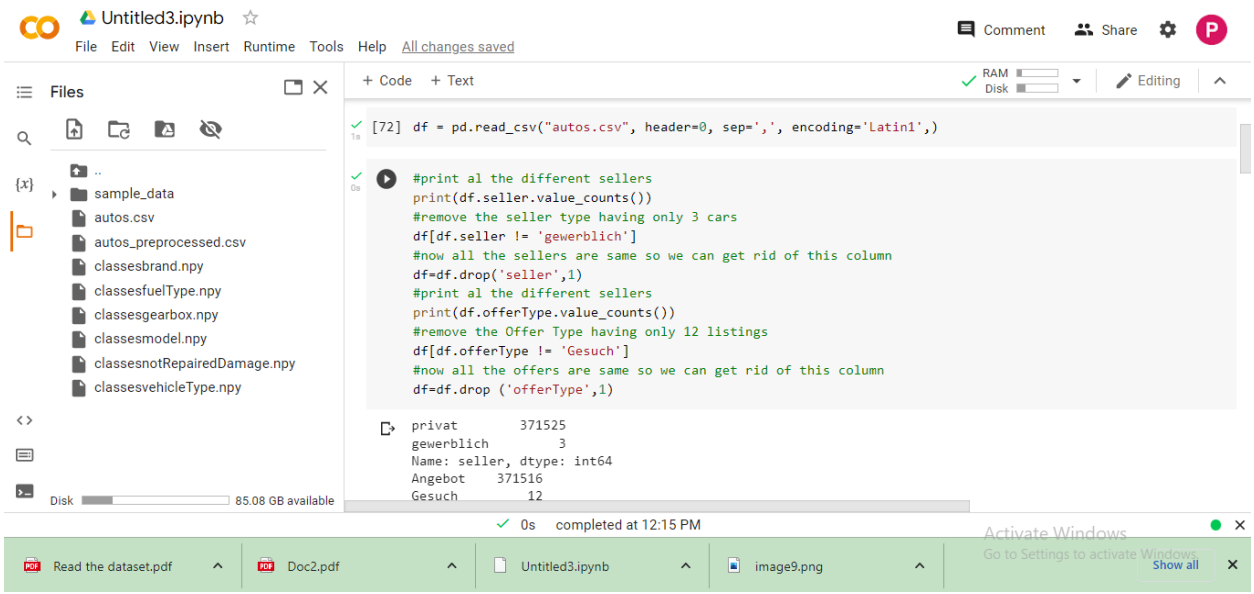
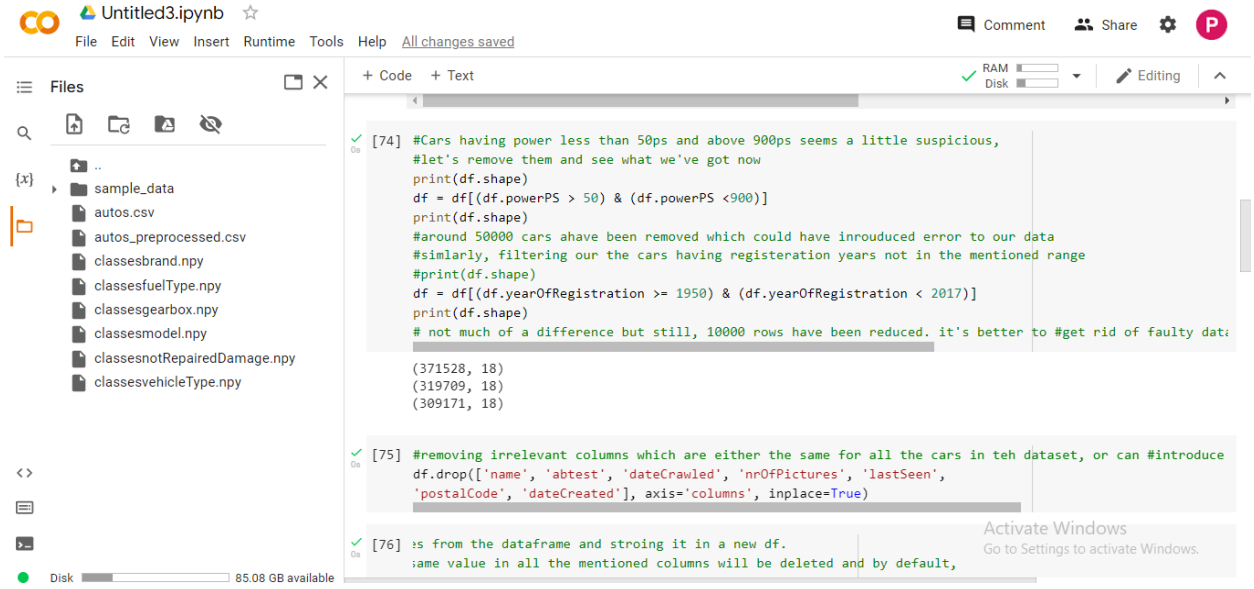
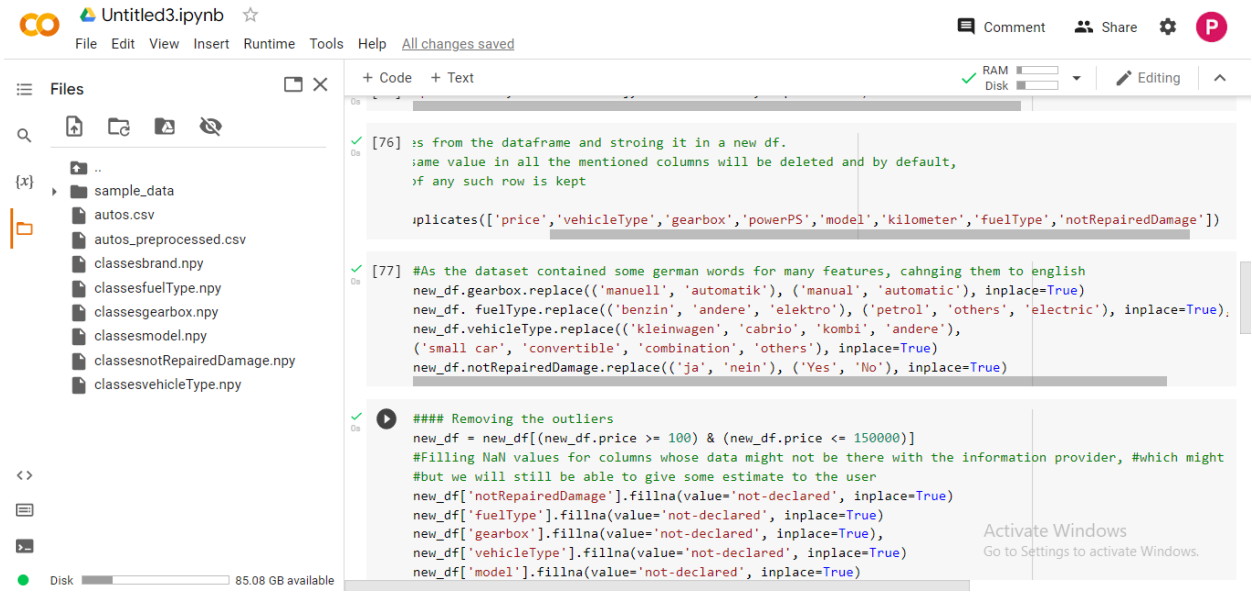
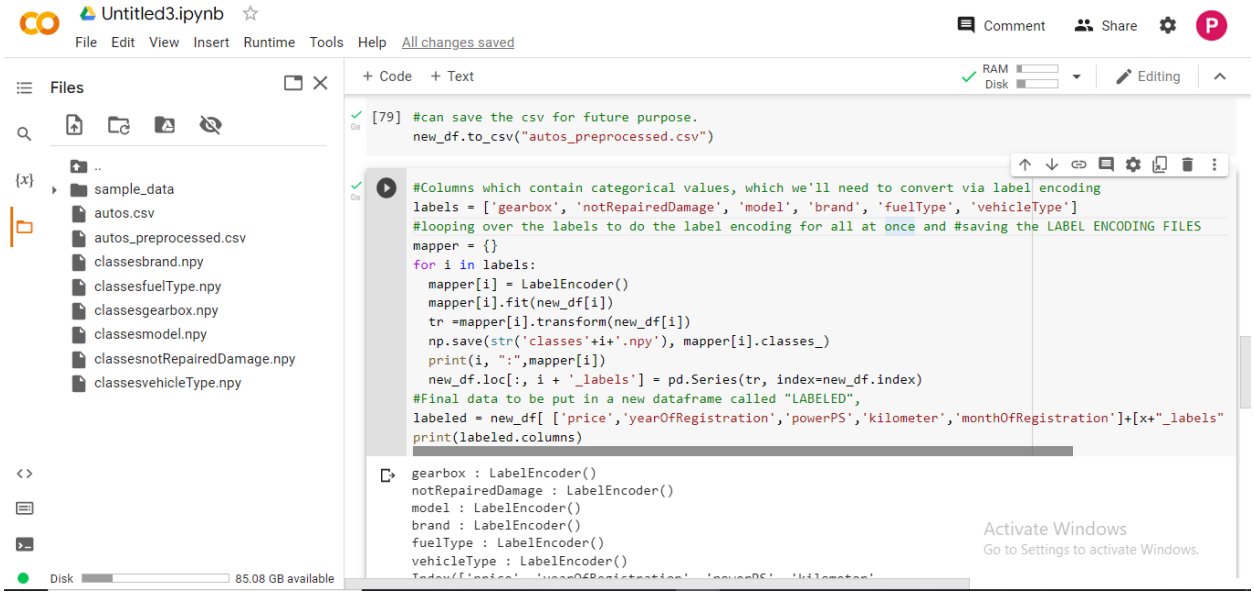
## CLEANING THE DATASET




Untitled3.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
Files
+ Code + Text
RAM Disk
Editing
sample_data
autos.csv
autos_preprocessed.csv
classesbrand.npy
classesfuelType.npy
classesgearbox.npy
classesmodel.npy
classesnotRepairedDamage.npy
classesvehicleType.npy
Disk 85.08 GB available
[76] s from the dataframe and stroing it in a new df.
ame value in all the mentioned columns will be deleted and by default,
f any such row is kept
plicates(['price','vehicleType','gearbox','powerPS','model','kilometer','fuelType','notRepairedDamage'])
[77] #As the dataset contained some german words for many features, cahnging them to english
new_df.gearbox.replace(('manuell', 'automatik'), ('manual', 'automatic'), inplace=True)
new_df. fuelType.replace(('benzin', 'andere', 'elektro'), ('petrol', 'others', 'electric'), inplace=True)
new_df.vehicleType.replace(('kleinwagen', 'cabrio', 'kombi', 'andere'),
('small car', 'convertible', 'combination', 'others'), inplace=True)
new_df.notRepairedDamage.replace(('ja', 'nein'), ('Yes', 'No'), inplace=True)
#### Removing the outliers
new_df = new_df[(new_df.price >= 100) & (new_df.price <= 150000)]
#Filling NaN values for columns whose data might not be there with the information provider, #which might
#but we will still be able to give some estimate to the user
new_df['notRepairedDamage'].fillna(value='not-declared', inplace=True)
new_df['fuelType'].fillna(value='not-declared', inplace=True)
new_df['gearbox'].fillna(value='not-declared', inplace=True),
new_df['vehicleType'].fillna(value='not-declared', inplace=True)
new_df['model'].fillna(value='not-declared', inplace=True)
Activate Windows
Go to Settings to activate Windows.


Untitled3.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
Files
+ Code + Text
RAM Disk
Editing
sample_data
autos.csv
autos_preprocessed.csv
classesbrand.npy
classesfuelType.npy
classesgearbox.npy
classesmodel.npy
classesnotRepairedDamage.npy
classesvehicleType.npy
Disk 85.08 GB available
[79] #can save the csv for future purpose.
new_df.to_csv("autos_preprocessed.csv")
#Columns which contain categorical values, which we'll need to convert via label encoding
labels = ['gearbox', 'notRepairedDamage', 'model', 'brand', 'fuelType', 'vehicleType']
#looping over the labels to do the label encoding for all at once and #saving the LABEL ENCODING FILES
mapper = {}
for i in labels:
mapper[i] = LabelEncoder()
mapper[i].fit(new_df[i])
tr =mapper[i].transform(new_df[i])
np.save(str('classes'+i+'.npy'), mapper[i].classes_)
print(i, ":",mapper[i])
new_df.loc[:, i + '_labels'] = pd.Series(tr, index=new_df.index)
#Final data to be put in a new dataframe called "LABELED",
labeled = new_df[ ['price','yearOfRegistration','powerPS','kilometer','monthOfRegistration']+[x+"_labels"
print(labeled.columns)
gearbox : LabelEncoder()
notRepairedDamage : LabelEncoder()
model : LabelEncoder()
brand : LabelEncoder()
fuelType : LabelEncoder()
vehicleType : LabelEncoder()
Activate Windows
Go to Settings to activate Windows.