```
#i want to train Bayesian neural network to predict the value of the
cars, so I have a lot of data which is categorial and numerical, I
want make model is up-to-date so after training I want to save the
model and use it in the future, so I want to know how to save the
model and load it again to use it in the future and I will give extra
new data before using it to predict the value of the car.

#download the data from the mongodb database
import pymongo
import pandas as pd
import numpy as np
myclient = pymongo.MongoClient("mongodb://localhost:27017/")
mydb = myclient["endterm"]
mycol = mydb["cars"]
mylist = list(mycol.find())
df = pd.DataFrame(mylist)
df = df.drop(columns=['_id'])
#drop all the rows which are duplicates
df = df.drop_duplicates()
df.head()
#print size of the data
```

```
           brand                model  year     price      city  \
0  Mercedes-Benz                E 230  1991   1500000    Алматы
1     Volkswagen               Passat  1992   1800000   Шымкент
2     ВАЗ (Lada)  Priora 2170 (седан)  2010   2800000  Костанай
3         Subaru             Forester  1998   3400000    Алматы
4        Hyundai             Grandeur  2021  23000000   Шымкент

                              generation  body_type engine_volume  \
0              1987 - 1993 W124 рестайлинг      седан  2.3 (бензин)
1                          1988 - 1993 B3  универсал  1.8 (бензин)
2                  2007 - 2015 1 поколение      седан  1.6 (бензин)
3             1997 - 2000 1 поколение (SF)  кроссовер    2 (бензин)
4  2019 - 2022 6 поколение рестайлинг (IG)      седан  3.3 (бензин)

  transmission            drive steering_wheel  custom_cleared
mileage  \
0      автомат    задний привод          слева            True
NaN
1     механика  передний привод          слева            True  245
555 км
2     механика  передний привод          слева            True  189
000 км
3      автомат    полный привод         справа            True
NaN
4      автомат  передний привод          слева            True    3
500 км

              color                VIN Наличие
```

```
0               NaN                NaN     NaN
1    серый металлик  11122233*44****66     NaN
2  зеленый металлик                NaN     NaN
3               NaN                NaN     NaN
4               NaN                NaN     NaN
```

```
#I want to bayesian neural network to predict the value of the cars,
so I have a lot of data which is categorial and numerical
#if mileage is nan I want to replace it with the 10000*(2023-year)
df['mileage'] = df['mileage'].fillna(10000*(2023-df['year']))
#if color is nan I want to replace it with the "не указан"
df['color'] = df['color'].fillna("не указан")
#if VIN is nan I want to replace it with the "не указан"
df['VIN'] = df['VIN'].fillna("не указан")
#if 'Наличие' is nan I want to replace it with the True, else False
df['Наличие'] = df['Наличие'].fillna(True)
df['Наличие'] = df['Наличие'].replace('На заказ', False)
df.head()
```

```
           brand                model  year     price      city  \
0  Mercedes-Benz                E 230  1991   1500000    Алматы
1     Volkswagen               Passat  1992   1800000   Шымкент
2     ВАЗ (Lada)  Priora 2170 (седан)  2010   2800000  Костанай
3         Subaru             Forester  1998   3400000    Алматы
4        Hyundai             Grandeur  2021  23000000   Шымкент

                              generation  body_type engine_volume  \
0              1987 - 1993 W124 рестайлинг      седан  2.3 (бензин)
1                          1988 - 1993 B3  универсал  1.8 (бензин)
2                 2007 - 2015 1 поколение      седан  1.6 (бензин)
3             1997 - 2000 1 поколение (SF)  кроссовер    2 (бензин)
4  2019 - 2022 6 поколение рестайлинг (IG)      седан  3.3 (бензин)

  transmission            drive steering_wheel  custom_cleared
mileage  \
0      автомат    задний привод          слева            True
320000
1     механика  передний привод          слева            True  245
555 км
2     механика  передний привод          слева            True  189
000 км
3      автомат    полный привод         справа            True
250000
4      автомат  передний привод          слева            True    3
500 км

              color                VIN  Наличие
0         не указан          не указан     True
1    серый металлик  11122233*44****66     True
2  зеленый металлик          не указан     True
```

```
3         не указан          не указан     True
4         не указан          не указан     True
```

```
#if engine_volume has "газ-бензин" make isPetrol=False, else
isPetrol=True
isPetrol = []
for i in range(len(df)):
    if df.engine_volume[i] == 'газ-бензин':
        isPetrol.append(False)
    else:
        isPetrol.append(True)
df['isPetrol'] = isPetrol
#delete all signs in engine_volume except numbers
import re
engine_volume = []
for i in range(len(df)):
    a = df.engine_volume[i]
    engine_volume.append(re.sub('[^\d\.]', '', a))

df['engine_volume'] = engine_volume

mileage = []
for i in range(len(df)):
    a = str(df.mileage[i])
    mileage.append(re.sub('[^\d\.]', '', a))

mileage= [int(i) for i in mileage]
df['mileage'] = mileage
df.head()
```

```
           brand                model  year     price      city  \
0  Mercedes-Benz                E 230  1991   1500000    Алматы
1     Volkswagen               Passat  1992   1800000   Шымкент
2     ВАЗ (Lada)  Priora 2170 (седан)  2010   2800000  Костанай
3         Subaru             Forester  1998   3400000    Алматы
4        Hyundai             Grandeur  2021  23000000   Шымкент

                                generation  body_type engine_volume  \
0             1987 - 1993 W124 рестайлинг      седан           2.3
1                          1988 - 1993 B3  универсал           1.8
2                 2007 - 2015 1 поколение      седан           1.6
3            1997 - 2000 1 поколение (SF)  кроссовер             2
4  2019 - 2022 6 поколение рестайлинг (IG)      седан           3.3

  transmission            drive steering_wheel  custom_cleared
mileage  \
0      автомат    задний привод          слева            True
320000
1     механика  передний привод          слева            True
245555
```

```
2     механика  передний привод          слева            True
189000
3      автомат    полный привод         справа            True
250000
4      автомат  передний привод          слева            True
3500

              color                VIN  Наличие  isPetrol
0         не указан          не указан     True      True
1    серый металлик  11122233*44****66     True      True
2  зеленый металлик          не указан     True      True
3         не указан          не указан     True      True
4         не указан          не указан     True      True
```

```
df=df.drop(columns=['VIN'])
df["engine_volume"] = df["engine_volume"].astype(float)
df.head()
cat_cols = ['brand', 'model', 'city', 'generation', 'body_type',
'transmission', 'drive',
            'steering_wheel', 'color']

#split the data to train and test
from sklearn.model_selection import train_test_split
X = df.drop(columns=['price'])
y = df['price']
#one hot encoding for categorical data
X = pd.get_dummies(X, columns=cat_cols)
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

#make random forest model and train it using tensorflow
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Make predictions on the test data
y_pred = model.predict(X_test)

# Evaluate the model using R-squared
r2 = r2_score(y_test, y_pred)
print('R-squared:', r2)
```

```
R-squared: 0.9023717104808213
```

```
#output the mean error of the model in the test data
from sklearn.metrics import mean_absolute_error
print(mean_absolute_error(y_test, y_pred))
```

```
6798510.664406779
```

```python
#find the biggest error in the test data and output it with the index
max_error = 0
index = 0
for i in range(len(y_test)):
    if abs(y_test.iloc[i]-y_pred[i]) > max_error:
        max_error = abs(y_test.iloc[i]-y_pred[i])
        index = i
print(max_error, index)
```

```
75051190.0 20
```

```python
print(y_test.iloc[20])
```

```
103000000
```

```python
print(y_pred[20])
```

```
157002666.6666667
```

```python
print(y_test.iloc[20]-y_pred[20])
```

```
-54002666.66666669
```

```python
print(X_test.iloc[20])
```

```
year                       2022
engine_volume               6.2
custom_cleared             True
mileage                   10000
Наличие                    True
                           ...
color_серый металлик          0
color_синий                   0
color_синий металлик          0
color_черный                  1
color_черный металлик         0
Name: 124, Length: 284, dtype: object
```

```python
#that's the biggest mistake model don't know anything about Cadillac
Escalade, but if we fit some info about it, it will be better

#example
#here we input df with  17 extra Cadillac Escalade(before we had
zero), after that we will fit this values and check the error
#make custom df
#in cars.csv we have json like objects, so we need to convert it to
the dictionary {'brand': 'Cadillac', 'model':

#upload all the documents from mongo, where brand is Cadillac and
model is Escalade
mylist = list(mycol.find({'brand': 'Cadillac', 'model': 'Escalade'}))
df = pd.DataFrame(mylist)
```

```
df = df.drop(columns=['_id'])
df.head()
```

```
      brand     model  year      price     city
generation  \
0  Cadillac  Escalade  2022  103000000   Алматы  2020 - н.в. 5
поколение
1  Cadillac  Escalade  2023   84000000  Шымкент  2020 - н.в. 5
поколение
2  Cadillac  Escalade  2023   89000000  Шымкент  2020 - н.в. 5
поколение
3  Cadillac  Escalade  2023   89000000  Шымкент  2020 - н.в. 5
поколение
4  Cadillac  Escalade  2021   81000000  Шымкент  2020 - н.в. 5
поколение

     body_type engine_volume transmission          drive
steering_wheel  \
0  внедорожник  6.2 (бензин)      автомат  полный привод
слева
1  внедорожник  6.2 (бензин)      автомат  полный привод
слева
2  внедорожник  6.2 (бензин)      автомат  полный привод
слева
3  внедорожник  6.2 (бензин)      автомат  полный привод
слева
4  внедорожник  6.2 (бензин)      автомат  полный привод
слева

   custom_cleared             color   Наличие    mileage  VIN
0            True            черный       NaN        NaN  NaN
1            True   черный металлик  На заказ        NaN  NaN
2            True  красный металлик  На заказ        NaN  NaN
3            True               NaN  На заказ        NaN  NaN
4            True   черный металлик       NaN  13 757 км  NaN
```

```
#if mileage is nan I want to replace it with the 10000*(2023-year)
df['mileage'] = df['mileage'].fillna(10000*(2023-df['year']))
#if color is nan I want to replace it with the "не указан"
df['color'] = df['color'].fillna("не указан")

#if engine_volume has "газ-бензин" make isPetrol=False, else
isPetrol=True
isPetrol = []
for i in range(len(df)):
    if df.engine_volume[i] == 'газ-бензин':
        isPetrol.append(False)
    else:
        isPetrol.append(True)
```

```
df['isPetrol'] = isPetrol
#delete all signs in engine_volume except numbers
import re
engine_volume = []
for i in range(len(df)):
    a = df.engine_volume[i]
    engine_volume.append(re.sub('[^\d\.]', '', a))

df['engine_volume'] = engine_volume
mileage = []
for i in range(len(df)):
    a = str(df.mileage[i])
    mileage.append(re.sub('[^\d\.]', '', a))

mileage= [int(i) for i in mileage]
df['mileage'] = mileage
df["engine_volume"] = df["engine_volume"].astype(float)
df['Наличие'] = df['Наличие'].fillna(True)
df['Наличие'] = df['Наличие'].replace('На заказ', False)
df.head()
```

```
      brand     model  year      price     city
generation  \
0  Cadillac  Escalade  2022  103000000   Алматы  2020 - н.в. 5
поколение
1  Cadillac  Escalade  2023   84000000  Шымкент  2020 - н.в. 5
поколение
2  Cadillac  Escalade  2023   89000000  Шымкент  2020 - н.в. 5
поколение
3  Cadillac  Escalade  2023   89000000  Шымкент  2020 - н.в. 5
поколение
4  Cadillac  Escalade  2021   81000000  Шымкент  2020 - н.в. 5
поколение

     body_type  engine_volume transmission          drive
steering_wheel  \
0  внедорожник            6.2      автомат  полный привод
слева
1  внедорожник            6.2      автомат  полный привод
слева
2  внедорожник            6.2      автомат  полный привод
слева
3  внедорожник            6.2      автомат  полный привод
слева
4  внедорожник            6.2      автомат  полный привод
слева

   custom_cleared             color  Наличие  mileage  VIN  isPetrol
0            True            черный     True    10000  NaN      True
1            True   черный металлик    False        0  NaN      True
```

```
2            True  красный металлик    False        0  NaN      True
3            True         не указан    False        0  NaN      True
4            True   черный металлик     True    13757  NaN      True
```

```
df=df.drop(columns=['VIN'])
```

```
X_new = df.drop(columns=['price'])
X_new = pd.get_dummies(X_new, columns=cat_cols)
#
y_new = df['price']
#resize the data to fit the model that columns are the same
X_new = X_new.reindex(columns = X_train.columns, fill_value=0)
model.fit(X_new, y_new)
```

```
RandomForestRegressor(random_state=42)
```

```
y_pred = model.predict(X_test)
```

```
print(y_pred[20])
```

```
96782500.0
```

```
print(y_test.iloc[20])
```

```
103000000
```

```
print(y_test.iloc[20]-y_pred[20])
```

```
6217500.0
```

```
#save the model to the file
import pickle
pickle.dump(model, open('model.pkl','wb'))
```

WE can that just by adding around 20 cars to the model we can reduce the error by a lot