```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
df = pd.read_csv('/content/drive/MyDrive/lab3/car_info.tsv',sep='\t')
```

```
import requests
import numpy as np
import csv
```

```
len(df)
```

```
5073
```

```
df.head(10)
```

| | mark | price | produced_year | imported_year | distance | motor_volume | color | type | hurd | hodolguur |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | Subaru Forester | 40.0 | 2017 | 2019 | 93000 км. | 2.5 л | Хар | Жийп | Буруу | Бензин |
| **1** | Toyota Prius 20 | 8.0 | 2004 | 2015 | 1111111 км. | 0.5 л | Бусад | Суудлын тэрэг | Буруу | Хайбрид |
| **2** | Toyota Alphard | 43.0 | 2013 | 2022 | 154279 км. | 2.4 л | Хар | Гэр бүлийн | Буруу | Бензин |
| **3** | Toyota Harrier | 22.0 | 2009 | 2013 | 150000 км. | 3.5 л | Шаргал | Жийп | Буруу | Бензин |
| **4** | Toyota Land Cruiser Prado | 101.0 | 2016 | 2024 | 54000 км. | 2.7 л | Саарал | Жийп | Буруу | Бензин |
| **5** | Nissan NV200 | 15.0 | 2013 | 2024 | 250 км. | 1.6 л | Саарал | Суудлын тэрэг | Буруу | Бензин |
| | Subaru XV | | | | | | | Гэр | | |

Next steps: Generate code with df  View recommended plots

```
df['distance'] = df['distance'].str.replace(' км.', '').astype(float)
```

```
<ipython-input-50-3c5e2011e05f>:1: FutureWarning: The default value of regex will change from True to False in a future version.
  df['distance'] = df['distance'].str.replace(' км.', '').astype(float)
```

```
df['motor_vol'] = df['motor_volume'].str.extract(r'(\d+\.\d+|\d+)').astype(float)
df.drop('motor_volume', axis=1, inplace=True)
```

```
df.head()
```

| | mark | price | produced_year | imported_year | distance | color | type | hurd | hodolguur | motor_vol |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Subaru Forester | 40.0 | 2017 | 2019 | 93000.0 | Хар | Жийп | Буруу | Бензин | 2.5 |
| 1 | Toyota Prius 20 | 8.0 | 2004 | 2015 | 1111111.0 | Бусад | Суудлын тэрэг | Буруу | Хайбрид | 0.5 |
| 2 | Toyota Alphard | 43.0 | 2013 | 2022 | 154279.0 | Хар | Гэр бүлийн | Буруу | Бензин | 2.4 |
| 3 | Toyota Harrier | 22.0 | 2009 | 2013 | 150000.0 | Шаргал | Жийп | Буруу | Бензин | 3.5 |
| 4 | Toyota Land | 101.0 | 2016 | 2024 | 54000.0 | Саарал | Жийп | Буруу | Бензин | 2.7 |

Next steps: Generate code with df View recommended plots

color баганыг хасав. Учир нь машины үнэд өнгө нөлөөлөхгүй гэж үзлээ.

```
df.drop('color', axis=1, inplace=True)
```

distance Баганы утгууд зарим нь худлаа бичигдсэн байна. Жишээ нь 2013 онд орж ирсэн машин 250 км явсан гэсэн байна. Энэ мэт датануудыг 1000 аар үржлээ.

```
def multiply_if_less_than_1000(value):
    if value < 1000:
        return value * 1000
    else:
        return value

df['distance'] = df['distance'].apply(multiply_if_less_than_1000)
```

```
df.head()
```

| | mark | price | produced_year | imported_year | distance | type | hurd | hodolguur | motor_vol |
|---|---|---|---|---|---|---|---|---|---|
| 0 | Subaru Forester | 40.0 | 2017 | 2019 | 93000.0 | Жийп | Буруу | Бензин | 2.5 |
| 1 | Toyota Prius 20 | 8.0 | 2004 | 2015 | 1111111.0 | Суудлын тэрэг | Буруу | Хайбрид | 0.5 |
| 2 | Toyota Alphard | 43.0 | 2013 | 2022 | 154279.0 | Гэр бүлийн | Буруу | Бензин | 2.4 |
| 3 | Toyota Harrier | 22.0 | 2009 | 2013 | 150000.0 | Жийп | Буруу | Бензин | 3.5 |
| 4 | Toyota Land Cruiser | 101.0 | 2016 | 2024 | 54000.0 | Жийп | Буруу | Бензин | 2.7 |

Next steps: Generate code with df View recommended plots

```
df['distance'] = df['distance'].replace(0, 50000)
```

```
avg_motor_vol = df['motor_vol'].mean()
avg_motor_vol
```

```
2.3868604419669524
```

```
df['motor_vol'] = df['motor_vol'].replace(0, avg_motor_vol)
```

```
df['distance_num'] = df['distance'].astype(float)
df.drop('distance', axis=1, inplace=True)
```

```
df.head()
```

| | mark | price | produced_year | imported_year | type | hurd | hodolguur | motor_vol | distance_num |
|---|---|---|---|---|---|---|---|---|---|
| **0** | Subaru Forester | 40.0 | 2017 | 2019 | Жийп | Бурyy | Бензин | 2.5 | 93000.0 |
| **1** | Toyota Prius 20 | 8.0 | 2004 | 2015 | Суудлын тэрэг | Бурyy | Хайбрид | 0.5 | 1111111.0 |
| **2** | Toyota Alphard | 43.0 | 2013 | 2022 | Гэр бүлийн | Бурyy | Бензин | 2.4 | 154279.0 |
| **3** | Toyota Harrier | 22.0 | 2009 | 2013 | Жийп | Бурyy | Бензин | 3.5 | 150000.0 |
| **4** | Toyota Land Cruiser | 101.0 | 2016 | 2024 | Жийп | Бурyy | Бензин | 2.7 | 54000.0 |

Next steps: Generate code with df  View recommended plots

```
def preprocess(app_set):
    ret = pd.DataFrame(columns = ["mark", "price", "produced_year", "imported_year", "type", "hurd", "hodolguur", "distance_num", "motor_vol"])
    for index, row in app_set.iterrows():
        if row['hurd'] == 'Бурyy':
          leas = False
        else:
          leas = True
        if row['distance_num'] > 1000000:
          road = row['distance_num'] / 10
        else:
          road = row['distance_num']

        ret = ret.append({'mark': row['mark'], 'price': row['price'], 'produced_year': row['produced_year'],
                          'imported_year': row['imported_year'], 'type': row['type'], 'hurd': leas,
                          'hodolguur': row['hodolguur'], 'distance_num': road, 'motor_vol' : row['motor_vol']}, ignore_index = True)
    return ret
```

```
finally_data = preprocess(df)
```

```
finally_data.head()
```

| | mark | price | produced_year | imported_year | type | hurd | hodolguur | distance_num | motor_vol |
|---|---|---|---|---|---|---|---|---|---|
| 0 | Subaru Forester | 40.0 | 2017 | 2019 | Жийп | False | Бензин | 93000.0 | 2.5 |
| 1 | Toyota Prius 20 | 8.0 | 2004 | 2015 | Суудлын тэрэг | False | Хайбрид | 111111.1 | 0.5 |
| 2 | Toyota Alphard | 43.0 | 2013 | 2022 | Гэр бүлийн | False | Бензин | 154279.0 | 2.4 |
| 3 | Toyota Harrier | 22.0 | 2009 | 2013 | Жийп | False | Бензин | 150000.0 | 3.5 |
| 4 | Toyota Land Cruiser | 101.0 | 2016 | 2024 | Жийп | False | Бензин | 54000.0 | 2.7 |

Next steps: Generate code with finally_data View recommended plots

```
finally_data.drop('type', axis=1, inplace=True)
finally_data.drop('hurd', axis=1, inplace=True)
```

```
finally_data.head()
```

| | mark | price | produced_year | imported_year | hodolguur | distance_num | motor_vol |
|---|---|---|---|---|---|---|---|
| 0 | Subaru Forester | 40.0 | 2017 | 2019 | Бензин | 93000.0 | 2.5 |
| 1 | Toyota Prius 20 | 8.0 | 2004 | 2015 | Хайбрид | 111111.1 | 0.5 |
| 2 | Toyota Alphard | 43.0 | 2013 | 2022 | Бензин | 154279.0 | 2.4 |
| 3 | Toyota Harrier | 22.0 | 2009 | 2013 | Бензин | 150000.0 | 3.5 |
| 4 | Toyota Land Cruiser Prado | 101.0 | 2016 | 2024 | Бензин | 54000.0 | 2.7 |

Next steps: Generate code with finally_data View recommended plots

```
!pip install -U sentence-transformers
```

```
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.15.1->sentence-transformers) (2.31.0)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.15.1->sentence-transformers) (6.0.1)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.15.1->sentence-transformers) (4.10.0)
Requirement already satisfied: packaging>=20.9 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.15.1->sentence-transformers) (24.0)
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->sentence-transformers) (1.12)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->sentence-transformers) (3.2.1)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->sentence-transformers) (3.1.3)
Collecting nvidia-cuda-nvrtc-cu12==12.1.105 (from torch>=1.11.0->sentence-transformers)
  Downloading nvidia_cuda_nvrtc_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (23.7 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 23.7/23.7 MB 27.1 MB/s eta 0:00:00
Collecting nvidia-cuda-runtime-cu12==12.1.105 (from torch>=1.11.0->sentence-transformers)
```

```
Collecting nvidia-curand-cu12==10.3.2.106 (from torch>=1.11.0->sentence-transformers)
  Downloading nvidia_curand_cu12-10.3.2.106-py3-none-manylinux1_x86_64.whl (56.5 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 56.5/56.5 MB 8.2 MB/s eta 0:00:00
Collecting nvidia-cusolver-cu12==11.4.5.107 (from torch>=1.11.0->sentence-transformers)
  Downloading nvidia_cusolver_cu12-11.4.5.107-py3-none-manylinux1_x86_64.whl (124.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 124.2/124.2 MB 8.3 MB/s eta 0:00:00
Collecting nvidia-cusparse-cu12==12.1.0.106 (from torch>=1.11.0->sentence-transformers)
  Downloading nvidia_cusparse_cu12-12.1.0.106-py3-none-manylinux1_x86_64.whl (196.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 196.0/196.0 MB 2.4 MB/s eta 0:00:00
Collecting nvidia-nccl-cu12==2.19.3 (from torch>=1.11.0->sentence-transformers)
  Downloading nvidia_nccl_cu12-2.19.3-py3-none-manylinux1_x86_64.whl (166.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 166.0/166.0 MB 2.2 MB/s eta 0:00:00
Collecting nvidia-nvtx-cu12==12.1.105 (from torch>=1.11.0->sentence-transformers)
  Downloading nvidia_nvtx_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (99 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 99.1/99.1 kB 11.5 MB/s eta 0:00:00
Requirement already satisfied: triton==2.2.0 in /usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->sentence-transformers) (2.2.0)
Collecting nvidia-nvjitlink-cu12 (from nvidia-cusolver-cu12==11.4.5.107->torch>=1.11.0->sentence-transformers)
  Downloading nvidia_nvjitlink_cu12-12.4.99-py3-none-manylinux2014_x86_64.whl (21.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 21.1/21.1 MB 57.3 MB/s eta 0:00:00
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10/dist-packages (from transformers<5.0.0,>=4.32.0->sentence-transformers) (2023.12.25)
Requirement already satisfied: tokenizers<0.19,>=0.14 in /usr/local/lib/python3.10/dist-packages (from transformers<5.0.0,>=4.32.0->sentence-transformers) (0.15.2)
Requirement already satisfied: safetensors>=0.4.1 in /usr/local/lib/python3.10/dist-packages (from transformers<5.0.0,>=4.32.0->sentence-transformers) (0.4.2)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn->sentence-transformers) (1.3.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn->sentence-transformers) (3.4.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch>=1.11.0->sentence-transformers) (2.1.5)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hub>=0.15.1->sentence-transformers) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hub>=0.15.1->sentence-transformers) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hub>=0.15.1->sentence-transformers) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hub>=0.15.1->sentence-transformers) (2024.2.2)
Requirement already satisfied: mpmath>=0.19 in /usr/local/lib/python3.10/dist-packages (from sympy->torch>=1.11.0->sentence-transformers) (1.3.0)
Installing collected packages: nvidia-nvtx-cu12, nvidia-nvjitlink-cu12, nvidia-nccl-cu12, nvidia-curand-cu12, nvidia-cufft-cu12, nvidia-cuda-runtime-cu12, nvidia-cuda-nvrtc-cu12,
Successfully installed nvidia-cublas-cu12-12.1.3.1 nvidia-cuda-cupti-cu12-12.1.105 nvidia-cuda-nvrtc-cu12-12.1.105 nvidia-cuda-runtime-cu12-12.1.105 nvidia-cudnn-cu12-8.9.2.26 nvi
```

```
from huggingface_hub import notebook_login
notebook_login()
```

Token is valid (permission: write).

Your token has been saved in your configured git credential helpers (store).

Your token has been saved to /root/.cache/huggingface/token

Login successful

```
from sentence_transformers import SentenceTransformer
sen_model = SentenceTransformer('joeddav/xlm-roberta-large-xnli')
```

```
WARNING:sentence_transformers.SentenceTransformer:No sentence-transformers model found with name joeddav/xlm-roberta-l
config.json: 100%  734/734 [00:00<00:00, 30.9kB/s]
pytorch_model.bin: 100%  2.24G/2.24G [00:52<00:00, 43.4MB/s]
tokenizer_config.json: 100%  25.0/25.0 [00:00<00:00, 742B/s]
sentencepiece.bpe.model: 100%  5.07M/5.07M [00:00<00:00, 17.1MB/s]
special_tokens_map.json: 100%  150/150 [00:00<00:00, 5.24kB/s]
```

```
sentences = []
y = []
for index, row in df.iterrows():
    sentences.append(row['mark'] + ', ' + str(row['produced_year']) + ' онд үйлдвэрлэсэн, ' + str(row['imported_year']) + ' онд орж ирсэн, ' + str(row['distance_num']) + ' км явсан, ' +
    y.append(row['price'])
```

```
sentences[:10]
```

```
['Subaru Forester, 2017 онд үйлдвэрлэсэн, 2019 онд орж ирсэн, 93000.0 км явсан, 2.5 л мотор',
 'Toyota Prius 20, 2004 онд үйлдвэрлэсэн, 2015 онд орж ирсэн, 1111111.0 км явсан, 0.5 л мотор',
 'Toyota Alphard, 2013 онд үйлдвэрлэсэн, 2022 онд орж ирсэн, 154279.0 км явсан, 2.4 л мотор',
 'Toyota Harrier, 2009 онд үйлдвэрлэсэн, 2013 онд орж ирсэн, 150000.0 км явсан, 3.5 л мотор',
 'Toyota Land Cruiser Prado, 2016 онд үйлдвэрлэсэн, 2024 онд орж ирсэн, 54000.0 км явсан, 2.7 л мотор',
 'Nissan NV200, 2013 онд үйлдвэрлэсэн, 2024 онд орж ирсэн, 250000.0 км явсан, 1.6 л мотор',
 'Subaru XV Crosstrek, 2018 онд үйлдвэрлэсэн, 2024 онд орж ирсэн, 150000.0 км явсан, 2.0 л мотор',
 'Lexus IS, 2014 онд үйлдвэрлэсэн, 2024 онд орж ирсэн, 160000.0 км явсан, 2.5 л мотор',
 'Toyota Prius 20, 2007 онд үйлдвэрлэсэн, 2023 онд орж ирсэн, 74000.0 км явсан, 1.5 л мотор',
 'Lexus IS, 2014 онд үйлдвэрлэсэн, 2020 онд орж ирсэн, 147000.0 км явсан, 2.0 л мотор']
```

```
sentence_vectors = sen_model.encode(sentences)
```

```
from sklearn.linear_model import LinearRegression
model = LinearRegression(fit_intercept=True)
```

```
x_ = sentence_vectors
```

```
model.fit(x_, y)
```

```
▾ LinearRegression
LinearRegression()
```

```
import re
def extract_car_info(sentence):
    patterns = {
        'mark': r'(.*?)\,',
        'produced_year': r'(\d+)\s+онд\s+үйлдвэрлэсэн',
        'imported_year': r'(\d+)\s+онд\s+орж\s+ирсэн',
        'distance_num': r'(\d+)\s+км\s+явсан',
        'motor_vol': r'(\d+(\.\d+)?)\s+л\s+мотор'
    }

    car_info = {key: [] for key in patterns}

    for key, pattern in patterns.items():
        match = re.search(pattern, sentence)
        if match:
            car_info[key].append(match.group(1))
        else:
            car_info[key].append(None)

    return car_info

sentence = "Toyota Land Cruiser 300, 2019 онд үйлдвэрлэсэн, 2021 онд орж ирсэн, 54000 км явсан, 3.0 л мотор"
```

```
car_info = extract_car_info(sentence)
print(car_info)
```

```
{'mark': ['Toyota Land Cruiser 300'], 'produced_year': ['2019'], 'imported_year': ['2021'], 'distance_num': ['54000'], 'motor_vol': ['3.0']}
```

```
def predictEstimate(sentence):
  p_embeddings = sen_model.encode([sentence])
  d = [extract_car_info(sentence)]
  p_ = []
  for emb in p_embeddings:
    #B = [d[0]]
    p_.append(emb)
  y_prediction =  model.predict(p_)
  return y_prediction[0]
```

```
print(predictEstimate("Lexus IS, 2014 онд үйлдвэрлэсэн, 2024 онд орж ирсэн, 160000.0 км явсан, 2.5 л мотор"))
```

```
49.457886
```

```
print(predictEstimate("Toyota Prius 20, 2019 онд үйлдвэрлэсэн, 2020 онд орж ирсэн, 100000.0 км явсан, 1.8 л мотор"))
```

```
26.253418
```

```
print(predictEstimate("Toyota Land Cruiser 100, 2021 онд үйлдвэрлэсэн, 2023 онд орж ирсэн, 80000 км явсан, 4.5 л мотор"))
```

```
147.24182
```

```
print(predictEstimate("Nissan GT-R, 2017 онд үйлдвэрлэсэн, 2017 онд орж ирсэн, 2000.0 км явсан, 3.8 л мотор"))
```

```
391.94208
```