

---

# База знаний автомобильных моделей

Нагавкина София

---

# Проблема

Выбор машины это:

- долго
- сложно
- непонятно

(особенно если не разбираешься в этом)

---

# Решение

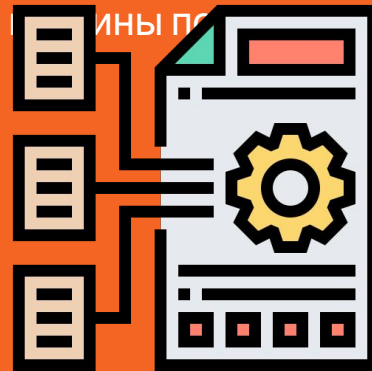
Построить рекомендательную систему на  
базе собранных данных



# Как

## Объединить несколько ресурсов:

1. [caradvice.com.au](http://caradvice.com.au) - характеристики автомобилей
2. [auto.ru](http://auto.ru) - доступность для покупки
3. [euroncar](http://euroncar) - безопасность
4. [automobiledimension.com/similar-autos.php](http://automobiledimension.com/similar-autos.php) - похожие машины по



# Для кого

1. Будущие покупатели
2. Колл-центр автоподбора
3. Data scientists

---

# Car tags

1. Cheap
2. first car
3. car as art (unicorns)
4. for woman
5. for man
6. family
7. outdoor
8. lux
9. city
10. country
- 11.

---

---

# Ресурсы

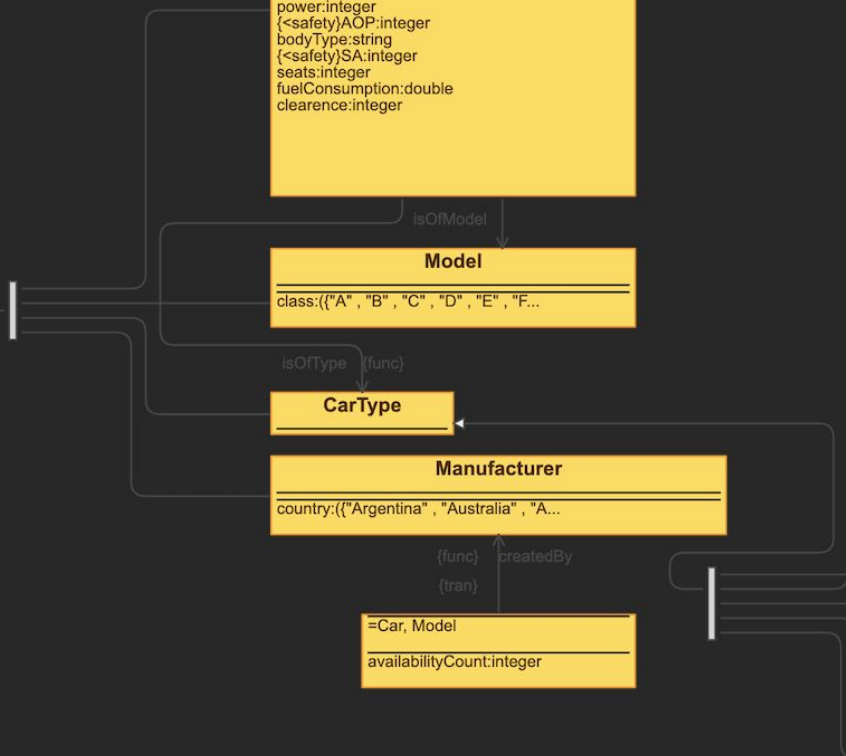
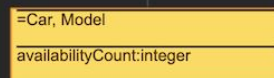
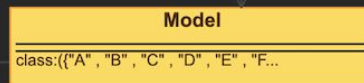
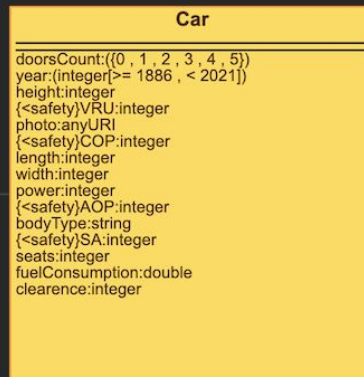
ресурс	данные	способ доступа
auto.ru	доступные модели и их характеристики	запросы к API
ancap.com.au/safety-ratings	оценка безопасности	запросы к API
carwow.co.uk	типы автомобилей (small/family/economy)	запросы к API

---

# Компетентностные вопросы

1. Найти популярную альтернативу конкретной машине по габаритам
  2. Какие редкие машины доступны на рынке?
  3. Какой семейный автомобиль самый распространенный?
  4. Какие есть редкие машины с низким расходом топлива?
  5. Сколько редких машин произведено в конкретной стране?
-





# SPARQL(1)

*# Вопрос: Найти популярную альтернативу конкретной машине по габаритам*

```
gres = graph.query("""SELECT ?m2 ?man
WHERE {
  values ?car { <http://www.semanticweb.org/sofia/ontologies/2021/3/cars#FIESTA_HATCHBACK_5_DOORS
    ?car      :clearance      ?c .
    ?s :clearance      ?c2 .
    ?car      :width          ?w .
    ?car      :length         ?l .
    ?car      :height         ?h .
    ?car      :isOfModel      ?m .
    ?s :clearance      ?c2 .
    ?s :width          ?w2 .
    ?s :length         ?l2 .
    ?s :height         ?h2 .
    ?s      :isOfModel      ?m2 .
    ?m2 :createdBy ?man
  FILTER(?c2 < ?c + 5) .
  FILTER(?c2 > ?c - 5) .
  FILTER(?h2 < ?h + 50) .
  FILTER(?h2 > ?h - 50) .
  FILTER(?l2 < ?l + 200) .
  FILTER(?l2 > ?l - 200) .
  FILTER(?w2 < ?w + 50) .
  FILTER(?w2 > ?w - 50) .
  FILTER(?m != ?m2) .
}
GROUP BY ?m2
""")
```

CITROEN C4  
HONDA CIVIC  
HYUNDAI I30  
KIA CEED  
CHEVROLET CORSA  
TOYOTA COROLLA  
VOLKSWAGEN GOLF  
HYUNDAI I20  
CHEVROLET AVEO  
TOYOTA AURIS  
PEUGEOT 308  
DACIA LOGAN  
FIAT STILO

# SPARQL(2)

*# Вопрос: Какие редкие машины доступны на рынке?*

```
qres = graph.query("""SELECT ?m ?man (sum(?count) AS ?Total)
```

```
SUZUKI MR_WAGON 3
OPEL COMMODORE 2
OPEL P4 2
AUDI Q8 4
PEUGEOT BIPPER 1
DODGE DAYTONA 4
FIAT SEICENTO 3
NISSAN 200SX 1
DAIHATSU MOVE_CONTE 3
ALFA_ROMEO 146 4
HONDA CROSSTOUR 3
HYUNDAI I10 3
DAEWOO DAMAS 3
CHRYSLER LHS 4
FORD BRONCO 4
CHEVROLET BEL_AIR 2
NISSAN XTERRA 4
HONDA EDIX 4
SUZUKI FORENZA 4
```

```
WHERE {
    ?car          :isOfModel      ?m .
    ?car          :availabilityCount ?count .
    ?m :createdBy ?man
    FILTER(?count < 5) .
}
GROUP BY ?m
HAVING (sum(?count) < 5)
""")
```

# SPARQL(3)

*# Вопрос: Какие есть редкие машины с низким расходом топлива?*

```
qres = graph.query( """SELECT ?m ?man ?fuel
```

```
WHERE {
```

```
  ?car          :isOfModel      ?m .
```

```
  ?car          :availabilityCount  ?count .
```

```
  ?car          :fuelConsumption    ?fuel .
```

```
  ?m :createdBy ?man
```

```
  FILTER(?count < 5) .
```

```
  FILTER(?fuel < 5) .
```

```
}
```

```
GROUP BY ?m
```

```
HAVING (sum(?count) < 5)
```

```
""")
```

```
TOYOTA AQUA 2
```

```
CITROEN C3 4
```

```
VOLVO S80 4
```

```
ALFA_ROMEO MITO 4
```

```
TOYOTA AYGO 4
```

```
CITROEN C4_CACTUS 4
```

```
VOLVO V70 4
```

```
CITROEN C4_SPACETOUSER 3
```

```
AUDI A3 4
```

```
AUDI A1 4
```

```
VOLVO V90 4
```

# SPARQL(4)

*# Вопрос: Сколько редких машин произведено в конкретной стране?*

```
gres = graph.query("""SELECT ?m ?man (sum(?count) AS ?Total)
                    WHERE {
                        ?car          :isOfModel      ?m .
                        ?car          :availabilityCount ?count .
                        ?car          :fuelConsumption ?fuel .
                        ?m :createdBy ?man.
                        ?man :country "Germany".
                        FILTER(?count < 5) .
                    }
                    GROUP BY ?m
                    HAVING (sum(?count) < 5)
                    """)
```

AUDI	E_TRON	4
AUDI	Q8	4
AUDI	S5	1
OPEL	GT	3

# VOID

```
# generate VOID
from rdflib.void import generateVoID
q = generateVoID(graph)
print(q)
q[0].serialize(destination='cars_with_void.rdf')
```

Individuals: Dataset\_class2



- ◆ Dataset
- ◆ Dataset\_class0
- ◆ Dataset\_class1
- ◆ Dataset\_class10
- ◆ Dataset\_class11
- ◆ Dataset\_class12
- ◆ Dataset\_class13
- ◆ Dataset\_class14
- ◆ Dataset\_class2
- ◆ Dataset\_class3
- ◆ Dataset\_class4
- ◆ Dataset\_class5
- ◆ Dataset\_class6
- ◆ Dataset\_class7
- ◆ Dataset\_class8
- ◆ Dataset\_class9
- ◆ Dataset\_property0



Annotations Usage

Annotations: Dataset\_class2

Annotations +

void:class

<http://www.semanticweb.org/sofia/ontologies/2021/3/cars#Model>

void:classes [type: xsd:integer]

1

void:distinctObjects [type: xsd:integer]

1702

void:distinctSubjects [type: xsd:integer]

1288

void:entities [type: xsd:integer]

1288

void:properties [type: xsd:integer]

Description: Dataset\_class2



# R2RML (mapping)

(using <https://github.com/nkons/r2rml-parser>)

```
map:model
  rr:logicalTable [ rr:tableName "models"; ];
  rr:subjectMap [
    rr:template 'http://www.semanticweb.org/sofia/ontologies/r2rml/cars#{"name"}';
    rr:class pref:Model;
  ];
.
map:car
  rr:logicalTable [ rr:tableName "car"; ];
  rr:subjectMap [
    rr:template 'http://www.semanticweb.org/sofia/ontologies/r2rml/cars#{"model"}_{"bodytype"}_{"year"}';
    rr:class pref:Car;
  ];
  rr:predicateObjectMap [
    rr:predicate pref:isOfModel;
    rr:objectMap [ rr:template 'http://www.semanticweb.org/sofia/ontologies/r2rml/cars#{"model"}' ];
  ];
  rr:predicateObjectMap [
    rr:predicate pref:year;
    rr:objectMap [rr:column "year"];
  ];
  rr:predicateObjectMap [
    rr:predicate pref:clearance;
    rr:objectMap [rr:column "clearance"];
  ];
  rr:predicateObjectMap [
    rr:predicate pref:carClass;
    rr:objectMap [rr:column "carClass"];
  ];
  rr:predicateObjectMap [
    rr:predicate pref:height;
    rr:objectMap [rr:column "height"];
  ];
  rr:predicateObjectMap [
    rr:predicate pref:length;
    rr:objectMap [rr:column "length"];
  ];
  ;
```

.rdf mapping file

# R2RML (result)

(using <https://github.com/nkons/r2rml-parser>)

The screenshot displays the R2RML editor interface, which is divided into several panels:

- Class hierarchy: Car**: Shows a tree structure starting from `owl:Thing`, with `Car` selected. Other classes in the hierarchy include `CarType`, `Manufacturer`, and `Model`.
- Direct instances: A8\_SEDAN\_2018**: A list of instances for the `Car` class. The instance `A8_SEDAN_2018` is highlighted. Other instances include `508_COUPE_1939`, `920_SEDAN_1938`, `A8_SEDAN_2017`, `A8_SEDAN_2019`, `A8_SEDAN_2020`, `A8_SEDAN_2021`, `ADMIRAL_SEDAN_1939`, `ASTRO_MINIVAN_1989`, `ASTRO_MINIVAN_1994`, `ASTRO_MINIVAN_2001`, `BB_PICKUP_2001`, `BEL_AIR_SEDAN_1955`, `BLUEBIRD_MAXIMA_SEDAN_1987`, `BLUEBIRD_MAXIMA_SEDAN_1988`, `C_10_PICKUP_1960`, and `CADDY_PICKUP_1988`.
- Annotations: A8\_SEDAN\_2018**: Shows the annotations for the selected instance. The annotation `carClass` is set to `-1`. The `isOfModel` property is set to `A8`.
- Description: A8\_SEDAN\_2018**: Shows the description of the selected instance. The type is `Car`.
- Property assertions: A8\_SEDAN\_2018**: Shows the property assertions for the selected instance. The assertions are: `width 1945`, `length 5302`, `photo "http://avatars.mds.yandex.net/get-"`, `fuelConsumption "-1"`, `bodyType "SEDAN"`, `cost 6000000`, `height 1485`, `availabilityCount "8"`, `doorsCount "4"`, `power 249`, `clearance "140"`, and `year 2018`.



# SHACL

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix ex: <http://www.example.org/#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix pref: <http://www.semanticweb.org/sofia/ontologies/2021/3/cars#> .
```

```
pref:Car
  a sh:NodeShape ;
  sh:targetClass pref:Car ; # Applies to all persons
  sh:property [
    sh:path pref:year ;      # constrains the values of ex:ssn
    sh:maxCount 1 ;
    sh:minInclusive 1800 ;
    sh:maxInclusive 2021 ;
  ] ;
  sh:property [
    sh:path pref:availabilityCount ;
    sh:maxCount 1;
    sh:minInclusive 0;
    sh:datatype xsd:integer;
  ] ;
  sh:property [
    sh:path pref:clearance ;
    sh:maxCount 1;
    sh:minInclusive 0;
    sh:maxInclusive 300;
    sh:datatype xsd:integer;
  ] ;
  sh:property [
    sh:path pref:bodyType ;
    sh:maxCount 1;
    sh:datatype xsd:string;
  ] ;
  sh:property [
    sh:path pref:color ;
  ] ;
```

SHACL constraint violations: 0/255

Severity	SourceShape	Message

# Documentation(widoco)

ontology	<http://www.semanticweb.org/ground/ontology/2021/3>
3	<http://www.semanticweb.org/ground/ontology/2021/3>
owl	<http://www.w3.org/2002/07/owl>
rdf	<http://www.w3.org/1999/02/22-rdf-syntax-ns>
xsd	<http://www.w3.org/2001/XMLSchema>
rdfs	<http://www.w3.org/2000/01/rdf-schema>

## 2. Car models: Overview

[back to ToC](#)

This ontology has the following classes and properties.

### Classes

[budget car](#) [car](#) [family car](#) [lux car](#) [rare car](#) [small car](#)

### Object Properties

[created by](#) [is of model](#) [is of type](#) [produces model](#)

### Data Properties

[a o p](#) [availability count](#) [body type](#) [c o p](#) [class](#) [clearance](#) [cost](#) [country](#) [doors count](#) [fuel consumption](#) [height](#) [length](#) [photo](#) [power](#) [s a](#) [safety](#)  
[seats](#) [v r u](#) [width](#) [year](#)

### Named Individuals

# Возникшие проблемы

- Protege фризилось при попытке работы на macOS (пришлось поднимать виртуальное окружение)
- Парсинг данных с сайта занял очень много времени (+ возникали ошибки в связи с наличием пробелов в данных)
- Мало справочных материалов по использованным утилитам (кроме документации, которая не идеальна)
- При конвертации R2RML возникла проблема с пробельными символами
- При работе с rdflib выкидываемые ошибки не достаточно информативны, что замедляет поиск ошибки