

Лабораторная работа

Деревья решений

Задания

1. Создайте набор данных. Выполните его визуализацию
2. Приведите данные к бинарному виду. Постройте дерево решений
3. Выполните обучение и тестирование. Опишите модель – исходные данные, выходные данные и результаты обучения.
4. Постройте дерево решений для регрессии

The screenshot shows a Google Colab notebook titled "DecTree.ipynb". The code cell contains the following Python code:

```
[ ] #лабораторная деревья решений

[1] from sklearn import tree
import pandas as pd
import pydotplus
from IPython.display import Image

golf_df = pd.DataFrame()
golf_df['Outlook'] = ['sunny', 'sunny', 'overcast', 'rainy', 'rainy', 'rainy',
                    'overcast', 'sunny', 'sunny', 'rainy', 'sunny', 'overcast',
                    'overcast', 'rainy']

golf_df['Temperature'] = ['hot', 'hot', 'hot', 'mild', 'cool', 'cool', 'cool',
                        'mild', 'cool', 'mild', 'mild', 'mild', 'hot', 'mild']

golf_df['Humidity'] = ['high', 'high', 'high', 'high', 'normal', 'normal', 'normal',
                    'high', 'normal', 'normal', 'normal', 'high', 'normal', 'high']

golf_df['Windy'] = ['false', 'true', 'false', 'false', 'false', 'true', 'true',
                  'false', 'false', 'false', 'true', 'true', 'false', 'true']

golf_df['Play'] = ['no', 'no', 'yes', 'yes', 'yes', 'no', 'yes', 'no', 'yes', 'yes', 'yes',
                 'yes', 'yes', 'no']

print(golf_df)
```

The output of the code is a DataFrame with 14 rows and 6 columns:

	Outlook	Temperature	Humidity	Windy	Play
0	sunny	hot	high	false	no
1	sunny	hot	high	true	no
2	overcast	hot	high	false	yes
3	rainy	mild	high	false	yes
4	rainy	cool	normal	false	yes
...
9	rainy	mild	normal	false	yes
10	sunny	mild	normal	true	yes
11	overcast	mild	high	true	yes
12	overcast	hot	normal	false	yes

TREE - Google | DecTree.ipynb | Copy of DecTree | Regression.ipynb | Linear Regression | Linear Regression | Python Decision | +

colab.research.google.com/drive/1g0tuLeR0AvYVVIPIjNx8jrDi35ZGqX54?complete=0&safe=active#scrollTo=B871EbNywJ-b

DecTree.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

RAM Disk Editing

```
[2] 9 sunny mild normal false yes
    10 sunny mild normal true yes
    11 overcast mild high true yes
    12 overcast hot normal false yes
    13 rainy mild high true no
```

```
[ ]
```

```
one_hot_data = pd.get_dummies(golf_df[ ['Outlook', 'Temperature', 'Humidity', 'Windy' ]])
one_hot_data
```

	Outlook_overcast	Outlook_rainy	Outlook_sunny	Temperature_cool	Temperature_hot	Temperature_mild	Humidity_high	Humidity_normal	Windy_false
0	0	0	1	0	1	0	1	0	1
1	0	0	1	0	1	0	1	0	0
2	1	0	0	0	1	0	1	0	1
3	0	1	0	0	0	1	1	0	1
4	0	1	0	1	0	0	0	1	1
5	0	1	0	1	0	0	0	1	0
6	1	0	0	1	0	0	0	1	0
7	0	0	1	0	0	1	1	0	1
8	0	0	1	1	0	0	0	1	1
9	0	1	0	0	0	1	0	1	1
10	0	0	1	0	0	1	0	1	0
11	1	0	0	0	0	1	1	0	0
12	1	0	0	0	1	0	0	1	1
13	0	1	0	0	0	1	1	0	0

```
[ ] # The decision tree classifier.
clf = tree.DecisionTreeClassifier()
# Training the Decision Tree
clf_train = clf.fit(one_hot_data, golf_df['Play'])
```

Введите здесь текст для поиска

is601 ... Почт... DecTr... W Перп... W Доку... 15:42 10.03.2020

TREE - Google | DecTree.ipynb | Copy of DecTree | Regression.ipynb | Linear Regressio... | Linear Regressio... | Python Decision | +

colab.research.google.com/drive/1g0tuLeR0AvYVWIPjNx8jrDi35ZXGqX54?complete=0&safe=active#scrollTo=PulxLffwKFq

DecTree.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[3]	11	1	0	0	0	0	1	1	0	0
	12	1	0	0	0	1	0	0	1	1
	13	0	1	0	0	0	1	1	0	0

```

[4] # 3
    clf = tree.DecisionTreeClassifier()
    # Обучаем
    clf_train = clf.fit(one_hot_data, golf_df['Play'])

# 4
print(tree.export_graphviz(clf_train, None))

dot_data = tree.export_graphviz(clf_train, out_file=None, feature_names=list(one_hot_data.columns.values),
                                class_names=['Not_Play', 'Play'], rounded=True, filled=True) #Gini decides which attribute/feature should be placed at

graph = pydotplus.graph_from_dot_data(dot_data)
Image(graph.create_png())

3 [label="X[2] <= 0.5\ngini = 0.5\nsamples = 2\nvalue = [1, 1]" ;
2 -> 3 ;
4 [label="gini = 0.0\nsamples = 1\nvalue = [1, 0]" ;
3 -> 4 ;
5 [label="gini = 0.0\nsamples = 1\nvalue = [0, 1]" ;
3 -> 5 ;
6 [label="gini = 0.0\nsamples = 3\nvalue = [0, 3]" ;
2 -> 6 ;
7 [label="X[1] <= 0.5\ngini = 0.32\nsamples = 5\nvalue = [4, 1]" ;
1 -> 7 ;
8 [label="gini = 0.0\nsamples = 3\nvalue = [3, 0]" ;
7 -> 8 ;
9 [label="X[0] <= 0.5\ngini = 0.5\nsamples = 2\nvalue = [1, 1]" ;
7 -> 9 ;
10 [label="gini = 0.0\nsamples = 1\nvalue = [0, 1]" ;
9 -> 10 ;
11 [label="gini = 0.0\nsamples = 1\nvalue = [1, 0]" ;
9 -> 11 ;
12 [label="gini = 0.0\nsamples = 4\nvalue = [0, 4]" ;
0 -> 12 [labeldistance=2.5, labelangle=-45, headlabel="False" ;
]

```

Введите здесь текст для поиска

15:42 10.03.2020

```

class = Play
  True
    Humidity_high <= 0.5
      gini = 0.5
      samples = 10
      value = [5, 5]
      class = Not_Play
        Windy_false <= 0.5
          gini = 0.32
          samples = 5
          value = [1, 4]
          class = Play
            Outlook_sunny <= 0.5
              gini = 0.5
              samples = 2
              value = [1, 1]
              class = Not_Play
                gini = 0.0
                samples = 1
                value = [1, 0]
                class = Not_Play
                gini = 0.0
                samples = 1
                value = [0, 1]
                class = Play
            Outlook_rainy <= 0.5
              gini = 0.32
              samples = 5
              value = [4, 1]
              class = Not_Play
                gini = 0.0
                samples = 3
                value = [3, 0]
                class = Not_Play
                Windy_true <= 0.5
                  gini = 0.5
                  samples = 2
                  value = [1, 1]
                  class = Not_Play
                    gini = 0.0
                    samples = 1
                    value = [0, 1]
                    class = Play
                    gini = 0.0
                    samples = 1
                    value = [1, 0]
                    class = Not_Play
  False
    gini = 0.0
    samples = 4
    value = [0, 4]
    class = Play
  
```

```

# 5 Тестируем
# Outlook = sunny, Temperature = hot, Humidity = normal, Windy = false
prediction = clf_train.predict([[0,0,1,0,1,0,0,1,1,0]])
prediction

array(['yes'], dtype=object)
  
```

Задание 4

Сгенерируйте данные, распределенные вокруг функции, постройте дерево решений и выполните прогноз

$$y = N \cdot e^{-x^2} + 1.5 \cdot x^{-(x-2)^2}$$

N – номер варианта

```

n_train = 150
n_test = 1000
noise = 0.1

def f(x):
    x = x.ravel()
    return np.exp(-x ** 2) + 1.5 * np.exp(-(x - 3) ** 2)

def generate(n_samples, noise):
    X = np.random.rand(n_samples) * 10 - 5
    X = np.sort(X).ravel()
    y = np.exp(-X ** 2) + 1.5 * np.exp(-(X - 2) ** 2) + \
        np.random.normal(0.0, noise, n_samples)
    X = X.reshape((n_samples, 1))
    return X, y

X_train, y_train = generate(n_samples=n_train, noise=noise)
X_test, y_test = generate(n_samples=n_test, noise=noise)

reg_tree = DecisionTreeRegressor(max_depth=5, random_state=17)

reg_tree.fit(X_train, y_train)
reg_tree_pred = reg_tree.predict(X_test)

plt.figure(figsize=(10, 6))
plt.plot(X_test, f(X_test), "b")
plt.scatter(X_train, y_train, c="b", s=20)
plt.plot(X_test, reg_tree_pred, "g", lw=2)
plt.xlim([-5, 5])
plt.title("Decision tree regressor, MSE = %.2f" % np.sum((y_test - reg_tree_pred) ** 2))
plt.show()

```