РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра информационных технологий

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4

Дисциплина: Методы машинного обучения

Студент: Турсунов Баходурхон Азимджонович

Группа: Нфибд-03-19

Москва 2022

Вариант № 7

1. Загрузите заданный в индивидуальном задании набор данных из Tensorflow Datasets, включая указанные в задании независимые признаки и метку класса.

In [1]:

```python
# !pip install -q tfds-nightly
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import tensorflow as tf
from sklearn import metrics
import matplotlib.pyplot as plt
import numpy as np
from sklearn import metrics
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
import warnings
warnings.filterwarnings("ignore")
from mlxtend.plotting import plot_decision_regions
import pandas as pd
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve, auc
from sklearn.model_selection import train_test_split
import tensorflow_datasets as tfds
```

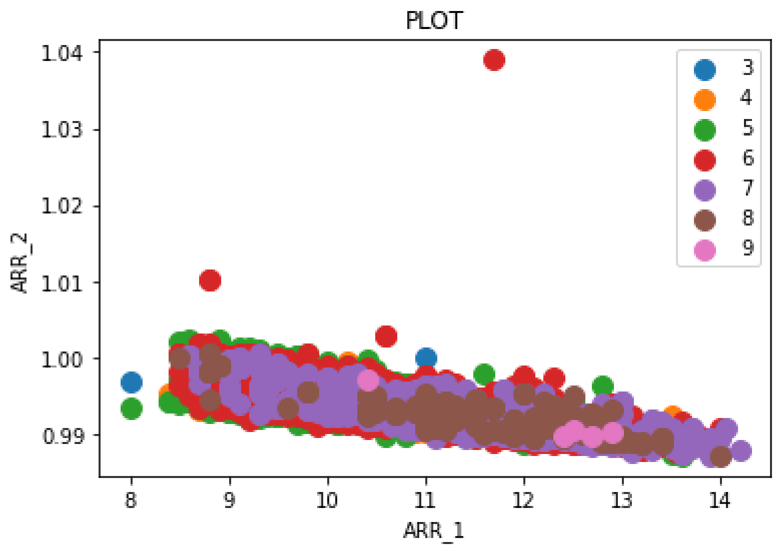
In [2]:

```python
ds = tfds.load("wine_quality", split='train')
df = tfds.as_dataframe(ds)
```

2. Визуализируйте точки набора данных на плоскости с координатами,

соответствующими двум независимым признакам, отображая точки различных классов разными цветами. Подпишите оси и рисунок, создайте легенду для классов набора данных.

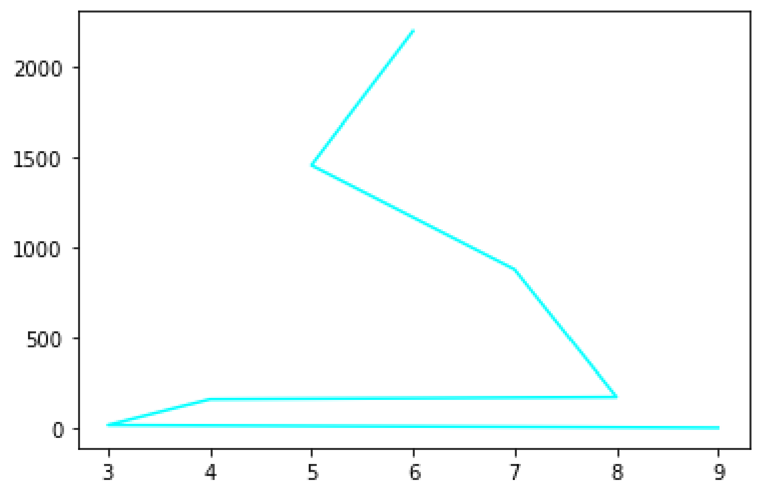
In [3]:
```
X=df[['features/alcohol', 'features/density']].values
y=df['quality']
fig = plt.figure()
ax = plt.axes()
for i in (np.unique(y)):
    row_ix = np.where(y == i)
    ax.scatter(X[row_ix, 0], X[row_ix, 1], s=100,label=i )
plt.title('PLOT')
plt.xlabel('ARR_1')
plt.ylabel('ARR_2')
plt.legend()
plt.show()
```



3. Если признак с метками классов содержит более двух классов, то объедините некоторые классы, чтобы получить набор для бинарной классификации. Объединяйте классы таким образом, чтобы положительный и отрицательный классы были сопоставимы по количеству точек.

In [4]:
```
y.value_counts().plot(color='cyan')
```

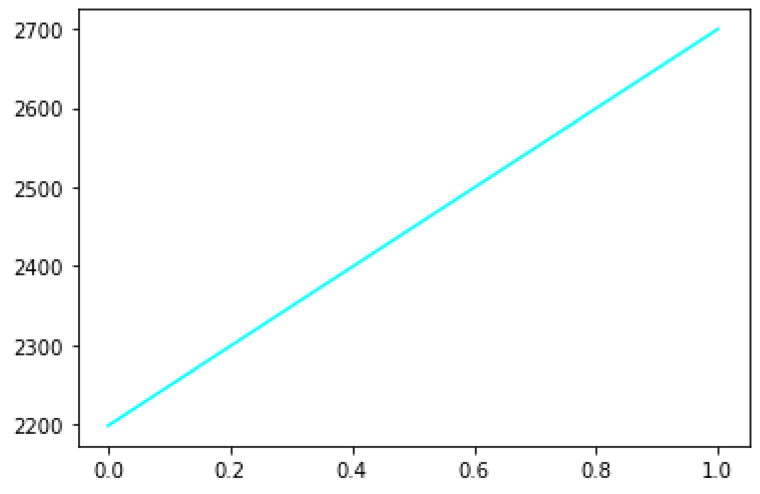
Out[4]:
```
<AxesSubplot:>
```



```
In [5]:  y = pd.Series([0 if species == 6 else 1 for species in df['quality']])
```

```
In [6]:  y.value_counts().plot(color='cyan')
```

```
Out[6]:  <AxesSubplot:>
```



4. Разбейте набор данных из двух признаков и меток класса на обучающую и тестовую выборки. Постройте нейронную сеть с нормализующим слоем и параметрами, указанными в индивидуальном задании, для бинарной классификации и обучите ее на обучающей выборке. Оцените качество бинарной классификации при помощи матрицы ошибок для тестовой выборки.

```
In [7]:  X_train, X_test, y_train, y_test = train_test_split(X, y)
         feature_normalizer = tf.keras.layers.Normalization(axis=None,input_shape=(X.shape[1]
         feature_normalizer.adapt(X_train)
```

```
model = tf.keras.Sequential([feature_normalizer,tf.keras.layers.Dense(128, activatio
])
model.compile(loss=tf.keras.losses.binary_crossentropy,metrics=[tf.keras.metrics.Bin
history = model.fit(X_train, y_train, epochs=200)
prediction = model.predict(X_test)
prediction = np.array([1 if prob > 0.5 else 0 for prob in np.ravel(prediction)])
np.unique(prediction)
```

```
Epoch 1/200
115/115 [==============================] - 2s 3ms/step - loss: 0.6876 - accuracy: 0.
5516
Epoch 2/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6847 - accuracy: 0.
5540
Epoch 3/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6828 - accuracy: 0.
5532
Epoch 4/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6819 - accuracy: 0.
5562
Epoch 5/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6803 - accuracy: 0.
5565
Epoch 6/200
115/115 [==============================] - 1s 5ms/step - loss: 0.6809 - accuracy: 0.
5633
Epoch 7/200
115/115 [==============================] - 1s 5ms/step - loss: 0.6793 - accuracy: 0.
5704
Epoch 8/200
115/115 [==============================] - 0s 4ms/step - loss: 0.6793 - accuracy: 0.
5701
Epoch 9/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6789 - accuracy: 0.
5459
Epoch 10/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6789 - accuracy: 0.
5649
Epoch 11/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6788 - accuracy: 0.
5521
Epoch 12/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6788 - accuracy: 0.
5608
Epoch 13/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6789 - accuracy: 0.
5614
Epoch 14/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6784 - accuracy: 0.
5622
Epoch 15/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6784 - accuracy: 0.
5546
Epoch 16/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6785 - accuracy: 0.
5576
Epoch 17/200
115/115 [==============================] - 1s 5ms/step - loss: 0.6777 - accuracy: 0.
5663
Epoch 18/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6775 - accuracy: 0.
5674
Epoch 19/200
115/115 [==============================] - 0s 4ms/step - loss: 0.6786 - accuracy: 0.
```

```
5614
Epoch 20/200
115/115 [==============================] - 1s 5ms/step - loss: 0.6780 - accuracy: 0.
5671
Epoch 21/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6784 - accuracy: 0.
5619
Epoch 22/200
115/115 [==============================] - 0s 4ms/step - loss: 0.6782 - accuracy: 0.
5666
Epoch 23/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6777 - accuracy: 0.
5625
Epoch 24/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6778 - accuracy: 0.
5565
Epoch 25/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6767 - accuracy: 0.
5644
Epoch 26/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6770 - accuracy: 0.
5663
Epoch 27/200
115/115 [==============================] - 0s 4ms/step - loss: 0.6772 - accuracy: 0.
5576
Epoch 28/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6779 - accuracy: 0.
5595
Epoch 29/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6775 - accuracy: 0.
5579
Epoch 30/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6776 - accuracy: 0.
5638
Epoch 31/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6776 - accuracy: 0.
5660
Epoch 32/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6775 - accuracy: 0.
5687
Epoch 33/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6766 - accuracy: 0.
5671
Epoch 34/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6775 - accuracy: 0.
5647
Epoch 35/200
115/115 [==============================] - 1s 5ms/step - loss: 0.6773 - accuracy: 0.
5674
Epoch 36/200
115/115 [==============================] - 0s 4ms/step - loss: 0.6776 - accuracy: 0.
5666
Epoch 37/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6772 - accuracy: 0.
5633
Epoch 38/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6771 - accuracy: 0.
5649
Epoch 39/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6774 - accuracy: 0.
5625
Epoch 40/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6773 - accuracy: 0.
5568
```

```
Epoch 41/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6774 - accuracy: 0.
5617
Epoch 42/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6766 - accuracy: 0.
5668
Epoch 43/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6777 - accuracy: 0.
5633
Epoch 44/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6770 - accuracy: 0.
5573
Epoch 45/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6773 - accuracy: 0.
5598
Epoch 46/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6763 - accuracy: 0.
5633
Epoch 47/200
115/115 [==============================] - 0s 4ms/step - loss: 0.6776 - accuracy: 0.
5668
Epoch 48/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6765 - accuracy: 0.
5603
Epoch 49/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6771 - accuracy: 0.
5603
Epoch 50/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6771 - accuracy: 0.
5614
Epoch 51/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6766 - accuracy: 0.
5655
Epoch 52/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6770 - accuracy: 0.
5619
Epoch 53/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6766 - accuracy: 0.
5677
Epoch 54/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6766 - accuracy: 0.
5617
Epoch 55/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6767 - accuracy: 0.
5728
Epoch 56/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6770 - accuracy: 0.
5663
Epoch 57/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6766 - accuracy: 0.
5598
Epoch 58/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6770 - accuracy: 0.
5608
Epoch 59/200
115/115 [==============================] - 1s 5ms/step - loss: 0.6768 - accuracy: 0.
5592
Epoch 60/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6766 - accuracy: 0.
5698
Epoch 61/200
115/115 [==============================] - 0s 2ms/step - loss: 0.6769 - accuracy: 0.
5652
Epoch 62/200
```

```
115/115 [==============================] - 0s 3ms/step - loss: 0.6764 - accuracy: 0.
5693
Epoch 63/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6774 - accuracy: 0.
5598
Epoch 64/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6765 - accuracy: 0.
5592
Epoch 65/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6768 - accuracy: 0.
5641
Epoch 66/200
115/115 [==============================] - 0s 4ms/step - loss: 0.6771 - accuracy: 0.
5685
Epoch 67/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6757 - accuracy: 0.
5704
Epoch 68/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6766 - accuracy: 0.
5685
Epoch 69/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6758 - accuracy: 0.
5644
Epoch 70/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6759 - accuracy: 0.
5619
Epoch 71/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6770 - accuracy: 0.
5671
Epoch 72/200
115/115 [==============================] - 0s 4ms/step - loss: 0.6767 - accuracy: 0.
5628
Epoch 73/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6758 - accuracy: 0.
5693
Epoch 74/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6773 - accuracy: 0.
5630
Epoch 75/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6773 - accuracy: 0.
5633
Epoch 76/200
115/115 [==============================] - 0s 2ms/step - loss: 0.6761 - accuracy: 0.
5674
Epoch 77/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6774 - accuracy: 0.
5622
Epoch 78/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6772 - accuracy: 0.
5598
Epoch 79/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6763 - accuracy: 0.
5723
Epoch 80/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6769 - accuracy: 0.
5628
Epoch 81/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6764 - accuracy: 0.
5625
Epoch 82/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6773 - accuracy: 0.
5655
Epoch 83/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6768 - accuracy: 0.
```

```
5608
Epoch 84/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6772 - accuracy: 0.
5666
Epoch 85/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6766 - accuracy: 0.
5652
Epoch 86/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6770 - accuracy: 0.
5666
Epoch 87/200
115/115 [==============================] - 0s 2ms/step - loss: 0.6761 - accuracy: 0.
5660
Epoch 88/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6769 - accuracy: 0.
5630
Epoch 89/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6761 - accuracy: 0.
5658
Epoch 90/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6764 - accuracy: 0.
5666
Epoch 91/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6763 - accuracy: 0.
5682
Epoch 92/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6770 - accuracy: 0.
5598
Epoch 93/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6767 - accuracy: 0.
5674
Epoch 94/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6758 - accuracy: 0.
5633
Epoch 95/200
115/115 [==============================] - 0s 4ms/step - loss: 0.6775 - accuracy: 0.
5660
Epoch 96/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6756 - accuracy: 0.
5668
Epoch 97/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6769 - accuracy: 0.
5614
Epoch 98/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6763 - accuracy: 0.
5622
Epoch 99/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6757 - accuracy: 0.
5668
Epoch 100/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6748 - accuracy: 0.
5668
Epoch 101/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6758 - accuracy: 0.
5677
Epoch 102/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6767 - accuracy: 0.
5723
Epoch 103/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6761 - accuracy: 0.
5668
Epoch 104/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6759 - accuracy: 0.
5652
```

```
Epoch 105/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6771 - accuracy: 0.
5649
Epoch 106/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6756 - accuracy: 0.
5682
Epoch 107/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6770 - accuracy: 0.
5606
Epoch 108/200
115/115 [==============================] - 0s 4ms/step - loss: 0.6769 - accuracy: 0.
5647
Epoch 109/200
115/115 [==============================] - 1s 5ms/step - loss: 0.6763 - accuracy: 0.
5606
Epoch 110/200
115/115 [==============================] - 0s 4ms/step - loss: 0.6760 - accuracy: 0.
5649
Epoch 111/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6765 - accuracy: 0.
5666
Epoch 112/200
115/115 [==============================] - 0s 4ms/step - loss: 0.6770 - accuracy: 0.
5641
Epoch 113/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6767 - accuracy: 0.
5647
Epoch 114/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6765 - accuracy: 0.
5734
Epoch 115/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6762 - accuracy: 0.
5660
Epoch 116/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6762 - accuracy: 0.
5644
Epoch 117/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6757 - accuracy: 0.
5628
Epoch 118/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6760 - accuracy: 0.
5587
Epoch 119/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6767 - accuracy: 0.
5717
Epoch 120/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6754 - accuracy: 0.
5701
Epoch 121/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6758 - accuracy: 0.
5663
Epoch 122/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6767 - accuracy: 0.
5638
Epoch 123/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6761 - accuracy: 0.
5668
Epoch 124/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6763 - accuracy: 0.
5592
Epoch 125/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6766 - accuracy: 0.
5720
Epoch 126/200
```

```
115/115 [==============================] - 0s 3ms/step - loss: 0.6755 - accuracy: 0.
5715
Epoch 127/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6767 - accuracy: 0.
5633
Epoch 128/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6772 - accuracy: 0.
5655
Epoch 129/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6753 - accuracy: 0.
5685
Epoch 130/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6761 - accuracy: 0.
5690
Epoch 131/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6753 - accuracy: 0.
5638
Epoch 132/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6768 - accuracy: 0.
5677
Epoch 133/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6751 - accuracy: 0.
5617
Epoch 134/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6769 - accuracy: 0.
5622
Epoch 135/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6765 - accuracy: 0.
5625
Epoch 136/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6765 - accuracy: 0.
5655
Epoch 137/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6760 - accuracy: 0.
5608
Epoch 138/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6761 - accuracy: 0.
5658
Epoch 139/200
115/115 [==============================] - 0s 2ms/step - loss: 0.6758 - accuracy: 0.
5690
Epoch 140/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6757 - accuracy: 0.
5652
Epoch 141/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6763 - accuracy: 0.
5638
Epoch 142/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6752 - accuracy: 0.
5687
Epoch 143/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6766 - accuracy: 0.
5685
Epoch 144/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6764 - accuracy: 0.
5584
Epoch 145/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6770 - accuracy: 0.
5638
Epoch 146/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6756 - accuracy: 0.
5663
Epoch 147/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6773 - accuracy: 0.
```

```
5641
Epoch 148/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6755 - accuracy: 0.
5701
Epoch 149/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6760 - accuracy: 0.
5658
Epoch 150/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6764 - accuracy: 0.
5668
Epoch 151/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6764 - accuracy: 0.
5638
Epoch 152/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6761 - accuracy: 0.
5677
Epoch 153/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6765 - accuracy: 0.
5663
Epoch 154/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6765 - accuracy: 0.
5636
Epoch 155/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6767 - accuracy: 0.
5674
Epoch 156/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6758 - accuracy: 0.
5720
Epoch 157/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6761 - accuracy: 0.
5655
Epoch 158/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6759 - accuracy: 0.
5690
Epoch 159/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6757 - accuracy: 0.
5641
Epoch 160/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6766 - accuracy: 0.
5589
Epoch 161/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6765 - accuracy: 0.
5608
Epoch 162/200
115/115 [==============================] - 0s 4ms/step - loss: 0.6766 - accuracy: 0.
5696
Epoch 163/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6766 - accuracy: 0.
5666
Epoch 164/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6765 - accuracy: 0.
5628
Epoch 165/200
115/115 [==============================] - 0s 4ms/step - loss: 0.6769 - accuracy: 0.
5625
Epoch 166/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6759 - accuracy: 0.
5652
Epoch 167/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6766 - accuracy: 0.
5674
Epoch 168/200
115/115 [==============================] - 0s 4ms/step - loss: 0.6759 - accuracy: 0.
5677
```

```
Epoch 169/200
115/115 [==============================] - 0s 4ms/step - loss: 0.6766 - accuracy: 0.
5587
Epoch 170/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6767 - accuracy: 0.
5647
Epoch 171/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6757 - accuracy: 0.
5682
Epoch 172/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6768 - accuracy: 0.
5595
Epoch 173/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6759 - accuracy: 0.
5644
Epoch 174/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6769 - accuracy: 0.
5658
Epoch 175/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6769 - accuracy: 0.
5628
Epoch 176/200
115/115 [==============================] - 0s 4ms/step - loss: 0.6757 - accuracy: 0.
5638
Epoch 177/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6759 - accuracy: 0.
5644
Epoch 178/200
115/115 [==============================] - 0s 4ms/step - loss: 0.6764 - accuracy: 0.
5655
Epoch 179/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6765 - accuracy: 0.
5625
Epoch 180/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6761 - accuracy: 0.
5647
Epoch 181/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6767 - accuracy: 0.
5641
Epoch 182/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6759 - accuracy: 0.
5600
Epoch 183/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6754 - accuracy: 0.
5641
Epoch 184/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6760 - accuracy: 0.
5709
Epoch 185/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6763 - accuracy: 0.
5668
Epoch 186/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6760 - accuracy: 0.
5652
Epoch 187/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6764 - accuracy: 0.
5655
Epoch 188/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6763 - accuracy: 0.
5622
Epoch 189/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6760 - accuracy: 0.
5628
Epoch 190/200
```

```
115/115 [==============================] - 0s 3ms/step - loss: 0.6765 - accuracy: 0.
5712
Epoch 191/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6762 - accuracy: 0.
5641
Epoch 192/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6757 - accuracy: 0.
5625
Epoch 193/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6762 - accuracy: 0.
5663
Epoch 194/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6761 - accuracy: 0.
5655
Epoch 195/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6762 - accuracy: 0.
5674
Epoch 196/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6761 - accuracy: 0.
5652
Epoch 197/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6771 - accuracy: 0.
5630
Epoch 198/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6769 - accuracy: 0.
5663
Epoch 199/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6759 - accuracy: 0.
5668
Epoch 200/200
115/115 [==============================] - 0s 3ms/step - loss: 0.6768 - accuracy: 0.
5668
```

Out[7]:

```
array([0, 1])
```

5. Визуализируйте границы принятия решений построенной нейронной сетью на обучающей и тестовой выборках.

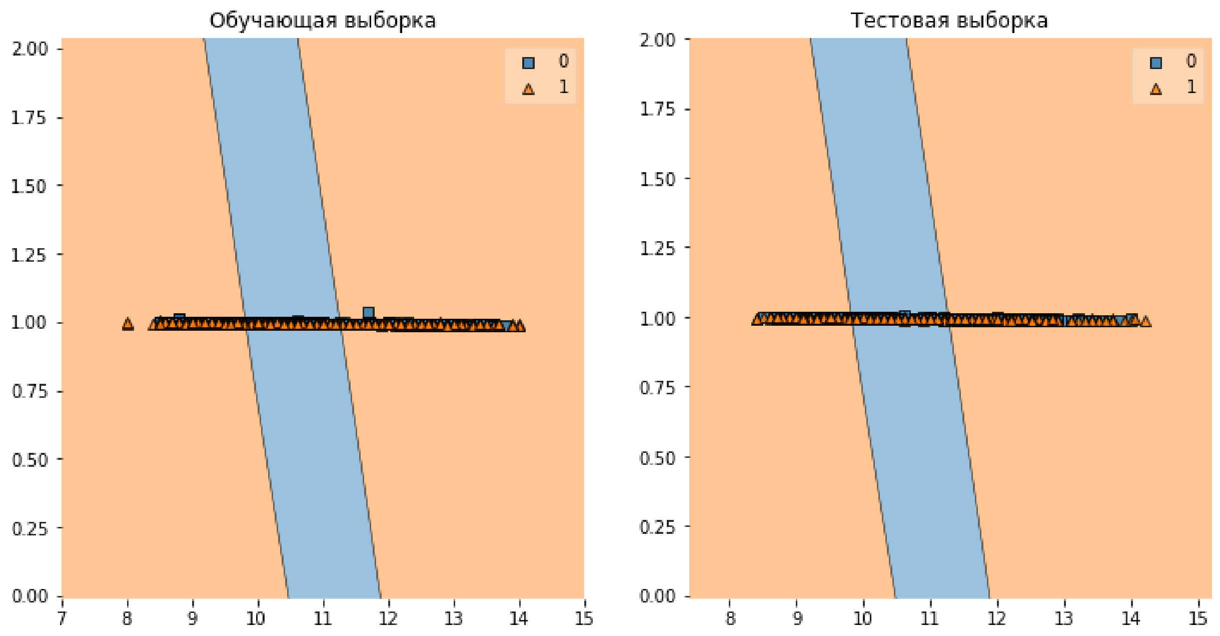
In [8]:

```python
import warnings
warnings.filterwarnings("ignore")

from mlxtend.plotting import plot_decision_regions
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.title("Обучающая выборка")
plot_decision_regions(X_train, y_train.values, model)
plt.subplot(1, 2, 2)
plt.title("Тестовая выборка")
plot_decision_regions(X_test, y_test.values, model)
```

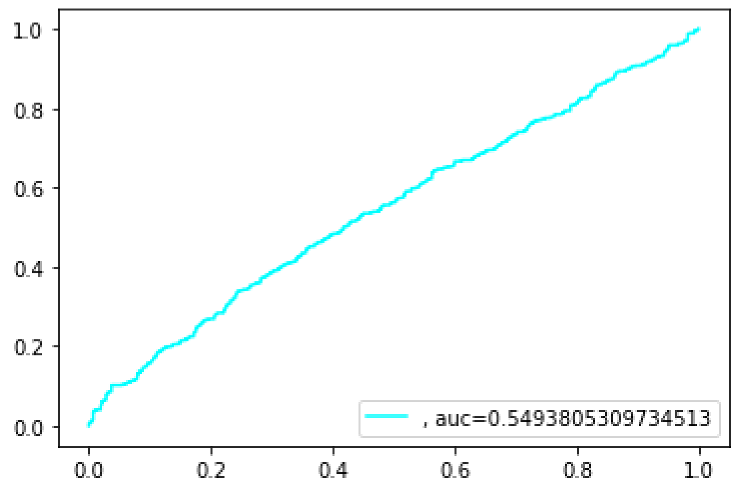
Out[8]:

```
<AxesSubplot:title={'center':'Тестовая выборка'}>
```



6. Визуализируйте ROC-кривую для построенного классификатора и вычислите площадь под ROC-кривой методом трапеций или иным методом.

In [9]:
```
fpr, tpr, _ = metrics.roc_curve(y_test,  model.predict(X_test))
auc = metrics.roc_auc_score(y_test, model.predict(X_test))
plt.plot(fpr,tpr,label=", auc="+str(auc),color='cyan')
plt.legend(loc=4)
plt.show()
```



7. Обучите на полном наборе данных нейронную сеть с одним слоем и одним выходным нейроном с функцией активации сигмоида и определите дополнительный признак, отличный от указанных в задании двух независимых

признаков, принимающий непрерывные значения и являющийся важным по абсолютному значению веса в обученной нейронной сети.

In [10]:

```
X = np.array(df.drop('quality', axis=1))
y = np.array(df['quality'])
feature_normalizer = tf.keras.layers.Normalization(axis=None,input_shape=(X.shape[1]
feature_normalizer.adapt(X)
model_aux = tf.keras.Sequential([feature_normalizer,tf.keras.layers.Dense(1, activat
])
model_aux.compile(loss=tf.keras.losses.binary_crossentropy)
model_aux.fit(X, y, epochs=100, verbose=0);
model_aux.layers[1].kernel
```

Out[10]:

```
<tf.Variable 'dense_4/kernel:0' shape=(11, 1) dtype=float32, numpy=
array([[-15.089177 ],
       [-15.275103 ],
       [-15.934899 ],
       [-15.687599 ],
       [-15.312748 ],
       [ 14.502067 ],
       [-14.912778 ],
       [-15.99827  ],
       [-14.7400255],
       [ 14.790203 ],
       [-16.062008 ]], dtype=float32)>
```

In [17]:

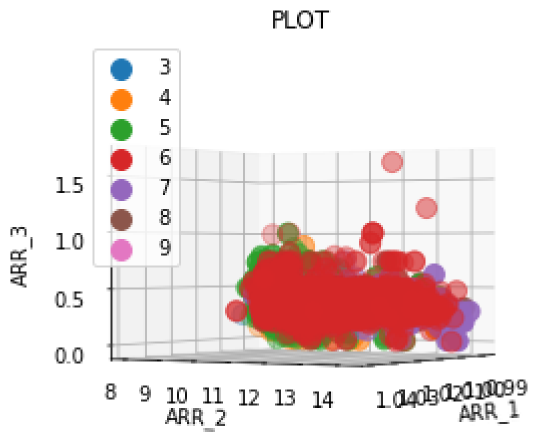
```
X = df[['features/density', 'features/alcohol', 'features/citric acid']].values
y = df['quality'].values
```

8. Визуализируйте точки набора данных в трехмерном пространстве с координатами, соответствующими трем независимым признакам, отображая точки различных классов разными цветами. Подпишите оси и рисунок, создайте легенду для классов набора данных.

In [20]:

```
fig = plt.figure()
ax = plt.axes(projection='3d')
for i in (np.unique(y)):
    row_ix = np.where(y == i)
    ax.scatter(X[row_ix, 0], X[row_ix, 1], X[row_ix, 2],s=100,label=i )
plt.title('PLOT')
plt.xlabel('ARR_1')
plt.ylabel('ARR_2')
ax.set_zlabel('ARR_3')
plt.legend()
```

```
ax.view_init( azim=30, elev=0 )
plt.show()
```



9. Разбейте полный набор данных на обучающую и тестовую выборки. Постройте нейронную сеть с нормализующим слоем и параметрами, указанными в индивидуальном задании, для многоклассовой классификации и обучите ее на обучающей выборке.

In [13]:

```
X_train, X_test, y_train, y_test = train_test_split(df.drop('quality', axis=1).value
def to_one_hot(labels, dimension=11):
    results = np.zeros((len(labels), dimension))
    for i, label in enumerate(labels):
        results[i, label] = 1.
    return results
y_train = to_one_hot(y_train)
y_test = to_one_hot(y_test)
y_train.shape, y_test.shape
feature_normalizer = tf.keras.layers.Normalization(axis=None,input_shape=(X_train.sh
feature_normalizer.adapt(X_train)
model = tf.keras.Sequential([feature_normalizer,tf.keras.layers.Dense(128, activatio
model.compile(optimizer="rmsprop",loss="categorical_crossentropy",metrics=["accuracy
history = model.fit(X_train,y_train,epochs=100,verbose=1,validation_split = 0.2)
```

```
Epoch 1/100
92/92 [==============================] - 1s 5ms/step - loss: 1.3675 - accuracy: 0.41
83 - val_loss: 1.2781 - val_accuracy: 0.4857
Epoch 2/100
92/92 [==============================] - 0s 3ms/step - loss: 1.2819 - accuracy: 0.42
99 - val_loss: 1.2715 - val_accuracy: 0.4776
Epoch 3/100
92/92 [==============================] - 0s 3ms/step - loss: 1.2552 - accuracy: 0.44
08 - val_loss: 1.2205 - val_accuracy: 0.4844
Epoch 4/100
92/92 [==============================] - 0s 3ms/step - loss: 1.2415 - accuracy: 0.43
33 - val_loss: 1.2276 - val_accuracy: 0.4925
Epoch 5/100
92/92 [==============================] - 0s 3ms/step - loss: 1.2290 - accuracy: 0.44
```

```
28 - val_loss: 1.2222 - val_accuracy: 0.4898
Epoch 6/100
92/92 [==============================] - 0s 4ms/step - loss: 1.2187 - accuracy: 0.44
72 - val_loss: 1.2017 - val_accuracy: 0.4898
Epoch 7/100
92/92 [==============================] - 0s 3ms/step - loss: 1.2190 - accuracy: 0.44
62 - val_loss: 1.2189 - val_accuracy: 0.4925
Epoch 8/100
92/92 [==============================] - 0s 3ms/step - loss: 1.2095 - accuracy: 0.44
55 - val_loss: 1.1907 - val_accuracy: 0.4993
Epoch 9/100
92/92 [==============================] - 0s 3ms/step - loss: 1.2039 - accuracy: 0.45
81 - val_loss: 1.2133 - val_accuracy: 0.4694
Epoch 10/100
92/92 [==============================] - 0s 3ms/step - loss: 1.2023 - accuracy: 0.45
10 - val_loss: 1.1788 - val_accuracy: 0.5075
Epoch 11/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1961 - accuracy: 0.46
77 - val_loss: 1.1594 - val_accuracy: 0.4993
Epoch 12/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1824 - accuracy: 0.46
90 - val_loss: 1.2013 - val_accuracy: 0.4490
Epoch 13/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1810 - accuracy: 0.47
41 - val_loss: 1.1511 - val_accuracy: 0.5129
Epoch 14/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1770 - accuracy: 0.47
52 - val_loss: 1.1671 - val_accuracy: 0.5020
Epoch 15/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1615 - accuracy: 0.47
89 - val_loss: 1.1442 - val_accuracy: 0.4925
Epoch 16/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1596 - accuracy: 0.49
01 - val_loss: 1.2416 - val_accuracy: 0.4340
Epoch 17/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1612 - accuracy: 0.48
06 - val_loss: 1.1326 - val_accuracy: 0.5265
Epoch 18/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1585 - accuracy: 0.48
71 - val_loss: 1.1333 - val_accuracy: 0.5265
Epoch 19/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1541 - accuracy: 0.47
69 - val_loss: 1.1515 - val_accuracy: 0.5279
Epoch 20/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1547 - accuracy: 0.49
18 - val_loss: 1.1258 - val_accuracy: 0.5143
Epoch 21/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1478 - accuracy: 0.48
30 - val_loss: 1.1456 - val_accuracy: 0.4871
Epoch 22/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1464 - accuracy: 0.49
39 - val_loss: 1.1348 - val_accuracy: 0.5088
Epoch 23/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1429 - accuracy: 0.48
57 - val_loss: 1.1260 - val_accuracy: 0.5197
Epoch 24/100
92/92 [==============================] - 0s 4ms/step - loss: 1.1421 - accuracy: 0.49
66 - val_loss: 1.1289 - val_accuracy: 0.5293
Epoch 25/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1388 - accuracy: 0.49
15 - val_loss: 1.1188 - val_accuracy: 0.5116
Epoch 26/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1389 - accuracy: 0.49
39 - val_loss: 1.1666 - val_accuracy: 0.4789
```

```
Epoch 27/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1391 - accuracy: 0.49
42 - val_loss: 1.1509 - val_accuracy: 0.5061
Epoch 28/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1351 - accuracy: 0.49
93 - val_loss: 1.1411 - val_accuracy: 0.5265
Epoch 29/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1319 - accuracy: 0.50
07 - val_loss: 1.1275 - val_accuracy: 0.5156
Epoch 30/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1267 - accuracy: 0.50
68 - val_loss: 1.1336 - val_accuracy: 0.5075
Epoch 31/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1345 - accuracy: 0.48
43 - val_loss: 1.1721 - val_accuracy: 0.4789
Epoch 32/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1289 - accuracy: 0.50
24 - val_loss: 1.1440 - val_accuracy: 0.5088
Epoch 33/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1252 - accuracy: 0.49
29 - val_loss: 1.1155 - val_accuracy: 0.5265
Epoch 34/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1262 - accuracy: 0.50
17 - val_loss: 1.1298 - val_accuracy: 0.5143
Epoch 35/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1254 - accuracy: 0.50
61 - val_loss: 1.1192 - val_accuracy: 0.5048
Epoch 36/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1217 - accuracy: 0.49
97 - val_loss: 1.1480 - val_accuracy: 0.5034
Epoch 37/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1206 - accuracy: 0.51
29 - val_loss: 1.1409 - val_accuracy: 0.5088
Epoch 38/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1213 - accuracy: 0.51
26 - val_loss: 1.1434 - val_accuracy: 0.5020
Epoch 39/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1205 - accuracy: 0.51
57 - val_loss: 1.1153 - val_accuracy: 0.5374
Epoch 40/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1193 - accuracy: 0.51
09 - val_loss: 1.1466 - val_accuracy: 0.5170
Epoch 41/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1177 - accuracy: 0.50
00 - val_loss: 1.1282 - val_accuracy: 0.5034
Epoch 42/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1130 - accuracy: 0.50
44 - val_loss: 1.1133 - val_accuracy: 0.5361
Epoch 43/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1153 - accuracy: 0.50
27 - val_loss: 1.1211 - val_accuracy: 0.5156
Epoch 44/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1113 - accuracy: 0.51
53 - val_loss: 1.1185 - val_accuracy: 0.5361
Epoch 45/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1138 - accuracy: 0.50
65 - val_loss: 1.1331 - val_accuracy: 0.5252
Epoch 46/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1109 - accuracy: 0.51
50 - val_loss: 1.1876 - val_accuracy: 0.4844
Epoch 47/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1056 - accuracy: 0.50
88 - val_loss: 1.1366 - val_accuracy: 0.5252
Epoch 48/100
```

```
92/92 [==============================] - 0s 3ms/step - loss: 1.1062 - accuracy: 0.50
75 - val_loss: 1.1232 - val_accuracy: 0.5388
Epoch 49/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1047 - accuracy: 0.51
97 - val_loss: 1.1390 - val_accuracy: 0.5184
Epoch 50/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1069 - accuracy: 0.50
85 - val_loss: 1.1393 - val_accuracy: 0.5224
Epoch 51/100
92/92 [==============================] - 0s 3ms/step - loss: 1.1039 - accuracy: 0.51
63 - val_loss: 1.1226 - val_accuracy: 0.5265
Epoch 52/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0999 - accuracy: 0.50
34 - val_loss: 1.1422 - val_accuracy: 0.5061
Epoch 53/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0982 - accuracy: 0.51
60 - val_loss: 1.1688 - val_accuracy: 0.4925
Epoch 54/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0988 - accuracy: 0.51
40 - val_loss: 1.1634 - val_accuracy: 0.5116
Epoch 55/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0980 - accuracy: 0.51
23 - val_loss: 1.1335 - val_accuracy: 0.5238
Epoch 56/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0956 - accuracy: 0.51
02 - val_loss: 1.1549 - val_accuracy: 0.5306
Epoch 57/100
92/92 [==============================] - 0s 4ms/step - loss: 1.0936 - accuracy: 0.51
94 - val_loss: 1.1396 - val_accuracy: 0.5211
Epoch 58/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0948 - accuracy: 0.51
97 - val_loss: 1.1545 - val_accuracy: 0.5224
Epoch 59/100
92/92 [==============================] - 0s 4ms/step - loss: 1.0970 - accuracy: 0.51
46 - val_loss: 1.2749 - val_accuracy: 0.4517
Epoch 60/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0958 - accuracy: 0.51
77 - val_loss: 1.1381 - val_accuracy: 0.5238
Epoch 61/100
92/92 [==============================] - 1s 5ms/step - loss: 1.0908 - accuracy: 0.51
97 - val_loss: 1.1605 - val_accuracy: 0.5102
Epoch 62/100
92/92 [==============================] - 0s 4ms/step - loss: 1.0876 - accuracy: 0.52
72 - val_loss: 1.1466 - val_accuracy: 0.5252
Epoch 63/100
92/92 [==============================] - 0s 4ms/step - loss: 1.0898 - accuracy: 0.52
01 - val_loss: 1.1314 - val_accuracy: 0.5333
Epoch 64/100
92/92 [==============================] - 0s 4ms/step - loss: 1.0819 - accuracy: 0.52
76 - val_loss: 1.1430 - val_accuracy: 0.5306
Epoch 65/100
92/92 [==============================] - 0s 4ms/step - loss: 1.0857 - accuracy: 0.52
11 - val_loss: 1.1574 - val_accuracy: 0.5320
Epoch 66/100
92/92 [==============================] - 0s 4ms/step - loss: 1.0863 - accuracy: 0.52
42 - val_loss: 1.1601 - val_accuracy: 0.5143
Epoch 67/100
92/92 [==============================] - 0s 4ms/step - loss: 1.0789 - accuracy: 0.52
21 - val_loss: 1.1390 - val_accuracy: 0.5306
Epoch 68/100
92/92 [==============================] - 0s 4ms/step - loss: 1.0788 - accuracy: 0.52
11 - val_loss: 1.1523 - val_accuracy: 0.5265
Epoch 69/100
92/92 [==============================] - 0s 4ms/step - loss: 1.0768 - accuracy: 0.53
```

```
34 - val_loss: 1.1561 - val_accuracy: 0.5293
Epoch 70/100
92/92 [==============================] - 0s 4ms/step - loss: 1.0775 - accuracy: 0.53
85 - val_loss: 1.1409 - val_accuracy: 0.5306
Epoch 71/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0696 - accuracy: 0.53
78 - val_loss: 1.1750 - val_accuracy: 0.5170
Epoch 72/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0788 - accuracy: 0.52
93 - val_loss: 1.1712 - val_accuracy: 0.5156
Epoch 73/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0742 - accuracy: 0.52
72 - val_loss: 1.2124 - val_accuracy: 0.5143
Epoch 74/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0736 - accuracy: 0.53
47 - val_loss: 1.1583 - val_accuracy: 0.5306
Epoch 75/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0769 - accuracy: 0.52
72 - val_loss: 1.1625 - val_accuracy: 0.5279
Epoch 76/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0739 - accuracy: 0.53
71 - val_loss: 1.2058 - val_accuracy: 0.5075
Epoch 77/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0720 - accuracy: 0.52
89 - val_loss: 1.1511 - val_accuracy: 0.5279
Epoch 78/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0674 - accuracy: 0.54
29 - val_loss: 1.1868 - val_accuracy: 0.5306
Epoch 79/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0677 - accuracy: 0.54
12 - val_loss: 1.1753 - val_accuracy: 0.5184
Epoch 80/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0633 - accuracy: 0.53
30 - val_loss: 1.1647 - val_accuracy: 0.5279
Epoch 81/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0649 - accuracy: 0.53
06 - val_loss: 1.1622 - val_accuracy: 0.5361
Epoch 82/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0594 - accuracy: 0.53
17 - val_loss: 1.1824 - val_accuracy: 0.5293
Epoch 83/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0582 - accuracy: 0.54
29 - val_loss: 1.2200 - val_accuracy: 0.5361
Epoch 84/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0636 - accuracy: 0.53
00 - val_loss: 1.1892 - val_accuracy: 0.4939
Epoch 85/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0620 - accuracy: 0.53
10 - val_loss: 1.2220 - val_accuracy: 0.5102
Epoch 86/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0549 - accuracy: 0.53
81 - val_loss: 1.2155 - val_accuracy: 0.5293
Epoch 87/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0551 - accuracy: 0.53
20 - val_loss: 1.1922 - val_accuracy: 0.4993
Epoch 88/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0547 - accuracy: 0.51
97 - val_loss: 1.2735 - val_accuracy: 0.5143
Epoch 89/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0558 - accuracy: 0.55
11 - val_loss: 1.1982 - val_accuracy: 0.5442
Epoch 90/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0515 - accuracy: 0.54
12 - val_loss: 1.2305 - val_accuracy: 0.5211
```

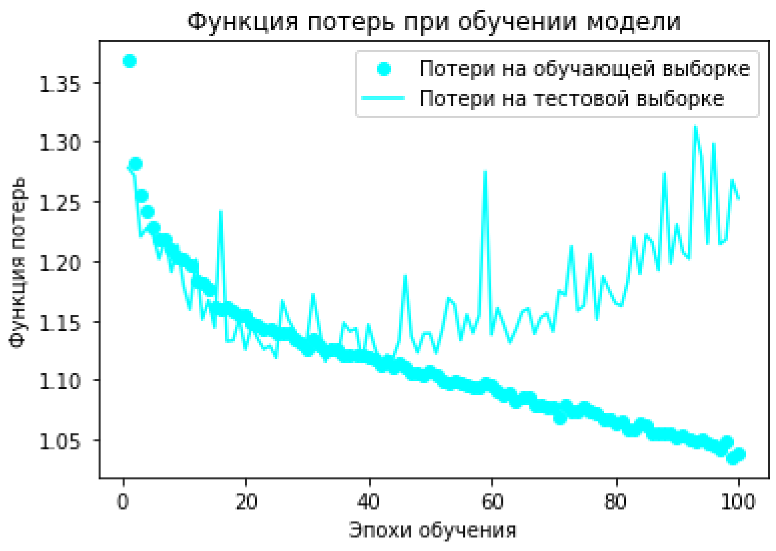
```
Epoch 91/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0545 - accuracy: 0.53
40 - val_loss: 1.2073 - val_accuracy: 0.5238
Epoch 92/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0505 - accuracy: 0.54
22 - val_loss: 1.2019 - val_accuracy: 0.5320
Epoch 93/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0485 - accuracy: 0.53
91 - val_loss: 1.3120 - val_accuracy: 0.4558
Epoch 94/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0498 - accuracy: 0.53
54 - val_loss: 1.2874 - val_accuracy: 0.4980
Epoch 95/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0478 - accuracy: 0.53
74 - val_loss: 1.2147 - val_accuracy: 0.5238
Epoch 96/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0460 - accuracy: 0.53
10 - val_loss: 1.2983 - val_accuracy: 0.4612
Epoch 97/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0425 - accuracy: 0.54
29 - val_loss: 1.2141 - val_accuracy: 0.5265
Epoch 98/100
92/92 [==============================] - 0s 3ms/step - loss: 1.0493 - accuracy: 0.53
85 - val_loss: 1.2179 - val_accuracy: 0.5524
Epoch 99/100
92/92 [==============================] - 0s 4ms/step - loss: 1.0355 - accuracy: 0.55
04 - val_loss: 1.2678 - val_accuracy: 0.5320
Epoch 100/100
92/92 [==============================] - 0s 4ms/step - loss: 1.0386 - accuracy: 0.53
71 - val_loss: 1.2525 - val_accuracy: 0.5211
```

In [14]:
```
model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 normalization_2 (Normalizat  (None, 11)               3         
 ion)                                                            
                                                                 
 dense_5 (Dense)             (None, 128)               1536      
                                                                 
 dense_6 (Dense)             (None, 128)               16512     
                                                                 
 dense_7 (Dense)             (None, 128)               16512     
                                                                 
 dense_8 (Dense)             (None, 11)                1419      
                                                                 
=================================================================
Total params: 35,982
Trainable params: 35,979
Non-trainable params: 3
_________________________________________________________________
```

10. Постройте кривые обучения в зависимости от эпохи обучения, подписывая оси и рисунок и создавая легенду.

```
In [15]:  loss = history.history["loss"]
          val_loss = history.history["val_loss"]
          epochs = range(1, len(loss) + 1)
          plt.plot(epochs, loss, "bo", label="Потери на обучающей выборке",color='cyan')
          plt.plot(epochs, val_loss, "b", label="Потери на тестовой выборке",color='cyan')
          plt.title("Функция потерь при обучении модели")
          plt.xlabel("Эпохи обучения")
          plt.ylabel("Функция потерь")
          plt.legend();
```


Функция потерь при обучении модели
Потери на обучающей выборке
Потери на тестовой выборке
Функция потерь
Эпохи обучения