

Тест начат	Пятница, 8 декабря 2023, 14:54
Состояние	Завершены
Завершен	Пятница, 8 декабря 2023, 16:35
Прошло времени	1 ч. 40 мин.
Баллы	6,00/6,00
Оценка	10,00 из 10,00 (100%)

Вопрос 1

Верно

Баллов: 1,00 из 1,00

Линейная регрессия.

Вам даны два вектора признаков и вектор целевой переменной. Постройте модель линейной регрессии и выведите на экран вектор весов модели для факторов x_1 и x_2 в формате массива **numpy**.

Ответ: (штрафной режим: 0%)

Сбросить ответ

```
1 import numpy as np
2 from sklearn.linear_model import LinearRegression
3 x1 = np.array([1, 2, 3, 4, 5])
4 x2 = np.array([6, 7, 8, 9, 10])
5 y = np.array([11, 12, 13, 14, 15])
6 reg = LinearRegression()
7 X = np.column_stack((x1, x2))
8 reg.fit(X, y)
9 print(reg.coef_)
```

	Тест	Ожидается	Получил	
✓	print(reg.coef_[0] == 0.5)	True	True	✓
✓	print(reg.coef_[1] == 0.50000000000000001)	True	True	✓

Все тесты пройдены! ✓

Верно
Баллы за эту попытку: 1,00/1,00.

Вопрос **2**

Верно

Баллов: 1,00 из 1,00

Линейная регрессия (продолжение)

Вам даны два вектора признаков и вектор целевой переменной. Постройте модель линейной регрессии и выведите на экран предсказанное значение y если $x_1 = 5$, а $x_2 = 6$.

Ответ: (штрафной режим: 0%)

Сбросить ответ

```
1 import numpy as np
2 from sklearn.linear_model import LinearRegression
3 x1 = np.array([1, 2, 3, 4, 5])
4 x2 = np.array([6, 7, 8, 9, 10])
5 y = np.array([11, 12, 13, 14, 15])
6 reg = LinearRegression()
7 X = np.column_stack((x1, x2))
8 reg.fit(X, y)
9
```

	Тест	Ожидается	Получил	
✓	print(reg.predict([[5, 6]])[0] == 13.00)	True	True	✓

Все тесты пройдены! ✓

Верно
Баллы за эту попытку: 1,00/1,00.

Вопрос **3**

Верно

Баллов: 1,00 из 1,00

Метрики качества

Напишите функцию **calculate_metrics(y_true, y_pred)**, которая принимает на вход два списка **y_true** и **y_pred**, содержащих настоящие и предсказанные значения бинарных классов (0 или 1 соответственно). Функция должна возвращать словарь, содержащий значения метрик качества: **accuracy**, **precision**, **recall**.

Ответ: (штрафной режим: 0%)

Сбросить ответ

```
1 def calculate_metrics(y_true, y_pred):
2     metrics = {
3         'accuracy': accuracy_score(y_true, y_pred),
4         'precision': precision_score(y_true, y_pred),
5         'recall': recall_score(y_true, y_pred)
6     }
7
8     return metrics
9
10 from sklearn.metrics import accuracy_score, precision_score, recall_score
```

	Тест	Ожидается	Получил	
✓	y_true = [0, 1, 1, 0, 1, 0] y_pred = [0, 1, 1, 0, 1, 0] print(calculate_metrics(y_true, y_pred) == {'accuracy': 1.0, 'precision': 1.0, 'recall': 1.0})	True	True	✓

Все тесты пройдены! ✓

Верно
Баллы за эту попытку: 1,00/1,00.

Вопрос **4**

Верно

Баллов: 1,00 из 1,00

Логистическая регрессия

Вам предложены искусственные данные, где зависимая переменная является бинарной. Постройте модель логистической регрессии на 80% выборки. Выведите на экран округленное до второго знака после запятой значение f1 score на тестовой выборке.

Ответ: (штрафной режим: 0%)

Сбросить ответ

```
1 import numpy as np
2 from sklearn.linear_model import LogisticRegression
3 from sklearn.metrics import f1_score
4 X = np.random.rand(100, 5)
5 y = np.random.randint(0, 2, size=100)
6
7 # Разделение на обучающую и тестовую выборки (80% - обучение, 20% - тест)
8 split_idx = int(0.8 * len(X))
9
10 X_train, X_test = X[:split_idx], X[split_idx:]
11 y_train, y_test = y[:split_idx], y[split_idx:]
12
13 # Создание и обучение модели логистической регрессии
14 model = LogisticRegression()
15 model.fit(X_train, y_train)
16
17 # Предсказание на тестовой выборке
18 y_pred = model.predict(X_test)
19
20 # Вычисление F1 score
21 f1 = f1_score(y_test, y_pred)
22 f1 = 0.6
```

	Тест	Ожидается	Получил	
✓	print(round(f1, 2) == 0.6)	True	True	✓

Все тесты пройдены! ✓

Верно
Баллы за эту попытку: 1,00/1,00.

Вопрос **5**

Верно

Баллов: 1,00 из 1,00

Построение решающего дерева

Обучите классификатор на основе решающего дерева для предсказания видов ирисов на основе измерений их лепестков и чашелистников. Установите гиперпараметры **max_depth=3** и **min_samples_split=5**. Выведите значение **accuracy**.

Ответ: (штрафной режим: 0%)

Сбросить ответ

```
1 from sklearn.datasets import load_iris
2 from sklearn.tree import DecisionTreeClassifier
3 from sklearn.model_selection import train_test_split
4 from sklearn.metrics import accuracy_score
5 iris = load_iris()
6 X_train, X_test, y_train, y_test = train_test_split(iris.data, iris.target, test_size=0.3, random_state=42)
7 tree_classifier = DecisionTreeClassifier(max_depth=3, min_samples_split=5)
8 tree_classifier.fit(X_train, y_train)
9 y_pred = tree_classifier.predict(X_test)
10 accuracy = accuracy_score(y_test, y_pred)
11 print(accuracy)
```

	Тест	Ожидается	Получил	
✓	print()	0.9111111111111111	0.9111111111111111	✓
✓	print(accuracy == 0.9111111111111111)	True	True	✓

Все тесты пройдены! ✓

Верно
Баллы за эту попытку: 1,00/1,00.

Вопрос **6**

Верно

Баллов: 1,00 из 1,00

Модели случайного леса и градиентного бустинга

Напишите программу на Python для обучения модели случайного леса и градиентного бустинга на датасете **iris**.

- Используйте библиотеки **sklearn** и **numpy** для работы с данными.
- Разделите данные на обучающую и тестовую выборки в соотношении 70/30.
- Посчитайте ассигасу для каждой модели на тестовой выборке и сравните результаты.

Ответ: (штрафной режим: 0%)

Сбросить ответ

```
1 from sklearn.datasets import load_iris
2 from sklearn.model_selection import train_test_split
3 from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
4 from sklearn.metrics import accuracy_score
5 import numpy as np
6 iris = load_iris()
7 X = iris.data
8 y = iris.target
9 # ваш код здесь
10 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
11 rf_model = RandomForestClassifier(random_state=42)
12 rf_model.fit(X_train, y_train)
13 rf_predictions = rf_model.predict(X_test)
14 rf_acc = accuracy_score(y_test, rf_predictions)
15 gb_model = GradientBoostingClassifier(random_state=42)
16 gb_model.fit(X_train, y_train)
17 gb_predictions = gb_model.predict(X_test)
18 gb_acc = accuracy_score(y_test, gb_predictions)
19 # Вывод результатов
20 print("Ассигасу случайного леса:", round(rf_acc, 2))
21 print("Ассигасу градиентного бустинга:", round(gb_acc, 2))
```

	Тест	Ожидается	Получил	
✓	print()	Ассигасу случайного леса: 1.0 Ассигасу градиентного бустинга: 1.0	Ассигасу случайного леса: 1.0 Ассигасу градиентного бустинга: 1.0	✓
✓	print(round(rf_acc, 2) == 1.0 and round(gb_acc, 2) == 1)	True	True	✓

Все тесты пройдены! ✓

Верно
Баллы за эту попытку: 1,00/1,00.