

Курс “Python для DataScience”

Практическое задание

Инструкция к сдаче

1. Настоятельно рекомендуем сдавать практическое задание в виде ссылки на личный репозиторий на github.
2. Рекомендуемый способ организации данных в репозитории: создать отдельные папки по темам и помещать в них отдельные файлы для каждой задачи с правильным расширением.

Ссылка на инструкцию по работе с git и сдачу практики:

https://docs.google.com/document/d/1RAT_ukE39iOfbz1xa39QXae2hBUEZ4U6Fko_wFDdrsM/edit

Ссылка на видеокурс по Git:

<https://geekbrains.ru/courses/66>

Если остались сложности с системой git, то обратитесь к преподавателю или наставнику.

Тема “Обучение с учителем”

Задание 1

Импортируйте библиотеки pandas и numpy.

Загрузите "Boston House Prices dataset" из встроенных наборов данных библиотеки sklearn. Создайте датафреймы X и y из этих данных.

Разбейте эти датафреймы на тренировочные (X_train, y_train) и тестовые (X_test, y_test) с помощью функции train_test_split так, чтобы размер тестовой выборки

составлял 30% от всех данных, при этом аргумент random_state должен быть равен 42.

Создайте модель линейной регрессии под названием lr с помощью класса LinearRegression из модуля sklearn.linear_model.

Обучите модель на тренировочных данных (используйте все признаки) и сделайте предсказание на тестовых.

Вычислите R2 полученных предсказаний с помощью r2_score из модуля sklearn.metrics.

Задание 2

Создайте модель под названием `model` с помощью `RandomForestRegressor` из модуля `sklearn.ensemble`.

Сделайте аргумент `n_estimators` равным 1000,

`max_depth` должен быть равен 12 и `random_state` сделайте равным 42.

Обучите модель на тренировочных данных аналогично тому, как вы обучали модель `LinearRegression`,

но при этом в метод `fit` вместо датафрейма `y_train` поставьте `y_train.values[:, 0]`,

чтобы получить из датафрейма одномерный массив Numpy,

так как для класса `RandomForestRegressor` в данном методе для аргумента `y` предпочтительно применение массивов вместо датафрейма.

Сделайте предсказание на тестовых данных и посчитайте R^2 . Сравните с результатом из предыдущего задания.

Напишите в комментариях к коду, какая модель в данном случае работает лучше.

*Задание 3

Вызовите документацию для класса `RandomForestRegressor`,

найдите информацию об атрибуте `feature_importances_`.

С помощью этого атрибута найдите сумму всех показателей важности,

установите, какие два признака показывают наибольшую важность.

*Задание 4

В этом задании мы будем работать с датасетом, с которым мы уже знакомы по домашнему заданию по библиотеке `Matplotlib`, это датасет `Credit Card Fraud Detection`. Для этого датасета мы будем решать задачу классификации - будем определять, какие из транзакции по кредитной карте являются мошенническими. Данный датасет сильно несбалансирован (так как случаи мошенничества относительно редки), так что применение метрики `ассигасу` не принесет пользы и не поможет выбрать лучшую модель. Мы будем вычислять `AUC`, то есть площадь под кривой `ROC`.

Импортируйте из соответствующих модулей `RandomForestClassifier`, `GridSearchCV` и `train_test_split`.

Загрузите датасет `creditcard.csv` и создайте датафрейм `df`.

С помощью метода `value_counts` с аргументом `normalize=True` убедитесь в том, что выборка несбалансирована. Используя метод `info`, проверьте, все ли столбцы содержат числовые данные и нет ли в них пропусков. Примените следующую настройку, чтобы можно было просматривать все столбцы датафрейма:

```
pd.options.display.max_columns = 100.
```

Просмотрите первые 10 строк датафрейма df.

Создайте датафрейм X из датафрейма df, исключив столбец Class.

Создайте объект Series под названием y из столбца Class.

Разбейте X и y на тренировочный и тестовый наборы данных при помощи функции train_test_split, используя аргументы: test_size=0.3, random_state=100, stratify=y.

У вас должны получиться объекты X_train, X_test, y_train и y_test.

Просмотрите информацию о их форме.

Для поиска по сетке параметров задайте такие параметры:

```
parameters = [{'n_estimators': [10, 15],  
'max_features': np.arange(3, 5),  
'max_depth': np.arange(4, 7)}]
```

Создайте модель GridSearchCV со следующими аргументами:

```
estimator=RandomForestClassifier(random_state=100),  
param_grid=parameters,  
scoring='roc_auc',  
cv=3.
```

Обучите модель на тренировочном наборе данных (может занять несколько минут).

Просмотрите параметры лучшей модели с помощью атрибута best_params_.

Предскажите вероятности классов с помощью полученной модели и метода predict_proba.

Из полученного результата (массив Numpy) выберите столбец с индексом 1 (вероятность класса 1) и запишите в массив y_pred_proba. Из модуля sklearn.metrics импортируйте метрику roc_auc_score.

Вычислите AUC на тестовых данных и сравните с результатом, полученным на тренировочных данных, используя в качестве аргументов массивы y_test и y_pred_proba.

***Дополнительные задания:**

1). Загрузите датасет Wine из встроенных датасетов sklearn.datasets с помощью функции load_wine в переменную data.

2). Полученный датасет не является датафреймом. Это структура данных, имеющая ключи аналогично словарю. Просмотрите тип данных этой структуры данных и создайте список data_keys, содержащий ее ключи.

3). Просмотрите данные, описание и названия признаков в датасете. Описание нужно вывести в виде привычного, аккуратно оформленного текста, без обозначений переноса строки, но с самими переносами и т.д. 4). Сколько классов содержит целевая переменная датасета? Выве

дите названия классов.

5). На основе данных датасета (они содержатся в двумерном массиве `Numpy`) и названий признаков создайте датафрейм под названием `X`.

6). Выясните размер датафрейма `X` и установите, имеются ли в нем пропущенные значения.

7). Добавьте в датафрейм поле с классами вин в виде чисел, имеющих тип данных `numpy.int64`. Название поля - `'target'`.

8). Постройте матрицу корреляций для всех полей `X`. Дайте полученному датафрейму название `X_corr`.

9). Создайте список `high_corr` из признаков, корреляция которых с полем `target` по абсолютному значению превышает 0.5 (причем, само поле `target` не должно входить в этот список).

10). Удалите из датафрейма `X` поле с целевой переменной. Для всех признаков, названия которых содержатся в списке `high_corr`, вычислите квадрат их значений и добавьте в датафрейм `X` соответствующие поля с суффиксом `'_2'`, добавленного к первоначальному названию признака. Итоговый датафрейм должен содержать все поля, которые, были в нем изначально, а также поля с признаками из списка `high_corr`, возведенными в квадрат. Выведите описание полей датафрейма `X` с помощью метода `describe`.