

Пожидаев Антон Николаевич ИТУ (маг.
1к.) Лабораторная работа №3
Вариант - 13

Задание

Решить задачу предсказания курса акций.

Данные были взяты с investing.com

Код программы

```
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
import pandas as pd
import matplotlib.pyplot as plt
plt %matplotlib inline

training_data =
pd.read_excel('/content/drive/MyDrive/deep_learning/sber_training_data.
xlsx')
training_data.head()
training_data.shape
training_data.info()
training_values = training_data['Цена']
training_values.head()
training_points = training_data.drop('Цена', axis=1)
training_points.head()
training_points.shape

from sklearn import linear_model, ensemble
linear_regression_model = linear_model.LinearRegression()
random_forest_model = ensemble.RandomForestRegressor()

linear_regression_model.fit(training_points, training_values)
random_forest_model.fit(training_points, training_values)

test_data =
pd.read_excel('/content/drive/MyDrive/deep_learning/sber_test_data.xlsx
')
test_data.head()
test_data.shape

test_data.info()

test_values = test_data['Цена']
```

```

test_points = test_data.drop('Цена', axis=1)
test_points.head()
test_points.shape
test_predictions_linear = linear_regression_model.predict(test_points)
test_predictions_random_forest =
random_forest_model.predict(test_points)
%matplotlib inline
plt.figure(figsize=(7, 7))
plt.scatter(test_values, test_predictions_linear) # рисуем точки,
соответствующие парам настоящее значение - прогноз
plt.plot([0, 3 * 10**2], [0, 3 * 10**2]) # рисуем прямую, на которой
предсказания и настоящие значения совпадают
plt.xlabel('Настоящая цена', fontsize=20)
plt.ylabel('Предсказанная цена', fontsize=20);
plt.figure(figsize=(7, 7))
plt.scatter(test_values, test_predictions_random_forest)
plt.plot([0, 3 * 10**2], [0, 3 * 10**2])
plt.xlabel('Настоящая цена', fontsize=20)
plt.ylabel('Предсказанная цена', fontsize=20);

from sklearn.metrics import mean_absolute_error, mean_squared_error

mean_absolute_error_linear_model = mean_absolute_error(test_values,
test_predictions_linear)
mean_squared_error_linear_model = mean_squared_error(test_values,
test_predictions_linear)

mean_absolute_error_random_forest_model =
mean_absolute_error(test_values, test_predictions_random_forest)
mean_squared_error_random_forest_model =
mean_squared_error(test_values, test_predictions_random_forest)

import numpy as np

print("MAE: {0:7.2f}, RMSE: {1:7.2f} для модели линейной
регрессии".format(
    mean_absolute_error_linear_model,
    np.sqrt(mean_squared_error_linear_model)))

print("MAE: {0:7.2f}, RMSE: {1:7.2f} для модели случайного
леса".format(
    mean_absolute_error_random_forest_model,
    mean_squared_error_random_forest_model**(1/2)))

```

```

random_forest_model.feature_importances_
feature_importance = pd.DataFrame(columns = ['Название признака',
'Важность признака'])
feature_importance['Название признака'] = training_points.keys()
feature_importance['Важность признака'] =
random_forest_model.feature_importances_
feature_importance
feature_importance.sort_values(by='Важность признака', ascending=False)

```

Результаты:

MAE: 0.97, RMSE: 1.50 для модели линейной регрессии
MAE: 0.46, RMSE: 0.75 для модели случайного леса

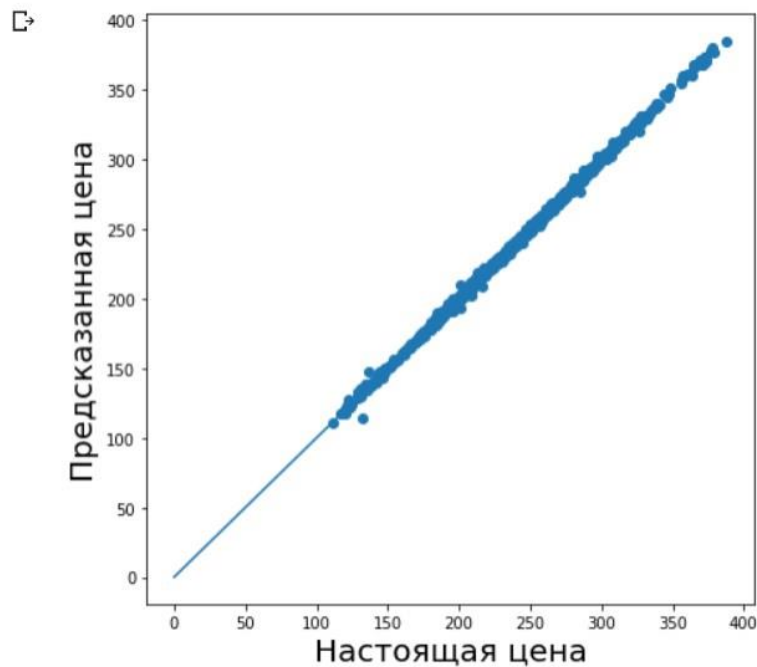


Рис.1. - Прогноз линейной модели

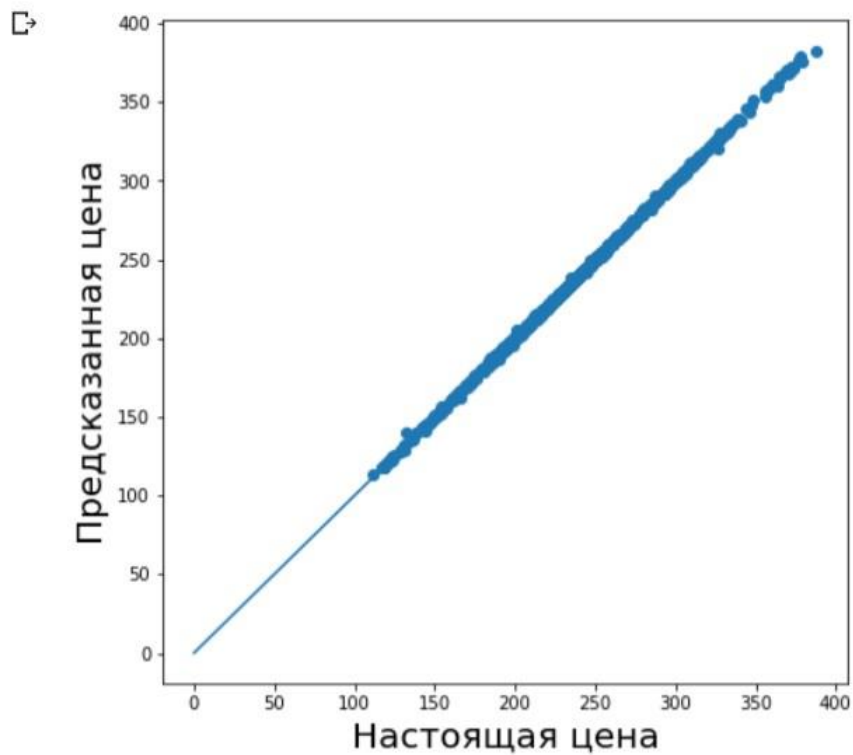


Рис. 2. - Прогноз модели случайного леса

Вывод

По метрикам видно, что модель случайного леса работает лучше. Т.к. абсолютная и среднеквадратичная ошибка меньше для линейной регрессии.

[colab](#)