Дунаева Алина Ивановна
Магистр
Московского государственного университета
имени М. В. Ломоносова
Москва, Россия
Истранин Артем Вадимович
Бакалавр
Санкт-Петербургского государственного экономического университета
Санкт-Петербург, Россия

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ ПРИ ПРОГНОЗЕ ДИНАМИКИ ИЗМЕНЕНИЯ ФИНАНСОВЫХ ВРЕМЕННЫХ РЯДОВ

Аннотация. Модели машинного обучения являются одним из основных инструментов при осуществлении анализа данных и проведения прогноза их динамики. В данной работе с помощью сравнительного анализа и применения методов машинного обучения исследуются различия в значение точности прогноза движения цены финансового актива в зависимости от выбранной модели.

Ключевые слова: методы машинного обучения, нейронные сети, метод опорных векторов, прогноз динамики финансовых временных рядов, boosting, MLP, neural networks, support vector machine.

Машинное обучение представляет собой подраздел искусственного интеллекта, изучающий методы построения самообучающихся алгоритмов. Многие методы машинного обучения изначально формировались в качестве альтернативы уже существующим статистическим методам. Применение на практике большой части этих подходов происходит с использованием дополнительных эвристик с целью сглаживания несоответствия в теоретических предположениях и реальных условиях.

В данной статье будут рассматриваться такие методы машинного обучения, как: метод опорных векторов, бустинг и нейронные сети (модель многослойного перцептрона (MLP – multilayer perceptron)).

Нейронные сети (MLP – multilayer perceptron)

В силу того, что данный метод является наиболее точным в предсказании, то в данной статье он будет рассмотрен наиболее подробно.

Перед подачей данных на обучение необходимо их тщательно обработать: применить нормализацию данных, удалить избыточную информацию, а также какие-либо аномальные значения и прочие шумы.

В случае с нейронными сетями также необходимо принимать во внимание, что на вход подаются определенные батчи (англ. – «batch», партии/серии) входных сигналов, которые также в свою очередь должны быть масштабированы, исходя из ранга персептрона активационной функции, для того, чтобы в дальнейшем сеть могла различать входные паттерны [2]. Это предпосылка с нормализации данных.

В данной статье в качестве примера с Google Finance были выгружены исторические котировки компании Tesla Inc. (тикер TSLA). Прогноз осуществлялся по значениям Close price. Ниже приведен код выгрузки данных на Python 3.6 в среде Jupyter Notebook.

```
import numpy as np
import pandas as pd
import pandas_datareader.data as web
import datetime as dt
from matplotlib import pyplot as plt
# Retrieve TSLA stock prices from Google finance
# from 01/01/2007 to 01/01/2017
start_time = dt.datetime(2007, 1, 1)
end_time = dt.datetime(2017, 1, 1)
df = web.DataReader('TSLA', 'google', start_time, end_time)['Close']
df = df.rename(columns={'Close': 'TSLA'})
```

```
plt.plot(df)
plt.show()
```

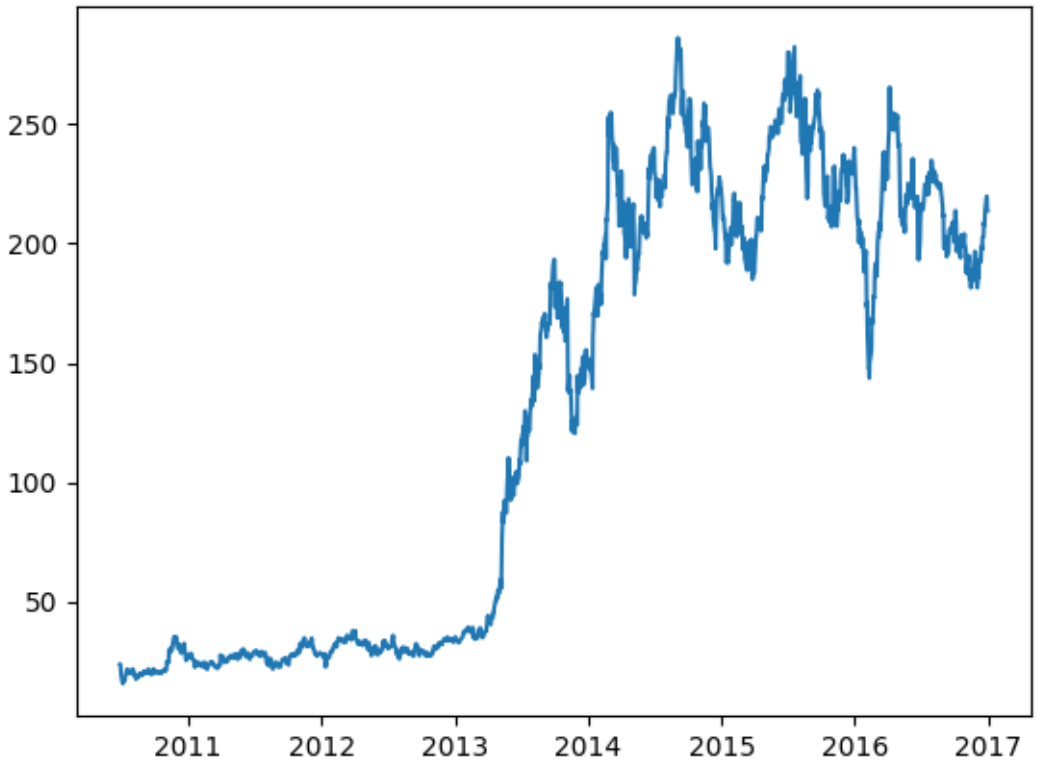

Рисунок 1. Динамика цен акций Tesla Inc. (ticker – TSLA)

Нормализация данных необходима, так как некоторые алгоритмы машинного обучения, такие как метод опорных векторов, сильно зависят от масштабирования данных. Более того, необходимо обратить внимание на какие-либо аномалии в данных (либо слишком высокие, либо слишком низкие значения), так как наличие таких «всплесков» может привести к неточности применения статистических методов, в силу того, что система будет стараться «подстроить» эти значения под общую тенденцию, что существенно снизит качество ожидаемого результата [1, 3].

Применение нормализации приводит данные к виду, указанному на Рисунке 2. Это данные, полученные без каких-то манипуляций со статистическими характеристиками, однако они уже лежат в пределе приблизительно от -0.2 до 0.2:

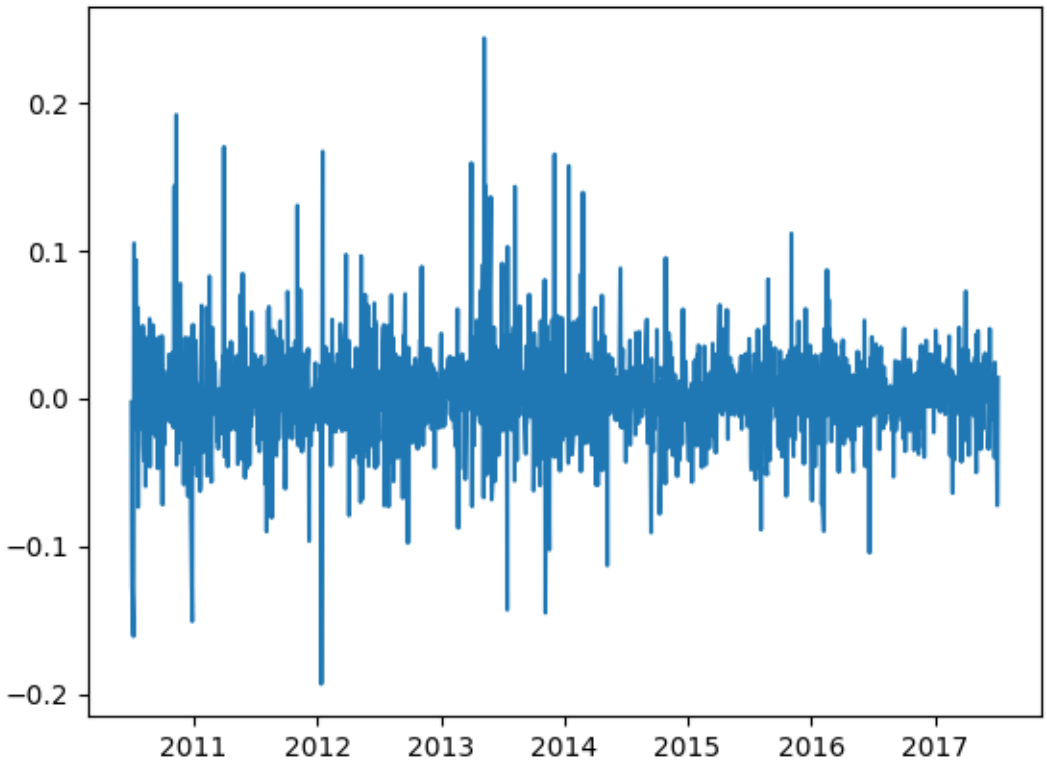

Рисунок 2. Нормированные данные

```
# Normalization (30 days window) with z-score without future values.
df = df.dropna(axis=0, how='any')
close_price_diffs = df.price.pct_change()
plt.plot(close_price_diffs)
plt.show()
```

Также, наличие в датасете (англ. – «dataset») двух или более переменных, сильно коррелирующих друг с другом, тоже могут привести к ухудшению продуктивности алгоритма и снижению способности к обучению. Поэтому необходимо оставить только те переменные, которые будут нести максимальную информативность за наименьшим их числом [4]. Для этого используется метод главных компонент (Principal Component Analysis - PCA).

После того, как данные очищены и подготовлены для анализа можно применять методы обучения. Данные были разделены на тренировочную и тестовую выборки в соотношении 85% и 15% соответственно, с помощью Python пакета sklearn:

```
from sklearn.model_selection import train_test_split
```

Таким образом, получились пары X, Y, которые представляют из себя цены закрытия за 30 дней и [1, 0] или [0, 1] в зависимости от того, выросло или упало значение цены для бинарной классификации; процентное изменение цен за 30 дней и изменение на следующий день для регрессии. Далее осуществлялось построение нейронной сети, с использованием Keras:

```
from keras.models import Sequential
from keras.layers import Dense, Activation, Dropout
from keras.layers import BatchNormalization
from keras.layers.advanced_activations import LeakyReLU
from keras.callbacks import ReduceLROnPlateau
from keras import regularizers
```

Далее проводилось обучение сети и ее оптимизация с помощью метода Adam, параметр loss – 'categorical_crossentropy', средняя квадратическая ошибка – 'mse' и для избежания переобучения применялась L2-регуляризация. Выбор гиперпараметров, таких как: размер окна, количество нейронов в скрытых слоях, шаг обучения проводился методом подбора. Для более точного предсказания применяется метод случайного поиска (RandomizedSearchCV).

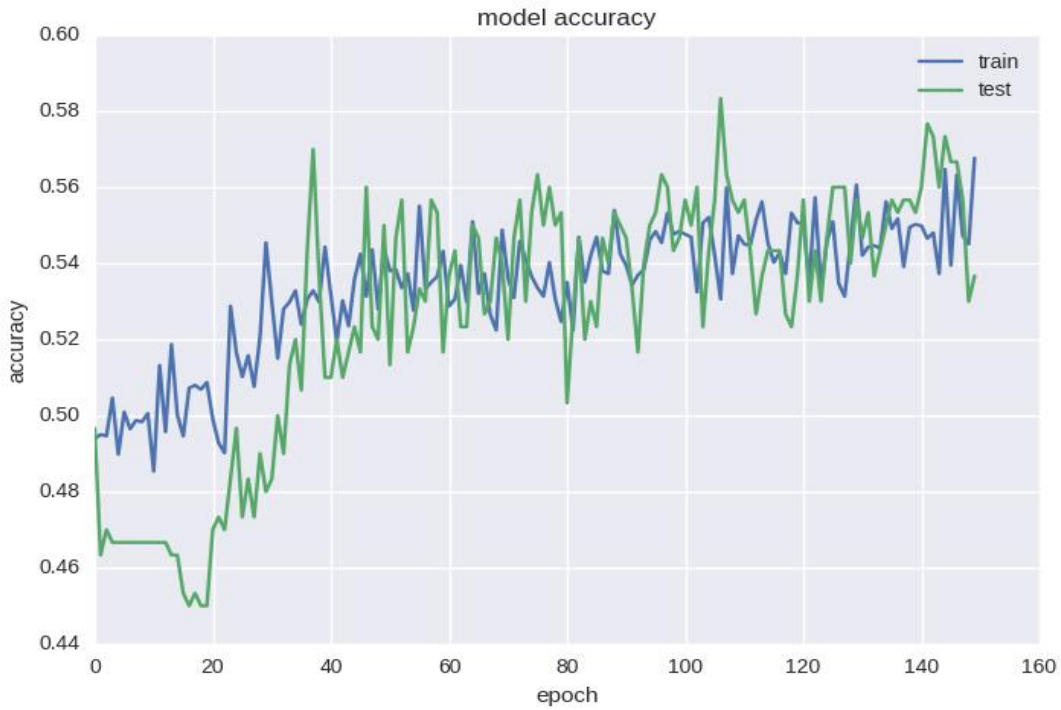

Рисунок 3. Обученная нейронная сеть с применением L2- регуляризации

На данном этапе задачи классификации (движение цены в следующий день) достигнута точность предсказания ~55%. Решение задачи регрессии (значение изменения цены) дает следующий результат при обучении на изменениях цен:

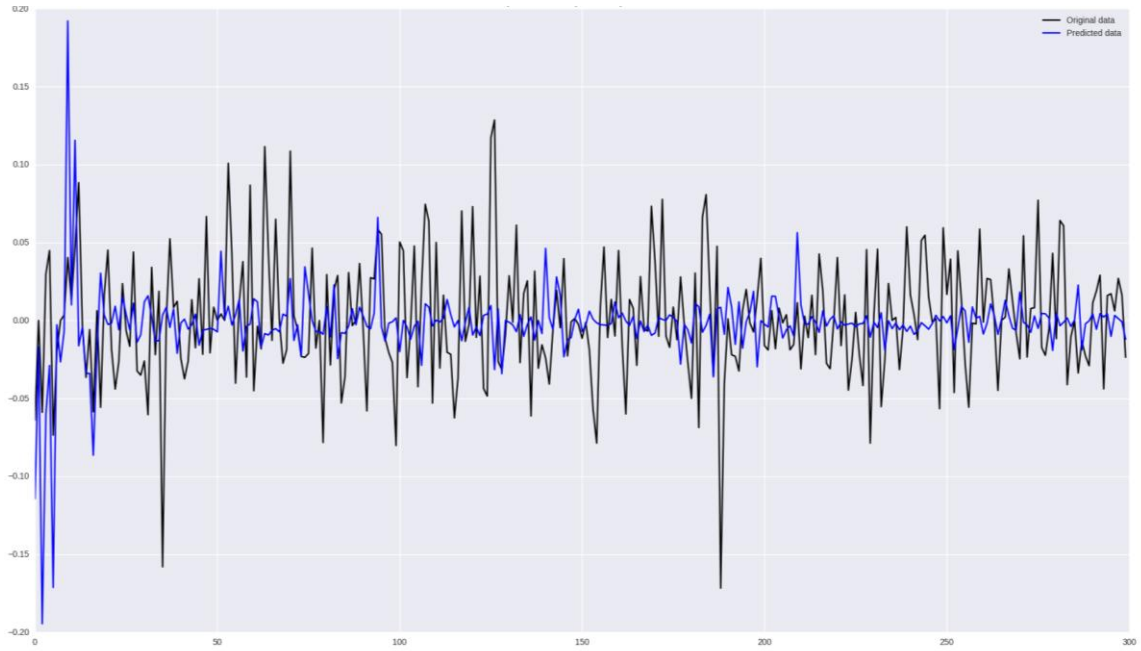

Рисунок 4. Результат обучения на изменениях цен

Таким образом, данный метод машинного обучения позволил добиться точности прогноза динамики изменения цен акций в ~63%.

Метод опорных векторов

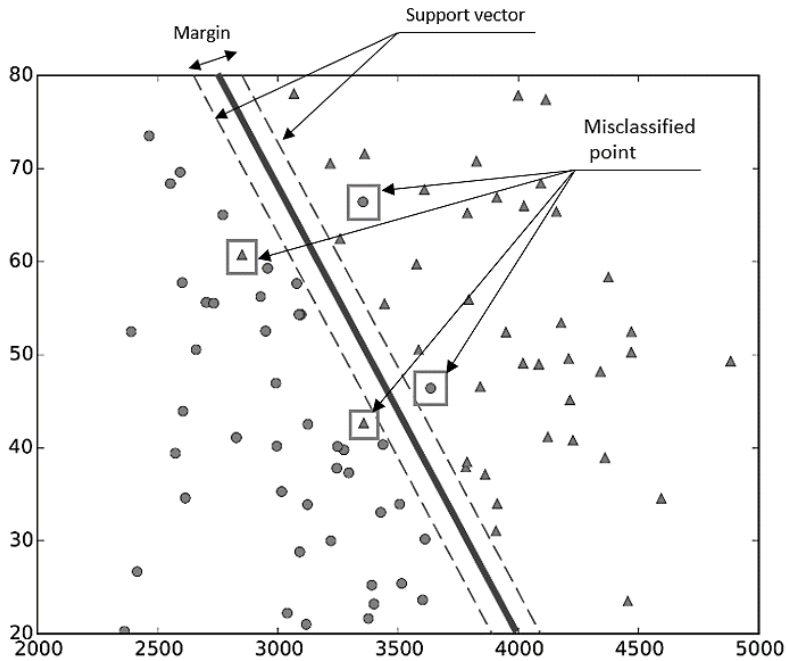

Рисунок 5. «Общий вид метода опорных векторов»

В результате применения данного метода получились следующие значения:

Корень среднеквадратичной ошибки (RMSE)	0.486 ± 0.012
Точность	60.20 ± 0.49%

Таблица 1. Результаты применения метода опорных векторов

Бустинг (англ. – «Boosting»)

Данный метод представляет из себя процесс последовательного построения совокупности алгоритмов машинного обучения так, что каждый последующий алгоритм стремится компенсировать недостатки предыдущего. Бустинг представляет собой «жадный» алгоритм и является одним из наиболее популярных методов машинного обучения, а также одним из наиболее точных в сравнении с нейронными сетями и методом опорных векторов [1, 5].

После использования алгоритма C-SVC к набору данных был применен алгоритм бустинга AdaBoostM1 — это позволило добиться серьезного улучшения точности.

Корень среднеквадратичной ошибки	0.467 ± 0.008
Точность	64.3% ± 3.9%

Таблица 2. Результаты применения алгоритма бустинга

Таким образом, на основе сделанного анализа можно вывести следующую сравнительную таблицу методов машинного обучения.

Метод	Точность	Корень среднеквадратичной ошибки
Бустинг	64.3% ± 3.9%	0.467 ± 0.008
Метод опорных векторов	60.2% ± 0.5%	0.486 ± 0.012
Нейронная сеть	65.0 ± 4.5%	

Таблица 3. Сравнительная таблица результатов применения методов машинного обучения

Из таблицы видно, что наиболее эффективным метод является нейронная сеть, однако точность в 60-70% может быть увеличена на ~5-10% за счет выполнения следующих шагов:

- Проводить обучение нейронной сети на более коротких временных промежутках, так как с увеличением количества данных увеличивается и точность предсказания, а также снижается вероятность переобучения. Следовательно, целесообразней брать промежутки в 5 мин или 15 мин;
- Внедрить в модель помимо цены закрытия цены открытия, наибольшее и наименьшее значения и объем;
- Использовать более подходящие функции потерь (а не просто MSE), предполагающие возможность штрафования сети за неправильные ответы;
- Использовать более динамичные модели, например, рекуррентные нейронные сети, которые способны самостоятельно адаптироваться под динамику рынка, удаляя или прибавляя в своей структуре число нейронов.

Построение моделей машинного обучения является одним из основных инструментов при осуществлении анализа данных и проведения прогноза их динамики. В данной работе с помощью сравнительного анализа и применения методов машинного обучения исследуются различия в точности прогноза движения цены финансового актива в зависимости от выбранной модели.

Литература

1. http://www.machinelearning.ru/wiki/images/9/9a/fonarev.overview_of_boosting_methods.pdf
2. http://mechanoid.kiev.ua/neural-net-backprop2.html
3. Курс лекций К.В.Воронцова по теме «Машинное обучение» на портале http://www.machinelearning.ru/
4. В.В.Вьюгин Математические основы теории машинного обучения и прогнозирования / Москва – 2013г. – С387.
5. Ю. С. Кашницкий, Д. И. Игнатов Ансамблевый метод машинного обучения, основанный на рекомендации классификаторов // Machine Learning in Python, Journal of Machine Learning Research, 12. – 2825-2830 (2011)