БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

На правах рукописи
УДК 519.178

Песецкий Артем Степанович

Проблемные вопросы применения IT для решения задачи прогнозирования

Реферат по
«Основам информационных технолгий»

Магистранта кафедры дискретной математики и алгоритмики факультета прикладной математики и информатики

Специальность: 1-31 80 09 — прикладная математика и информатика

Рецензент:
Мамай Дарья Сергеевна

Минск, 2018

ОГЛАВЛЕНИЕ

Введение

В последние несколько лет все больше разнообразных повседневных задач решаются с помощью алгоритмов машинного обучения. Машинное обучение (англ. Machine Learning, ML) — класс методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение в процессе применения решений множества сходных задач. Для построения таких методов используются средства математической статистики, численных методов, методов оптимизации, теории вероятностей, теории графов, различные техники работы с данными в цифровой форме. Теоретические основы машинного обучения были заложены еще в середине прошлого века, однако в течение долгого времени эта область компьютерной науки практически не развивалась, так для использования алгоритмов машинного обучения необходимы были высокопроизводительные вычислительные системы. С достижением необходимой производительности в начале 2010-х алгоритмы машинного обучения стали все чаще использоваться в решении разнообразных практических задач, таких как распознавание речи, жестов и образов; техническая и медицинская диагностика, обнаружение спама и другие. Одной из областей машинного обучения являются алгоритмы прогнозирования. Приведем несколько примеров решенных задач прогнозирования и алгоритмов, применяемых для этого:

1. Прогнозирование финансовых процессов и биржевых индексов. В частности, Дегтярев В.М использовал многослойные нейронные сети для прогнозирования поведения валютной пары доллар США/ швейцарский франк и в результате построил модель для торговли на бирже, прибыль работы с при помощи которой составляла порядка 7 процентов [1]. Также можно выделить работу Samuel Edet, который использовал рекуррентные нейронные сети для прогнозирования изменений значения индеков SA SnP 500 [2]. Точность полученной им модели составляла порядка 75 процентов.

2. Медицинская диагностика. Примером работ в этой области можно привести можно привести работу сотрудников университета Стэнфорда, которые использовали сверточную нейронную сеть для прогнозирования возможной аритмии у пациентов по данным кардиограммы. [3]

3. Спортивное прогнозирование. Алгоритмы машинного обучения использовались для прогнозирования результатов матчей в самых разных видах спорта. В частности, Kahn использовал многослойную нейронную сеть для построения модели классификации для прогнозирования результатов матчей NFL в 2003 году. В качестве обучающей выборки были

использованы первые 192 матча сезона. В результате результаты прогноза модели превзошли результаты предсказаний профессиональных экспертов. [4]

В качестве задачи прогнозирования была выбрана задача теннисного прогнозирования.

Основные определения и понятия

Машинное обучение (англ. machine learning, ML) — класс методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение в процессе применения решений множества сходных задач. Для построения таких методов используются средства математической статистики, численных методов, методов оптимизации, теории вероятностей, теории графов, различные техники работы с данными в цифровой форме.

Теннис — это игра с ракеткой, в которую могут играть как один на один, так и парами. Для простоты сфокусируемся на предсказании результатов одиночных матчей. Дадим определение задачи теннисного прогнозирования.Пусть имеется следующая информация о предстоящем теннисном матче: имена участников, тип покрытия и позиции игроков в рейтинге ATP; а также историческая информация об уже сыгранных матчах. Необходимо построить модель на основе исторических данных, которая будет наилучшим образом прогнозировать результаты предстоящих матчей. Выбор данной задачи был сделан по нескольким причинам. Во-первых, теннис является одним из самых популярных видов спорта, а рынок ставок на теннис является одним из крупнейших. Во-вторых, на данный момент большинство средств для решения данной задачи используют в своей основе статистические модели, а алгоритмы машинного обучения для этой задачи практически не используются.

Обучающая выборка - набор данных для построения, обучения и анализа алгоритма машинного обучения. В случае задачи теннисного прогнозирования данная выборка представлена ранее сыгранными теннисными матчами и информацией о них.

Глава 1
Инструменты для обработки и анализа данных

1.1 Общая схема решения задачи прогнозирования

Приведем шаги для решения задачи прогнозирования:

1. Поиск и фильтрация данных, построение обучающей выборки;
2. Отбор наиболее релевантных признаков для построения обучающей выборки;
3. Построение и обучение модели;
4. Анализ полученных результатов.

В качестве языка программирования был использован Python. Python — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций. Python поддерживает несколько парадигм программирования, в том числе структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное. Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений и удобные высокоуровневые структуры данных. Код в Python организовывается в функции и классы, которые могут объединяться в модули (они в свою очередь могут быть объединены в пакеты).

Выбор в пользу данного языка был обусловлен наличием на нем большого количества библиотек, предназначенных для разработки и обучения алгоритмов машинного обучения. Важным преимуществом этих библиотек является наличие имплементаций низкоуровневых операций на языке C, что благоприятно влияет на производительность.

Проведем обзор иструментов и библиотек, которые были использованы для решения поставленных выше задач.

1.2 Построение обучающей выборки

Для построения обучающей выборки из источника 1 была загружена информация об мужских одиночных теннисных матчах, сыгранных под эгидой ATP, за 2004-2018 годы. Для фильтрации данных и построения выборки использовался фреймворк Pandas.

Pandas – библиотека языка Python, предназначенная для обработки и анализа данных. Работа библиотеки построена поверх библиотеки NumPy. Библиотекой предоставляются разнообразные структуры и алгоритмы числовыми данными и временными рядами. Дадим краткое описание возможностей библиотеки:

- объект DataFrame, позволяющий манипулировать индексированными двумерными данными;
- возможность обмена файлами разнообразных типов, а также между структурами в оперативной памяти;
- наличие инструментов объединения данных, а также возможность обработки недостающей информации;
- переформатирование информации, в том числе создание сводных таблиц данных;
- наличие возможности получения среза данных по индексу, возможность получения выборки из больших объемов информации. [5]

1.3 Отбор признаков. Построение и обучение алгоритмов.

После построения набора признаков для каждого из матча полученной обучающей выборки из него с помощью были выбраны наиболее релевантные признаки. Для этого реализация алгоритма RFE(recursive feature elimination) из библиотеки Scikit-learn.

Scikit-learn – библиотека для машинного обучения, написанная на языке Python. В нее включены разнообразные алгоритмы классификации, регрессии и кластеризации, такие как случайный лес, метод опорных векторов, k-means, DBSCAN и другие. Разработка библиотеки началась Дэвидом Корнапеу в рамках проекта Google Summer of Code. Позже оригинальная кодовая база была полностью переписана другими разработчиками. Релиз библиотеки

состоялся в феврале 2010 года после того, как в проект включились студенты французского государственного института исследований в информатике и автоматике INRIA. На данный момент данная библиотека завоевала популярность, и разработка обновлений продолжается. Основная часть кодовой базы написана на Python, часть алгоритмов написана на Cython для достижения большей производительности. Поддержка метода нелинейных опорных векторов реализована с помощью обертки на Cython вокруг библиотеки LIBSVM, а поддержка линейной регрессии и линейного метода опорных векторов с помощью обертки вокруг библиотеки LIBLINEAR. В данной работе эта библиотека использовалась для построения моделей логистической регрессии и метода опорных векторов, а также для отбора наиболее релевантных признаков с помощью алгоритма RFE.

После получения набора наиболее релевантных признаков с их помощью были построены 3 модели на основании следующих алгоритмов:

- логистическая регрессия - это статистическая модель, используемая для предсказания вероятности возникновения некоторого события путём подгонки данных к логистической кривой?;
- искусственная нейронная сеть (ИНС) — математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма. Это понятие возникло при изучении процессов, протекающих в мозге, и при попытке смоделировать эти процессы;
- метод опорных векторов (англ. SVM, support vector machine) — набор схожих алгоритмов обучения с учителем, использующихся для задач классификации и регрессионного анализа. Принадлежит семейству линейных классификаторов и может также рассматриваться как специальный случай регуляризации по Тихонову. Особым свойством метода опорных векторов является непрерывное уменьшение эмпирической ошибки классификации и увеличение зазора, поэтому метод также известен как метод классификатора с максимальным зазором.

Для построения указанных алгоритмов использовалась выше описанная библиотека Scikit-learn, а также библиотека Keras.

Keras – открытая библиотека на языке Python, предназначенная для построения нейросетевых моделей. Представляет собой интерфейс, надстроенный над Deeplearning4j, TensorFlow и Theano. Основным предназначением библиотеки является работа с нейронными сетями глубокого обучения. Библиотека было представлена в рамках проекта ONEIROS. Основным автором

библиотеки является инженер Google Франсуа Шолле. Keras содержит в себе все основные реализации широко применяемых при создании нейронных сетей элементов, такие как слои, целевые и передаточные функции, оптимизаторы, а также набор инструментов для упрощения работы с картинками и текстом. Исходный код библиотеки размещен в открытом репозитории на Github. В работе данная модель использовалась для построения и обучения нейронной сети.

Для демострации работы моделей необходимо построить программу с графическим интерфейсом. Для его построения была использована библиотека Qt. Qt — кроссплатформенный фреймворк для разработки программного обеспечения на языке программирования C++. Есть также «привязки» ко многим другим языкам программирования: Python — PyQt, PySide; Ruby — QtRuby; Java — Qt Jambi; PHP — PHP-Qt и другие.

Со времени своего появления в 1996 году библиотека легла в основу многих программных проектов. Кроме того, Qt является фундаментом популярной рабочей среды KDE, входящей в состав многих дистрибутивов Linux.

Qt позволяет запускать написанное с его помощью программное обеспечение в большинстве современных операционных систем путём простой компиляции программы для каждой системы без изменения исходного кода. Включает в себя все основные классы, которые могут потребоваться при разработке прикладного программного обеспечения, начиная от элементов графического интерфейса и заканчивая классами для работы с сетью, базами данных и XML. Является полностью объектно-ориентированным, расширяемым и поддерживающим технику компонентного программирования.

Отличительная особенность — использование метаобъектного компилятора — предварительной системы обработки исходного кода. Расширение возможностей обеспечивается системой плагинов, которые возможно размещать непосредственно в панели визуального редактора. Также существует возможность расширения привычной функциональности виджетов, связанной с размещением их на экране, отображением, перерисовкой при изменении размеров окна.

1.4 Выводы

Средства языка Python, а также дополнительные библиотеки такие как Pandas, Scikit-lear, Keras и QT позволяют эффективно строить модели на основе алгоритмов машинного обучения для решения разнообразных задач прогнозировани, а также графический интерфейс для демонстрации работы полученных моделей.

Глава 2

Средства для написания кода и написания работы

2.1 Средства для написания кода

Самыми популярными средами написания кода на языке программирования Python являются следующие IDE(англ. Integrated Development Environment - интегрированння среда разработки): IntelliJ IDEA PyCharm, Jupiter Notebook расширение для IPython и Spyder.

PyCharm — интегрированная среда разработки для языка программирования Python. Предоставляет средства для анализа кода, графический отладчик, инструмент для запуска юнит-тестов и поддерживает веб-разработку на Django. PyCharm разработана компанией JetBrains[5] на основе IntelliJ IDEA. PyCharm работает под операционными системами Windows, Mac OS X и Linux. PyCharm Professional Edition имеет несколько вариантов лицензий, которые отличаются функциональностью, стоимостью и условиями использования.PyCharm Professional Edition является бесплатным для образовательных учреждений и проектов с открытым исходным кодом.Существует также бесплатная версия Community Edition, обладающая усеченным набором возможностей. Распространяется под лицензией Apache 2.

Ключевые возможности PyCharm:

- статический анализ кода, подсветка синтаксиса и ошибок.
- навигация по проекту и исходному коду: отображение файловой структуры проекта, быстрый переход между файлами, классами, методами и использованиями методов.
- рефакторинг: переименование, извлечение метода, введение переменной, введение константы, подъём и спуск метода и т. д.
- инструменты для веб-разработки с использованием фреймворка Django
- встроенный отладчик для Python
- встроенные инструменты для юнит-тестирования
- разработка с использованием Google App Engine
- поддержка систем контроля версий: общий пользовательский интерфейс для Mercurial, Git, Subversion, Perforce и CVS с поддержкой списков изменений и слияния.

Spyder (ранее Pydee) — свободная и кроссплатформенная интерактивная IDE для научных расчетов на языке Python, обеспечивающая простоту использования функциональных возможностей и легковесность программной части. Spyder является частью модуля spyderlib для Python, основанного на PyQt4, pyflakes, rope и Sphinx, предоставляющего мощные виджеты на PyQt4, такие как редактор кода, консоль Python (встраиваемая в приложения), графический редактор переменных (в том числе списков, словарей и массивов).

Ключевые возможности Spyder:

- редактор с подсветкой синтаксиса Python, C/C++ и Fortran
- динамическая интроспекция кода (с помощью rope) — автодополнение, переход к определению объекта по клику мыши
- нахождение ошибок на лету (с использованием pyflakes)
- поддержка одновременного использования множества консолей Python (включая оболочку IPython)
- просмотр и редактирование переменных с помощью GUI (поддерживаются различные типы данных - числа, строки, списки, массивы, словари)
- встроенные средства доступа к документации (в формате Sphinx)
- гибко настраиваемый интерфейс
- интеграция с научными библиотеками Python - NumPy, SciPy, Matplotlib, Pandas.

IPython (англ. Interactive Python) — интерактивная оболочка для языка программирования Python, которая предоставляет расширенную интроспекцию, дополнительный командный синтаксис, подсветку кода и автоматическое дополнение. Является компонентом пакетов программ SciPy и Anaconda. IPython позволяет осуществлять неблокирующее (англ. non-blocking) взаимодействие с Tkinter, GTK, Qt и WX. Стандартная библиотека Python включает лишь Tkinter. IPython может интерактивно управлять параллельными кластерами, используя асинхронные статусы обратных вызовов и/или MPI. IPython может использоваться как замена стандартной командной оболочки операционной системы, особенно на платформе Windows, возможности оболочки которой ограничены. Поведение по умолчанию похоже на поведение оболочек UNIX-подобных систем, но тот факт, что работа происходит в окружении Python, позволяет добиваться большей настраиваемости и гибкости.

Начиная с версии 4.0, монолитный код был разбит на модули, и независимые от языка модули были выделены в отдельный проект Jupyter. Наиболее известной веб-оболочкой для IPython является Jupyter Notebook (ранее известный как IPython Notebook), позволяющая объединить код, текст и диаграммы, и распространять их для других пользователей.

Для хранения исходного кода была использована система контроля версий Git. Git — распределённая система управления версиями. Проект был создан Линусом Торвальдсом для управления разработкой ядра Linux, первая версия выпущена 7 апреля 2005 года. На сегодняшний день его поддерживает Джунио Хамано. Среди проектов, использующих Git — ядро Linux, Swift, Android, Drupal, Cairo, GNU Core Utilities, Mesa, Wine, Chromium, Compiz Fusion, FlightGear, jQuery, PHP, NASM, MediaWiki, DokuWiki, Qt, ряд дистрибутивов Linux.Программа является свободной и выпущена под лицензией GNU GPL версии 2. По умолчанию используется TCP порт 9418.

Система спроектирована как набор программ, специально разработанных с учётом их использования в сценариях. Это позволяет удобно создавать специализированные системы контроля версий на базе Git или пользовательские интерфейсы. Например, Cogito является именно таким примером оболочки к репозиториям Git, а StGit использует Git для управления коллекцией исправлений (патчей).Git поддерживает быстрое разделение и слияние версий, включает инструменты для визуализации и навигации по нелинейной истории разработки. Как и Darcs, BitKeeper, Mercurial, Bazaar и Monotone[en], Git предоставляет каждому разработчику локальную копию всей истории разработки, изменения копируются из одного репозитория в другой.Удалённый доступ к репозиториям Git обеспечивается git-демоном, SSH- или HTTP-сервером. TCP-сервис git-daemon входит в дистрибутив Git и является наряду с SSH наиболее распространённым и надёжным методом доступа. Метод доступа по HTTP, несмотря на ряд ограничений, очень популярен в контролируемых сетях, потому что позволяет использовать существующие конфигурации сетевых фильтров.

2.2 Средства для написания реферата

В качестве системы текстовой верстки был использован Latex. Latex — наиболее популярный набор макрорасширений (или макропакет) системы компьютерной вёрстки TeX, который облегчает набор сложных документов. В типографском наборе системы TeX форматируется традиционно как Latex.

Пакет позволяет автоматизировать многие задачи набора текста и подготовки статей, включая набор текста на нескольких языках, нумерацию разделов и формул, перекрёстные ссылки, размещение иллюстраций и таблиц

на странице, ведение библиографии и др. Кроме базового набора существует множество пакетов расширения Latex. Первая версия была выпущена Лесли Лэмпортом в 1984 году; текущая версия, Latex2e, после создания в 1994 году испытывала некоторый период нестабильности, окончившийся к концу 1990-х годов, а в настоящее время стабилизировалась (хотя раз в год выходит новая версия).

Общий внешний вид документа в LaTeX определяется стилевым файлом. Существует несколько стандартных стилевых файлов для статей, книг, писем и т. д., кроме того, многие издательства и журналы предоставляют свои собственные стилевые файлы, что позволяет быстро оформить публикацию, соответствующую стандартам издания.

Возможности системы, не ограничены, благодаря механизму программирования новых макросов. Вот список некоторых возможностей, предлагаемых стандартными макросами и теми, которые можно скачать с сервера CTAN:

- алгоритмы расстановки переносов, определения междусловных пробелов, балансировки текста в абзацах;
- автоматическая генерация содержания, списка иллюстраций, таблиц и т. д.;
- механизм работы с перекрёстными ссылками на формулы, таблицы, иллюстрации, их номер или страницу;
- механизм цитирования библиографических источников, работы с библиографическими картотеками;
- размещение иллюстраций (иллюстрации, таблицы и подписи к ним автоматически размещаются на странице и нумеруются);
- оформление математических формул, возможность набирать многострочные формулы, большой выбор математических символов;
- оформление химических формул и структурных схем молекул органической и неорганической химии;
- оформление графов, схем, диаграмм, синтаксических графов;
- оформление алгоритмов, исходных текстов программ (которые могут включаться в текст непосредственно из своих файлов) с синтаксической подсветкой;
- разбивка документа на отдельные части (тематические карты).

Расширенные средства работы с библиографическими данными предоставляются программой BibTeX. Базовые возможности работы с математическими формулами расширяются с помощью пакета AMS-LaTeX.

2.3 Выводы

Современные интегрированные среды разработки вкупе с системой контроля версий позволяют эффективно использовать средства языка Python и дополнительных библиотек при разработки моделей алгоритмов для машинного обчуения.

Текствоый процессор Latex позволяет эффективно создавать и форматировать документы.

ЗАКЛЮЧЕНИЕ

В реферате были изучены програмные средства и библиотеки, которые могут применяться для построения моделей машинного обучения, решающих задачу прогнозирования. Использование описанных алгоритмов и программных средств позволило построить модели, решающие задачу теннисного прогнозирования.Точность полученных моделей составила порядка 75 процентов.

СПИСОК ЛИТЕРАТУРЫ

1. В. Дегтярев. Прогнозирование валютных курсов с использованием эконометрических моделей и искусственных нейронных сетей: дис. : Дегтярев В. М – 2012 – 103с.

2. Recurrent Neural Networks in Forecasting SnP 500 Index // Научный портал https://papers.ssrn.com/ [Электронный ресурс]. – 2017. - Режим доступа: https://papers.ssrn.com/sol3/papers.cfm?abstractid=3001046 - Дата доступа: 04.05.2018.

3. Cardiologist-Level Arrhythmia Detection with Convolutional Neural Networks. // Интернет-портал Cournell University Library [Электронный ресурс]. – 2017. - Режим доступа: https://arxiv.org/abs/1707.01836 - Дата доступа: 04.05.2018.

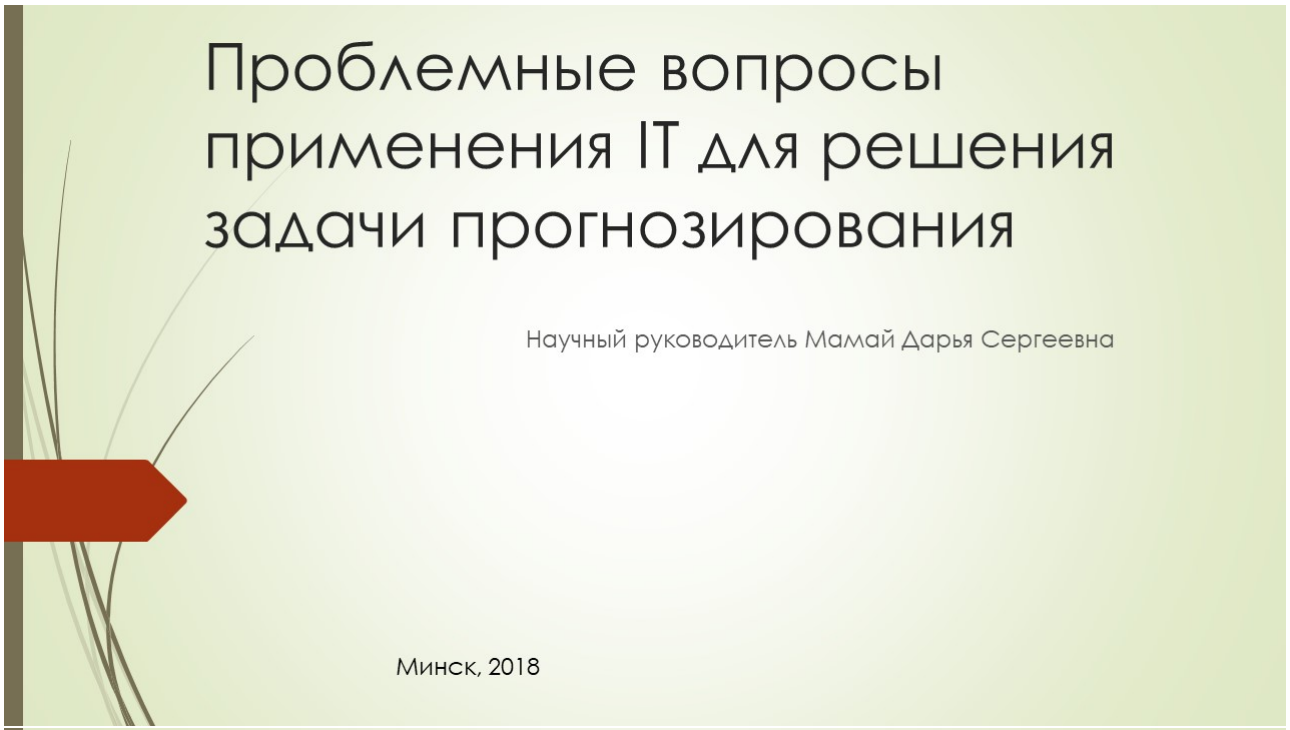
4. J. Kahn. Neural network prediction of NFL football games / J. Kahn. // World Wide Web Electronic Publication. - 2003 – P. 9 – 15.

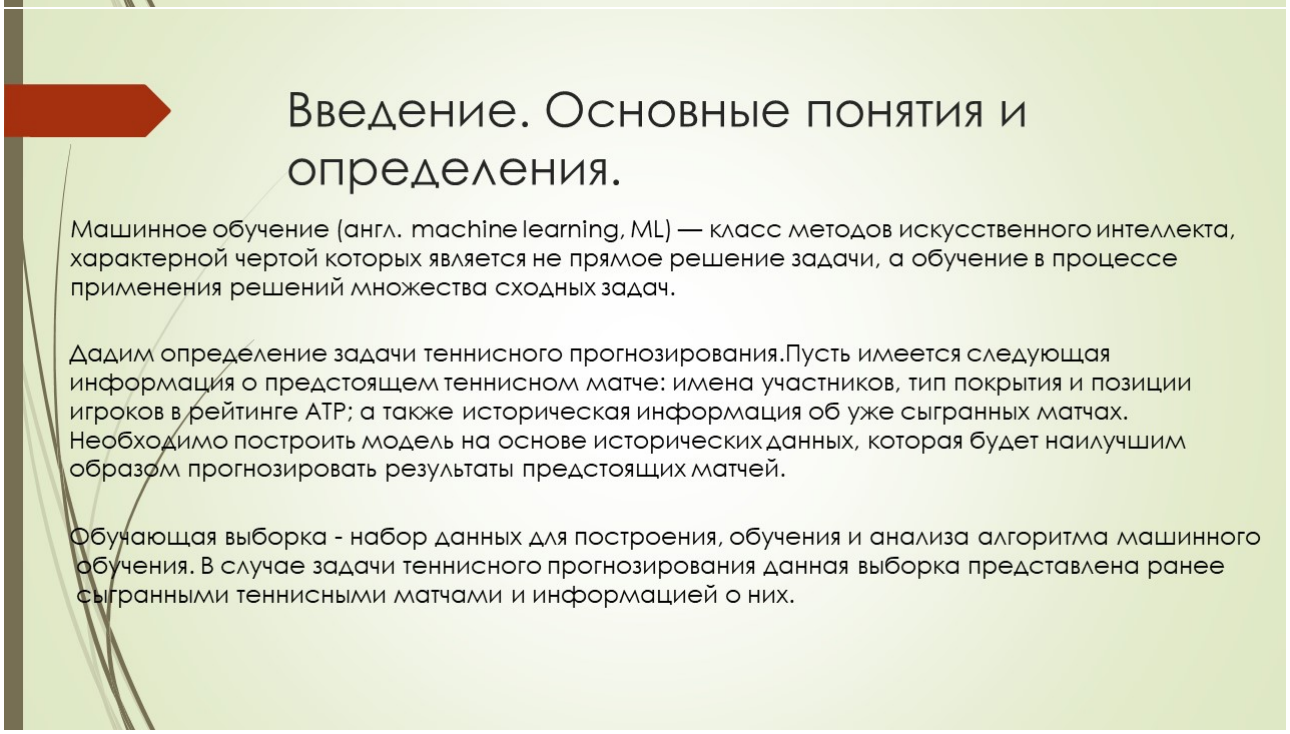
5. Pandas package overview // Информационный портал http://pandas.pydata.org/ [Электронный ресурс]. – 2018. - Режим доступа: http://pandas.pydata.org/pandas-docs/stable/overview.html - Дата доступа: 04.11.2018.

ПРИЛОЖЕНИЯ

Приложение А

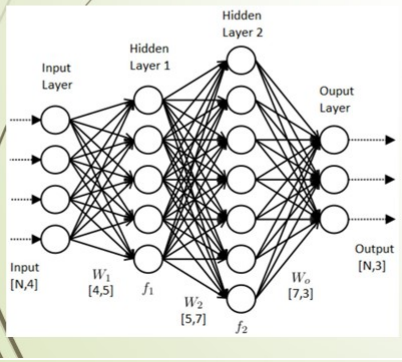
Презентация





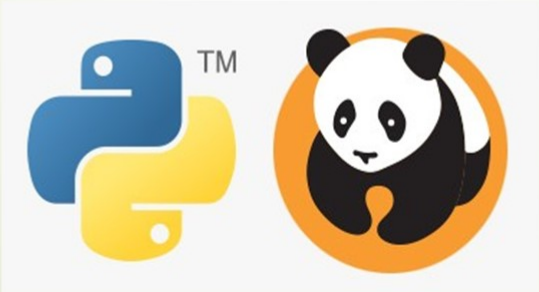
Общая схема решения задачи прогнозирования

1. Поиск и фильтрация данных, построение обучающей выборки;
2. Отбор наиболее релевантных признаков для построения обучающей выборки;
3. Построение и обучение модели;
4. Анализ полученных результатов.



Построение обучающей выборки

Для построения обучающей выборки из источника 1 была загружена информация об мужских одиночных теннисных матчах, сыгранных под эгидой ATP, за 2004-2018 годы. Для фильтрации данных и построения выборки использовался фреймворк Pandas.
Pandas – библиотека языка Python, предназначенная для обработки и анализа данных. Работа библиотеки построена поверх библиотеки NumPy. Библиотекой предоставляются разнообразные структуры и алгоритмы числовыми данными и временными рядами.



Построение и обучение алгоритмов.

Алгоритмы для решения поставленной задачи:

1. Логистическая регрессия;
2. Искусственная нейронная сеть (ИНС)
3. Метод опорных векторов

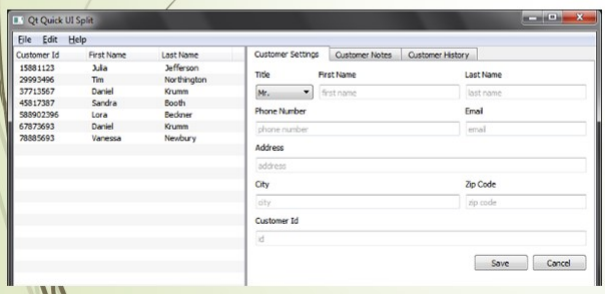
Keras – открытая библиотека на языке Python, предназначенная для построения нейросетевых моделей. Представляет собой интерфейс, надстроенный над Deeplearning4j, TensorFlow и Theano.
Scikit-learn – библиотека для машинного обучения, написанная на языке Python. В нее включены разнообразные алгоритмы классификации, регрессии и кластеризации, такие как случайный лес, метод опорных векторов, k-means, DBSCAN и другие.

scikits learn
machine learning in Python

K Keras
A deep learning library

Построение графического интерфейса

Qt — кроссплатформенный фреймворк для разработки программного обеспечения на языке программирования C++. Есть также «привязки» ко многим другим языкам программирования: Python — PyQt, PySide; Ruby — QtRuby; Java — Qt Jambi; PHP — PHP-Qt и другие.



Qt

Средства для написания реферата

Latex — наиболее популярный набор макрорасширений (или макропакет) системы компьютерной вёрстки TeX, который облегчает набор сложных документов. В типографском наборе системы TeX форматируется традиционно как Latex.