Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных сетей

ОТЧЕТ

по производственной практике

на предприятии УПУП «АйБиЭй АйТи Парк»

|  |  |
| --- | --- |
| Выполнил |  |
| студент гр. 751001: | Бобко А. В. |
| Руководитель практики |  |
| от университета: | Шостак Е. В. |
| Руководитель практики |  |
| от предприятия: | Запрутский Г. К. |

Минск 2020

СОДЕРЖАНИЕ

[**ВВЕДЕНИЕ** 4](#_Toc44282779)

[**1.** **ОБЩИЕ СВЕДЕНИЯ О ПРЕДПРИЯТИИ** 5](#_Toc44282780)

[1.1 Структура предприятия 5](#_Toc44282781)

[1.2 Предметная область предприятия 6](#_Toc44282782)

[1.3 Стандарты и нормативные документы предприятия 9](#_Toc44282783)

[**2.** **ОБЗОР ИСПОЛЬЗУЕМЫХ ТЕХНОЛОГИЙ** 11](#_Toc44282784)

[2.1 Python 11](#_Toc44282785)

[2.2 Jupyter Notebook 12](#_Toc44282786)

[2.3 NumPy 13](#_Toc44282787)

[2.4 Pandas 13](#_Toc44282788)

[2.5 Matplotlib 14](#_Toc44282789)

[2.6 Scikit-learn 15](#_Toc44282790)

[2.7 CatBoost 16](#_Toc44282791)

[2.8 LibROSA 17](#_Toc44282792)

[**3.** **ВЫПОЛНЕНИЕ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ** 18](#_Toc44282793)

[3.1 Музыкальная теория 18](#_Toc44282794)

[3.2 Написание программы 19](#_Toc44282795)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** 26](#_Toc44282796)

**ВВЕДЕНИЕ**

Одним из главных этапов учебного процесса и подготовки специалистов является производственная практика, которая позволяет приобрести практические навыки, а также закрепить теоретическую базу, изученную на занятиях в университете.

Цель прохождения производственной практики – обучение применению изученной теоретической базы посредством ознакомления с работой реального предприятия, использованию методов и принципов разработки программного обеспечения в реальных условиях на предприятии, приобретение новых полезных профессиональных навыков, таких как работа в команде, ответственное выполнение своей части работы в срок, общение с руководителем и другие необходимые для работы по специальности качества и умения.

В качестве типового задания по производственной практике были поставлены следующие задачи:

* Ознакомление со структурой предприятия;
* Изучить технологии разработки программных средств на предприятии;
* Изучить стандарты и нормативные документы, применяемые на предприятии при разработке программных средств;
* Выполнить индивидуальное задание.

1. **ОБЩИЕ СВЕДЕНИЯ О ПРЕДПРИЯТИИ**

Компания «АйБиЭй» является альянсом компаний, специализирующихся в области информационных технологий, разработчиков, производителей и поставщиков решений и услуг в ИТ-сфере в странах Центральной и Восточной Европы. Компания достаточно крупная и состоит из множества отделов и подразделений. Она является одной из крупнейших белорусских организаций в сфере ИТ. В организации работает свыше 2500 высококвалифицированных специалистов и менеджеров, имеющих опыт работы в различных направлениях информационных технологий. Многие специалисты компании прошли профессиональное обучение в учебных центрах компаний IBM, SAP, Microsoft, PTC, а также в Institute IBA и имеют соответствующие сертификаты. Значительная часть сотрудников предприятия имеет многолетний опыт в области разработки и сопровождения прикладного и системного программного обеспечения, многие из них участвуют в международных проектах IBA Group.

# Структура предприятия

Так как «АйБиЭй» является международной компанией, ее офисы

располагаются по всему миру. Их офисы имеются в Беларуси, Болгарии, Германии, США и т.д. Полный список стран изображен на рисунке 1.1

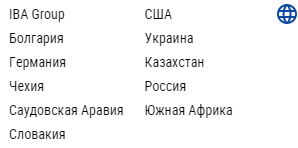


Рисунок 1.1 – Список стран, в которых присутствуют офисы компании «АйБиЭй»

В Беларуси офисы компании располагаются в таких городах как Минск, Гомель и Новополоцк. В Минске находится около 4 офисов + «Институт АйБиЭй». Головной белорусский офис находится по адресу Программистов, 3.

В Беларуси у «АйБиЭй» имеется около 12 крупных департаментов, которые в свою очередь подразделяются на отделы и команды. Каждый департамент занимается решением своих определенных задач. У каждого департамента есть свой директор, департаменты постоянно сотрудничают друг с другом в разработке программного обеспечения. Во главе компании находится ее президент, у которого есть свои заместители. Таким образом должности можно представить в виде сложного дерева, в корне которого находится президент компании, а в листьях – молодые специалисты.

# Предметная область предприятия

Компании приходится работать, используя огромное число направлений и технологий. У «АйБиЭй» имеется огромный опыт в разработке продуктов для различных отраслей, таких как энергетика, добывающая промышленность, финансы и т.д. Список отраслей и некоторых клиентов компании можно видеть на рисунке 1.2.



Рисунок 1.2 – Список отраслей и клиентов, с которыми приходилось работать компании «АйБиЭй»

Компания создает продукты, используя различные направления в ИТ и технологии. Они решают разнообразные бизнес-задачи клиентов в области цифровой трансформации предприятия, автоматизации бизнес-процессов, HR, управления информацией, информационной безопасности, взаимодействия с поставщиками, клиентами, аналитики, планирования и прогнозирования, сервис по поддержке решений на zOS.

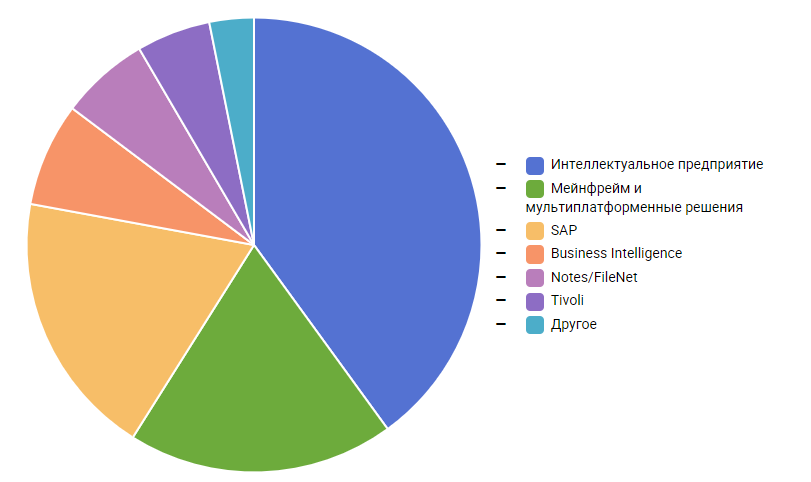


Рисунок 1.3 – Соотношение разрабатываемых продуктов компанией в бизнес-сфере

Компания занимается такими областями, как исследование и анализ данных, машинное обучение, мобильная разработка, SAP, тестирование и поддержка продукта, разработка WEB-приложений и прочее. Направления и компетенции компании можно увидеть на следующем изображении.

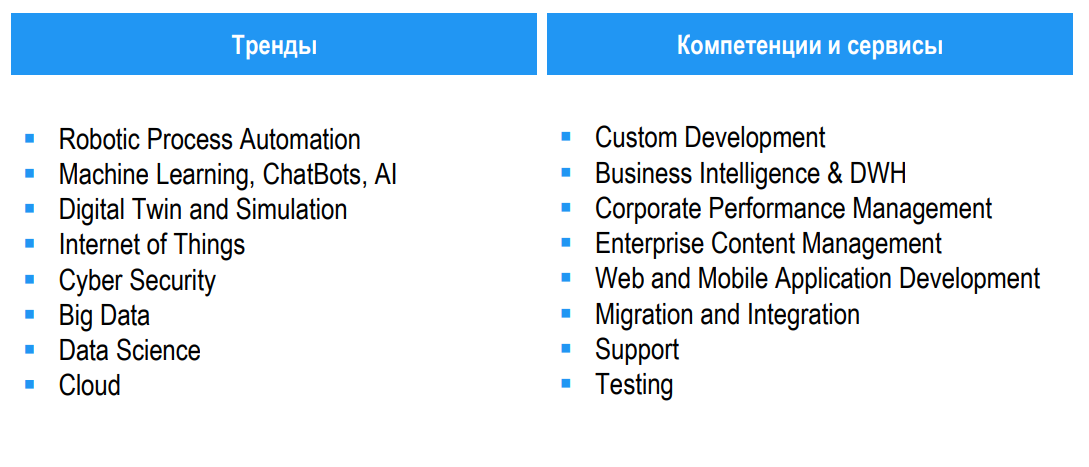


Рисунок 1.4 – Направления и компетенции компании

Компания старается соответствовать современным тенденциям и современных технологиям, поэтому она ведет разработки в набирающих обороты направлениях (IoT, Big Data, Cloud).

Также компания занимается разработкой собственных продуктов. Среди них можно привести следующие примеры:

* VAS – продукт, который анализирует разрозненные данные, находит скрытые взаимосвязи и помогает принимать обоснованные решения;
* Канцлер NEXT – программный продукт для управления цифровым контентом предприятия и автоматизации документооборота;
* Talent Constructor – семейство программных продуктов для эффективного управления человеческими ресурсами;
* AgronomX – платформа для управления растениеводством с помощью мониторинга и анализа данных о состоянии почвы, погоды, вегетации, потребности в удобрениях и увлажнении. Она помогает принимать верные решения по сохранению урожая, используя технологии интернета вещей (IoT) и точные агрономические модели.

Компания регулярно получает разнообразные награды за собственные продукты и решения. Разнообразие проектов и слежка за тенденциями позволяет «АйБиЭй» занимать одну из лидирующих позиций на рынке.

Компания сотрудничает с огромным числом известных брендов и организаций. В их числе такие знаменитые компании, как RedHat, PTC, Huawei, Hewlett Packard Enterprise. Другие партнеры компании изображены на рисунке 1.5.



Рисунок 1.5 – Партнеры компании «АйБиЭй»

# Стандарты и нормативные документы предприятия

Для обеспечения высокого качества выпускаемых продуктов компания «АйБиЭй» старается придерживаться определенных стандартов и нормативных документов в своей работе.

При работе с клиентами используются различные методологии управления проектами, которые подбираются индивидуально для каждого клиента. Примеры таких методологий: AGILE, Waterfall (“Водопадная” классическая модель), интерактивная модель.

Для обеспечения устойчивого развития «АйБиЭй» работает над созданием, внедрением и совершенствованием систем менеджмента. Компания регулярно подтверждает сертификатами соответствие своих систем менеджмента определенным стандартам. Компания имеет следующие сертификаты:

* сертификат соответствия системы менеджмента качества проведения научно-исследовательских работ, проектирования, разработки, производства, интеграции, инсталляции, настройки и сопровождения программного обеспечения и автоматизированных информационных систем требованиям СТБ ISO 9001–2015;
* сертификат соответствия системы менеджмента информационной безопасности применительно к проектированию, разработке, производству и сопровождению программного обеспечения и автоматизированных информационных систем требованиям СТБ ISO/IEC 27001-2016 (ISO/IEC 27001:2013);
* сертификат соответствия системы менеджмента качества проведения научно-исследовательских работ, проектирования, разработки, производства, интеграции, инсталляции, настройки и сопровождения программного обеспечения и автоматизированных информационных систем требованиям DIN EN ISO 9001:2015.

И это далеко не исчерпывающий список сертификатов. В 1999, 2002, 2005, 2010, 2015 и 2018 годах «АйБиЭй» удостоена [Премии Правительства Республики Беларусь](https://iba.by/about-iba-group/awards-recognitions/premii-pravitelstva-respubliki-belarus/) за достижения в области качества.

«АйБиЭй» ответственно относится к соблюдению права каждого клиента, поставщика, бизнес-партнера и сотрудника на конфиденциальность и неприкосновенность персональных данных. Компания использует самые современные технологии и стандарты для обеспечения информационной безопасности своих клиентов. Компания принимает все необходимые меры по организации контроля физического доступа, контроля логического доступа, внутриорганизационного контроля безопасности, защиты передачи данных и аудита процессинга. «АйБиЭй» и внешние аудиторы проводят ежегодную проверку безопасности на всех объектах компании. Служба информационной безопасности «АйБиЭй» постоянно контролирует корпоративную сеть и проверяет деятельность организации на соответствие нормативным актам.

Сотрудники «АйБиЭй» проходят инструктаж по требованиям безопасности и в письменном виде подтверждают свои обязательства по соблюдению норм информационной безопасности. Кроме того, сотрудники, имеющие доступ к конфиденциальным данным, подписывают соглашение о неразглашении. Политика информационной безопасности компании основана на требованиях международных стандартов и стандартах безопасности бизнес-партнеров компании. «АйБиЭй» проводит регулярные выборочные проверки соблюдения ее сотрудниками корпоративных и клиентских (при работе по внешним контрактам) стандартов информационной безопасности, а также использования информации о компании и ее программных и аппаратных ресурсов.

Для управления физическим доступом к информационным ресурсам все помещения компании разделены на зоны контролируемого доступа в соответствии с международными, национальными и корпоративными стандартами.

«АйБиЭй» сертифицировала свою систему менеджмента информационной безопасности на соответствие требованиям стандарта СТБ ISO/IEC 27001. Компания имеет следующие сертификаты:

* сертификат соответствия системы менеджмента информационной безопасности применительно к проектированию, разработке, производству и сопровождению программного обеспечения и автоматизированных информационных систем требованиям СТБ ISO/IEC 27001-2016 (ISO/IEC 27001:2013);
* сертификат соответствия внедренной системы менеджмента информационной безопасности проектирования, разработки и сопровождения программного обеспечения требованиям СТБ ISO/IEC 27001-2016 (ISO/IEC 27001:2013);
* сертификат соответствия системы менеджмента информационной безопасности: ИТ-решения – разработка и поддержка программного обеспечения требованиям ČSN ISO/IEC 27001:2014.

1. **ОБЗОР ИСПОЛЬЗУЕМЫХ ТЕХНОЛОГИЙ**

Для реализации индивидуального задания была необходима группа технологий и инструментов. В данном разделе представлен их краткий обзор.

# 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. Данный язык является мультипарадигменным: он позволяет писать код, используя структурное, обобщенное, объектно-ориентированное, функциональное и аспектно-ориентированное программирование.

На практике чаще всего используется CPython - интерпретатор байт-кода Python написанный на C. Он поддерживает большинство активно используемых платформ и является бесплатным.

Разработчики языка Python придерживаются определённой философии программирования, называемой «The Zen of Python». Принципы данной философии изображены на рисунке 2.1.

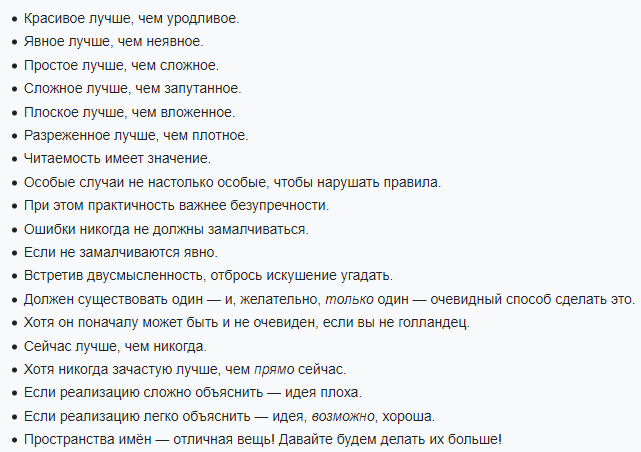


Рисунок 2.1 – Философия Python («The Zen of Python»)

Данный язык очень удобен и является подходящим вариантов для быстрого написания кода и выполнения сложным математических вычислений.

# Jupyter Notebook

Jupyter Notebook – невероятно мощный инструмент для интерактивной разработки и представления проектов в области наук о данных. Он очень удобен для разработки программ на Python, а также создания презентаций.

Проект Jupyter является приемником более раннего проекта IPython Notebook, который впервые был опубликован в 2010 году в качестве прототипа. IPython — интерактивная оболочка для языка программирования Python, которая предоставляет расширенную интроспекцию, дополнительный командный синтаксис, подсветку кода и автоматическое дополнение. Клиентская часть Jupyter Notebook показана на рисунке 2.2.

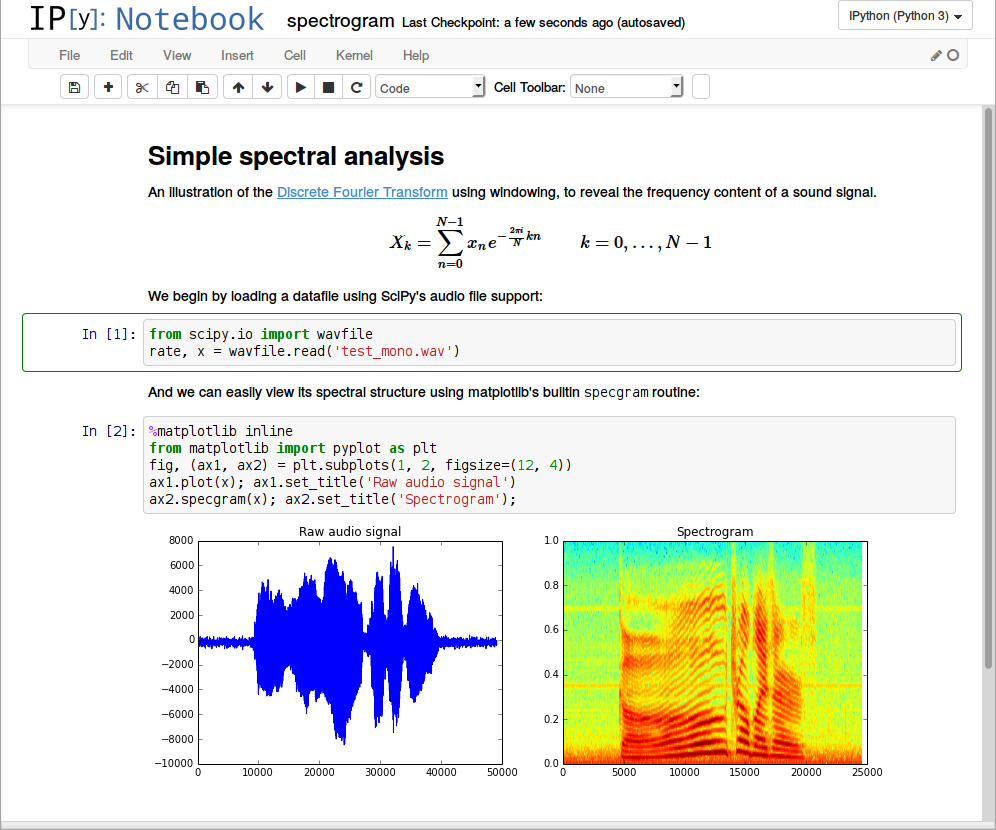


Рисунок 2.2 – Внешний вид клиентской части Jupyter Notebook

# NumPy

NumPy – библиотека с открытым исходным кодом для языка программирования Python. Она очень часто применяется при работе с большими данными, так как она имеет хорошую поддержку многомерных массивов и высокоуровневых математических функций, предназначенных для работы с матрицами.

Математические алгоритмы, реализованные на интерпретируемых языках (например, Python), часто работают гораздо медленнее тех же алгоритмов, реализованных на компилируемых языках (например, Си, Java). Библиотека NumPy предоставляет реализации вычислительных алгоритмов (в виде функций и операторов), оптимизированные для работы с многомерными массивами. В результате любой алгоритм, который может быть выражен в виде последовательности операций над массивами (матрицами) и реализованный с использованием NumPy, работает так же быстро, как эквивалентный код, выполняемый в MATLAB.

# Pandas

Pandas – библиотека на языке Python для обработки и анализа данных. Работа Pandas с данными строится поверх библиотеки NumPy, являющейся инструментом более низкого уровня. Предоставляет специальные структуры данных и операции для манипулирования числовыми таблицами и временными рядами. Данная библиотека позволяет удобно работать с различными таблицами данных, чем-то напоминая синтаксис при работе с базами данных.

Pandas имеет два основных типа данных: Series (серия) и DataFrame (дата фрейм). Структура Series представляет из себя объект, похожий на одномерный массив, но отличительной его чертой является наличие ассоциированных меток (индексов), вдоль каждого элемента из списка. Такая особенность превращает его в ассоциативный массив или словарь в Python.

Объект DataFrame лучше всего представлять себе в виде обычной таблицы и это правильно, ведь DataFrame является табличной структурой данных. В любой таблице всегда присутствуют строки и столбцы. Столбцами в объекте DataFrame выступают объекты Series, строки которых являются их непосредственными элементами. Пример DataFrame можно увидеть на следующем рисунке.

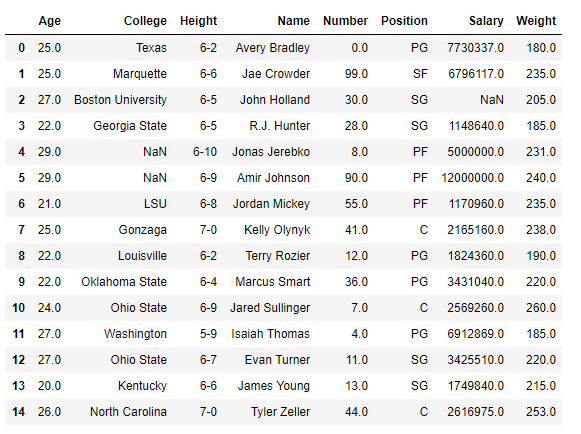


Рисунок 2.3 – Пример объекта типа данных DataFrame

# Matplotlib

Matplotlib является одним из основных инструментов для анализа данных и работы с ними. Matplotlib – библиотека на языке программирования Python для визуализации данных двумерной графикой.

Пакет поддерживает многие виды графиков и диаграмм:

* графики;
* диаграммы разброса;
* столбчатые диаграммы;
* круговые диаграммы;
* ствол-лист диаграммы;
* контурные графики;
* поля градиентов;
* спектральные диаграммы.

Примеры графиков, построенных с помощью Matplotlib представлены на рисунке ниже.

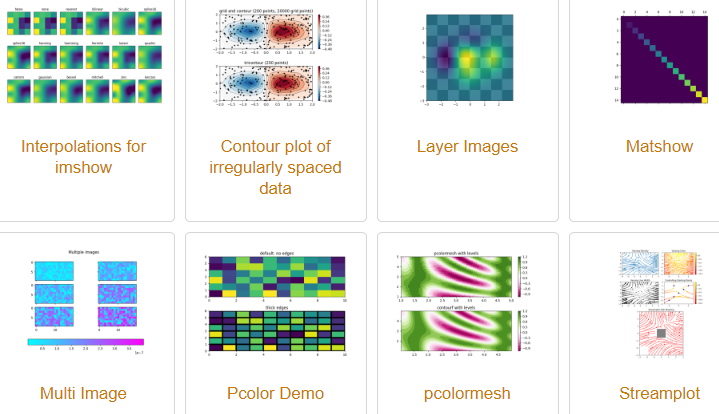


Рисунок 2.4 – Примеры графиков, построенных с помощью библиотеки Matplotlib

# Scikit-learn

Scikit-learn - бесплатная библиотека машинного обучения для языка программирования Python.

Она содержит реализацию множества алгоритмов машинного обучения для решения проблем регрессии, классификации, кластеризации, уменьшения размерности и т.д.

Данная библиотека содержит множество инструментов для предобработки данных, такие как LabelEncoder, OneHotEncoder и т.д. Также в библиотеке присутствует огромное число метрик и инструментов для машинного обучения, среди которых можно привести наиболее известные: средняя квадратичная ошибка, точность, полнота и т.д. Благодаря содержащимся в ней классам мы легко можем разбивать наши данные на подгруппы для обучения и тестирования нашей модели. С помощью классов, обеспечивающих перекрестную проверку, можно проверить эффективность нашей модели, а класс GridSearchCV позволяет выбрать из группы гиперпараметров наиболее оптимальные.

У Scikit-learn очень удобная документация, которая отлично расписывает все возможности библиотеки с примерами.

# CatBoost

CatBoost – открытая программная библиотека, разработанная компанией Яндекс и реализующая уникальный патентованный алгоритм построения моделей машинного обучения, использующий одну из оригинальных схем градиентного бустинга.

Данный инструмент от Яндекса обладает более высокой точностью, нежели другие алгоритмы градиентного бустинга. Также он позволяет использовать не только числовые данные для обучения, что позволяет извлекать всю возможную информацию из данных

Градиентный бустинг – это техника машинного обучения для задач классификации и регрессии, которая строит модель предсказания в форме ансамбля слабых предсказывающих моделей, обычно деревьев решений. Смысл алгоритмов бустинга заключается в том, что у нас есть последовательность из моделей машинного обучения, где каждая следующая модель обучается на наших данных, но при этом уделяет больше внимания тем данным, на которых предыдущая модель ошиблась. В ходе множества испытаний выяснилось, что такая схема является очень эффективной и позволяет добиваться точных результатов.

Задача регрессии – предсказать числовое значение какого-либо параметра, а классификации – отнести данный объект к одному из существующих классов.

В общем виде алгоритмы бустинга можно описать следующей картинкой.

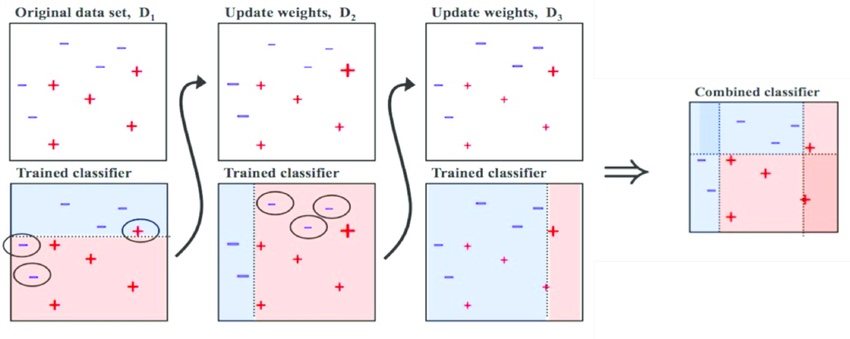


Рисунок 2.5 – Схема работы алгоритмов бустинга в машинном обучении

# LibROSA

Также была использована библиотека для Python под названием LibROSA. Это модуль для анализа звуковых сигналов, предназначенный для работы с музыкой. Он включает все необходимое для создания системы MIR (поиск музыкальной информации) и подробно задокументирован вместе со множеством примеров и руководств.

Аудиоанализ – область, включающая автоматическое распознавание речи (ASR), цифровую обработку сигналов, а также классификацию, тегирование и генерацию музыки – представляет собой развивающийся поддомен приложений глубокого обучения. Некоторые из самых популярных и распространенных систем машинного обучения, такие как виртуальные помощники Alexa, Siri и Google Home, – это продукты, созданные на основе моделей, извлекающих информацию из аудиосигналов.

1. **ВЫПОЛНЕНИЕ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ**

В качестве индивидуального задания выступает следующая задача: необходимо написать простой классификатор для распознавания аккордов в музыкальных композициях. В качестве исходных данных имеется множество аудиофайлов с расширением .wav, на которых записано различное звучание музыкальных аккордов на гитаре.

# Музыкальная теория

Музыкальные композиции состоят из звуков, которые обозначаются нотами.  Нота в музыке – графическое обозначение звука музыкального произведения, один из основных символов современной музыкальной нотации. Вариации в изображении ноты, а также сочетания с другими символами позволяют задать такие характеристики обозначаемого нотой звука, как высоту, длительность и порядок исполнения по отношению к другим звукам. В современном темперированном строе нот – 7 (до, ре, ми, фа, соль, ля, си), а звуков – 12 (в пределах каждой октавы). Обозначение 12 звуков показано на рисунке 3.1.

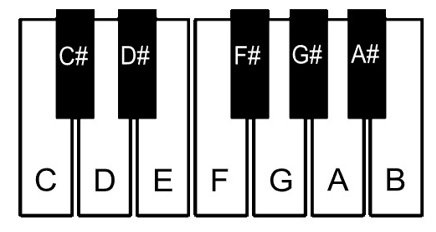


Рисунок 3.1 – Символическое обозначение 12 звуков

Октава – музыкальный интервал, в котором соотношение частот между звуками составляет один к двум. Субъективно на слух октава воспринимается как устойчивый, базисный музыкальный интервал. Два звука, отстоящие на октаву, воспринимаются очень похожими друг на друга, хотя явно различаются по высоте.

Также стоит упомянуть про бемоль и диез. Бемоль – знак альтерации, обозначающий понижение стоящих справа от него нот на один полутон. Диез – знак альтерации, обозначающий повышение стоящих справа от него нот на один полутон. Изображения диеза и бемоля представлены на рисунке 3.2.

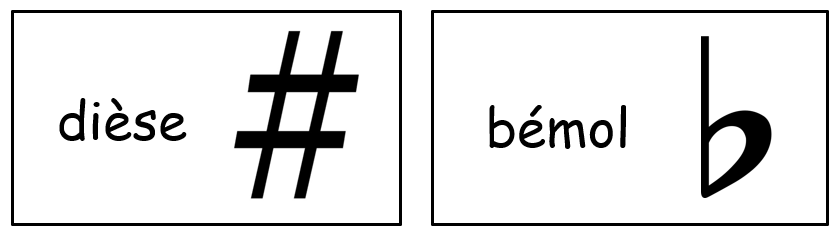


Рисунок 3.2 – Символическое обозначение диеза и бемоля

Звуки могут складываться в аккорды. Аккорд – одновременное сочетание трёх и более музыкальных звуков разной высоты, воспринимаемое слухом как целостный элемент звуковысотной вертикали. У аккордов также есть свои обозначения. Примеры нескольких простых аккордов показаны на рисунке.

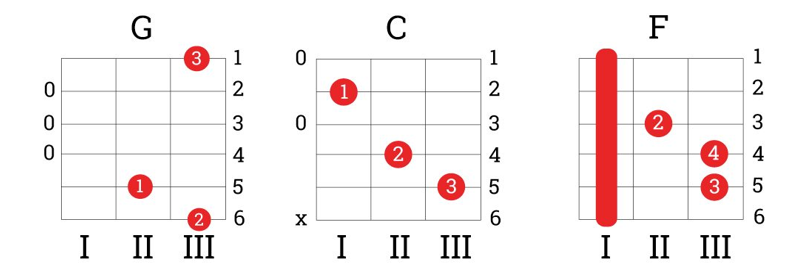


Рисунок 3.3 – Схемы нескольких аккордов на гитаре

# Написание программы

Первоначальной идеей было разделение песни на аккорды, распознавание этих аккордов и вывод их пользователю. Библиотека LibROSA позволяет получить хромаграмму песни. Хромаграмма – это вектор, содержащий высоту каждого из 12 звуков.  Код извлечения хромаграммы показан ниже.

y, sr = librosa.load(file\_path)

chroma\_orig = librosa.feature.chroma\_cqt(y=y, sr=sr)

chroma\_os = librosa.feature.chroma\_cqt(y=y,

sr=sr,

bins\_per\_octave=12\*3

)

y\_harm = librosa.effects.harmonic(y=y, margin=8)

chroma\_os\_harm = librosa.feature.chroma\_cqt(

y=y\_harm,

sr=sr,

bins\_per\_octave=12\*3

)

chroma\_filter = np.minimum(

chroma\_os\_harm,

     librosa.decompose.nn\_filter(chroma\_os\_harm,

                                 aggregate=np.median,

                                 metric='cosine')

)

Если отрисовать график, у которого по оси Ox находится время, по Oy – звук, а цветом будет отображена величина звука, то получится картинка, похожая на рисунок 3.4.

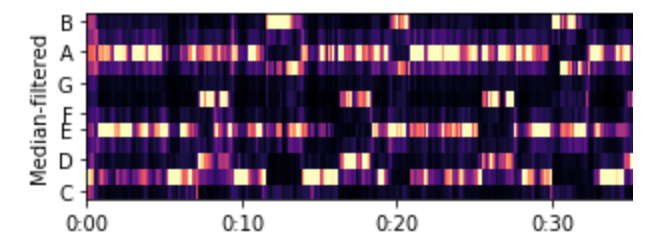


Рисунок 3.4 – График хромаграмм музыкальной композиции

Нам необходимо из данных получить обучающую выборку. Идея заключается в том, чтобы брать запись аккорда, из определенных ее моментов получать высоту звуков и записывать ее в таблицу. Это можно описать следующим кодом.

COEFFICIENTS = [0.5]

    df = pd.DataFrame(columns=[\*COLUMNS, "Chord"])

    for chord in chords:

        for chord\_num in range(num\_of\_chords):

            chord\_chromagram = get\_chromagram(

                "{0}\\{1}\\{2}{3}.{4}".format(

                    dir\_path, chord, chord,

chord\_num+1,

file\_format

)

            )

            for coef in COEFFICIENTS:

                index = int(len(chord\_chromagram)\*coef)

                chord\_chromagram\_slice = [

                    chord\_chromagram[i][index]

for i in range(12)

                ]

                chord\_data = {

                    k: v for k, v in

zip(COLUMNS, chord\_chromagram\_slice)

                }

                chord\_data["Chord"] = chord

                df = df.append(chord\_data,

ignore\_index=True

)

Полученный DataFrame из хромаграмм имеет следующий вид.

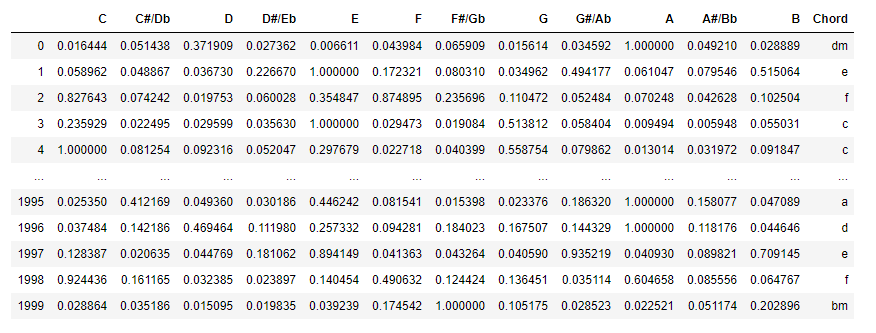


Рисунок 3.5 – DataFrame хромаграмм

Уже на графике хромаграмм заметно определенное разделение на аккорды. Попробуем самостоятельно разделить полученный ранее график, чтобы сделать определенные выводы. Ручное разбиение графика изображено на рисунке 3.6.

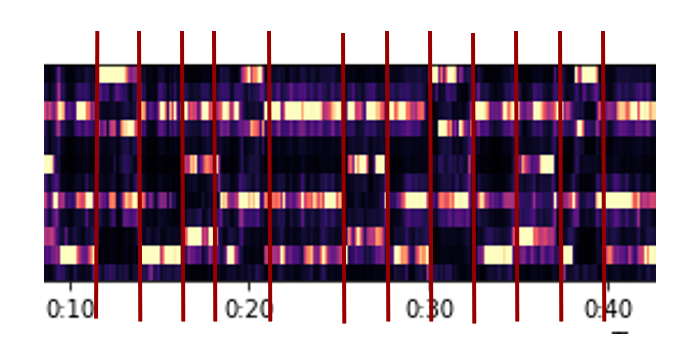


Рисунок 3.6 – Ручное разбиение графика хромаграмм на аккорды

Следующая идея состояла в том, чтобы в каждый момент времени звуки, сила звучания которых очень слабая, приравнять к 0, а сильные звуки сделать равными 1. Однако, в дальнейшем было замечено, что данный способ не эффективен и приводит к ошибкам и неопределенности. Тогда появилась другая идея: распознавать все аккорды в конкретные моменты времени, объединять эту информацию, делать соответствующие выводы и показывать результат пользователю.

Разобьем наши данные на обучающую и тестовую выборки. После чего обучим нашу модель на имеющихся данных. Процесс разбиения и обучения представлен в следующем коде.

from catboost import CatBoostClassifier

cb\_chord\_classifier = CatBoostClassifier()

cb\_chord\_classifier.fit(X\_train, y\_train)

Следующим этапом проверяем, используя метрики точности и полноты, качество нашей модели. Расчет метрик отображен на рисунке 3.7.

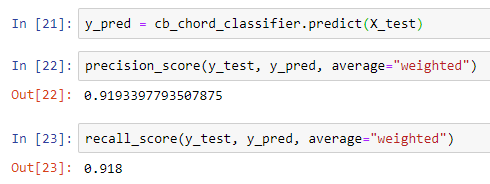


Рисунок 3.7 – Расчет метрик качества модели

Как видим, наши точность и полнота соответственно равны 0.919 и 0.918, что является очень хорошим результатом. Попробуем изобразить график, на котором показаны аккорды, звучащие в определенный промежуток времени. Фрагмент такого графика показан на следующем рисунке.

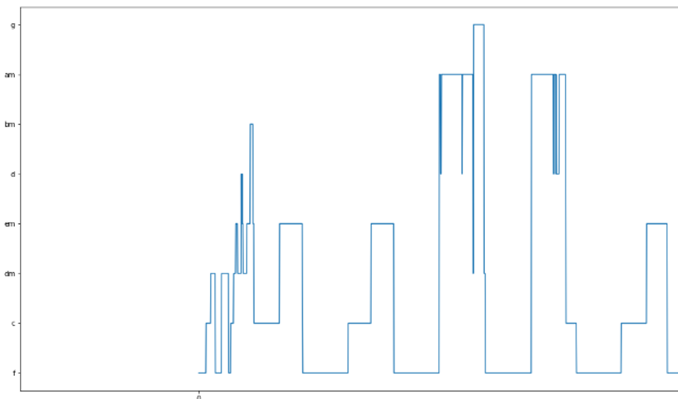


Рисунок 3.8 – График звучания аккордов в музыкальной композиции

Получили график, в котором присутствует много шумов и который не отображает реальную картину, так как аккорды не могут меняться с такой огромной скоростью (меньше, чем за 1 секунду), потому что у человека имеются определенные физические ограничения. Поэтому нам необходимо произвести очистку результата от таких выбросов. Идея состоит в том, чтобы смотреть длительность звучания аккорда и, если она слишком маленькая, смотреть на аккорды, которые окружают данный, затем заменять выброс на нормальный аккорд. Код очистки результата представлен ниже.

def clear\_chords(chords, time, MIN\_CHORD\_DURATION=1):

    chords = chords.copy()

    li = 0

    last\_ok = chords[0]

    while li < len(chords) - 1:

        last = chords[li]

        hi = li + 1

        while (hi < len(chords) - 1)

and (chords[hi] == last):

hi += 1

if time[hi] - time[li] < MIN\_CHORD\_DURATION:

            chords[li: hi] = [last\_ok

for \_ in range(hi-li)

]

        else:

            last\_ok = last

        li = hi

    return chords

Рисуем график полученного результата.

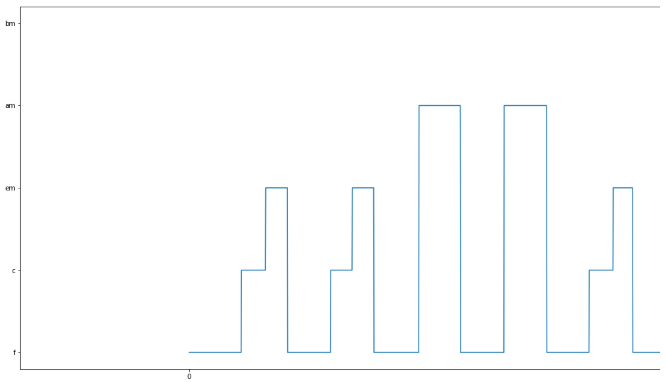


Рисунок 3.9 – Новый график звучания аккордов в музыкальной композиции

Данный график больше похож на правду и в нем прослеживаются четкие шаблоны звучания композиции. Такая ситуация является приемлемой для вывода пользователю.

**ЗАКЛЮЧЕНИЕ**

В процессе прохождения производственной практики были изучены и закреплены навыки в анализе и работе с данными. Было произведено ознакомление со структурой предприятия по месту практики, были изучены технологии разработки программных средств на предприятии. Также были рассмотрены стандарты и нормативные документы, применяемые на предприятии при разработке программных средств.

По итогу практики было выполнено индивидуально задание, выданное на предприятии, которое заключалось в написании простого классификатора аккордов в музыкальной композиции. В результате практики были приобретены навыки эксплуатации ИТ-систем и их компонентов.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

[[1] Форум для программистов. Электронные данные. Режим доступа: https://stackoverflow.com.

[2] Python 3.8.3 documentation. Электронные данные. Режим доступа: https://docs.python.org/3/.

[3] Sklearn documentation. Электронные данные. Режим доступа: https://scikit-learn.org.

[4] NumPy Documentation. Электронные данные. Режим доступа: https://numpy.org.

[5] Pandas Documentation. Электронные данные. Режим доступа: https://pandas.pydata.org.

[6] LibROSA Documentation. Электронные данные. Режим доступа: https://pypi.org/project/librosa/.

[7] Matplotlib documentation. Электронные данные. Режим доступа: https://matplotlib.org.

[8] Официальный сайт компании IBA Group. Электронные данные. Режим доступа: https://iba.by/.

[9] Википедия – свободная энциклопедия. Электронные данные. Режим доступа: https://ru.wikipedia.org.

[10] Официальный информационный портал компании IBA Group. Электронные данные. Режим доступа: http://iba.by/about-iba-group/executive-summary/.