МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

01.03.02 Прикладная математика и информатика
(код и наименование направления подготовки, специальности)

Компьютерные технологии и математическое моделирование
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему: «Разработка системы классификации эффективности сотрудников с применением технологий машинного обучения (на примере ООО «Гран Лимитед»)»

Студент Н.А. Лисенков
(И.О. Фамилия) (личная подпись)

Руководитель канд.тех.наук, доцент кафедры ПМИ, В.С. Климов
(ученая степень, звание, И.О. Фамилия)

Консультант старший преподаватель кафедры ТиПП, М.В. Дайнеко
(ученая степень, звание, И.О. Фамилия)

Тольятти 2021

АННОТАЦИЯ

Темой бакалаврской работы является «Разработка системы классификации эффективности сотрудников на предприятии ООО «Гран Лимитед» с применением технологий машинного обучения».

Объектом исследования является бизнес-процесс расчета заработной платы на предприятии.

Предметом исследования выпускной квалификационной работы является алгоритм классификации эффективности сотрудников.

Целью исследования является разработка системы расчета зарплатных грейдов с применением технологий машинного обучения.

Для достижения поставленной цели были выдвинуты следующие задачи:

- Изучить существующие модели оплаты труда сотрудников.
- Изучить стандартные методики классификации объектов.
- Выбрать инструменты разработки системы.
- Сформировать набор данных.
- Оценить эффективность ранжирования.

Выпускная квалификационная работа состоит из введения, трех глав, заключения и списка литературы.

В введении описывается общая структура работы, обозначается комплекс задач, подлежащих решению.

В первой главе описываются существующие модели оплаты труда сотрудников, определяются актуальные проблемы при расчете заработной платы, описываются особенности грейдовой системы оплаты труда и мотивации.

Во второй главе разбираются алгоритмы, используемые на практике для ранжирования объектов, производится учет общих рисков и ожидаемых потерь, осуществляется выбор инструментов для разработки системы.

В третьей главе описываются используемые библиотеки, формируется набор данных, реализуется проектное решение и оценивается его эффективность путем тестирования.

В заключение вынесены выводы о проделанной работе.

Данная выпускная квалификационная работа содержит в себе пояснительную записку, состоящую из 66 страниц, 21 рисунка, 3 таблиц, 16 формул и списка литературы из 31 источника.

ABSTRACT

The title of the graduation work is «Development of a system for classifying the employees' performance at company "Gran Limited" by applying machine-aided learning technologies».

The graduation work consists of 29 figures, 3 tables, 16 formulae, the list of 31 references including 6 foreign sources and 1 appendix.

The object of the graduation work is the business process of effective payroll calculation and the staff motivation.

The subject of the research is the algorithm for classifying employees' performance.

The aim of the study is to develop a system for calculating payrolls using machine-aided learning technologies.

In order to achieve this goal, the following objectives are set:

- to examine the existing employees' labour remuneration models;
- to consider the standard methods for classifying objects;
- to select system development tools;
- to form a data set;
- to evaluate the ranking system effectiveness.

The introduction describes the general structure of the present graduation work and reveals the objectives to be solved.

The first chapter deals with the existing employees' labour remuneration models, identifies current problems related to calculating the payroll, as well as dwells on the features of the remuneration and motivation graded system.

In the second chapter the algorithms used in practice for ranking objects are analyzed, general risks and expected losses are taken into account, as well as the tools for system development are selected.

In the third chapter, the used references are described, a data set is formed, a design solution is implemented and its effectiveness is evaluated by testing.

Содержание

Введение

Проблема организации работы с сотрудниками, занимающимися умственным трудом, достаточно сложна и включает следующие аспекты: необходимость подбора специалистов согласно некоторым выбранным критериям, разворачивание системы обучения специалистов для постоянного повышения их уровня, корректное определение механизмов расчета размеров заработной платы для удержания сотрудников в компании.

И если первым аспектам посвящено множество методик и большая часть этих методов автоматизирована, то вопрос финансовой мотивации сотрудников на сегодняшний день остается открытым.

Актуальность работы заключается в разработке и внедрении автоматизированной системы, позволяющей поддерживать высокую мотивацию сотрудников к эффективной работе.

Целью работы является разработка системы расчета заработной планы по заказу компании "Гран Лимитед" с применением технологий машинного обучения.

Для достижения поставленной цели в работе предполагается решение целого комплекса задач:

– исследование проблемы организации финансовой мотивации сотрудников;

– рассмотрение различных форм оплаты труда и особенностей грейдовой системы;

– анализ существующих алгоритмов, используемых для ранжирования объектов, в том числе и с применением машинного обучения;

– разработка проекта системы и выбор инструментов реализации;

– реализация и тестирование проектного решения;

– оценка эффективности проведенного ранжирования в ходе эксперимента.

Структура работы представлена тремя главами. Первая глава посвящена исследованию проблемы организации финансовой мотивации сотрудников. При этом рассматриваются варианты расчета заработной платы в зависимости от результатов труда, а также модели оплаты труда сотрудников. Отдельная часть посвящена изучению технологий определения размеров оплаты труда с использованием грейдовой системы.

Вторая глава касается алгоритмов, которые используются в основном для ранжирования объектов и включают стандартные методики и технологии машинного обучения, которым уделено особое внимание. Также в рамках второй главы описаны технологии учета общих рисков и ожидаемых потерь в ходе применения классификации с учетом используемых алгоритмов.

Реализация проекта отражена в третьей главе работы, которая включает формирование набора данных, описание используемых библиотек, описание интерфейса разработанного приложения. Также третья глава посвящена отражению результатов проведенного эксперимента и оценки эффективности используемых методов для разделения сотрудников на грейды на основании предварительных оценок их деятельности в компании.

1 Исследование проблемы организации финансовой мотивации сотрудников

1.1 Проблема расчета заработной платы в зависимости от результатов труда

Как правило, компании не используют информационные системы для организации работы с персоналом. Вследствие этого оценка методов дальнейшего развития работников любого предприятия становится если и выполнимой, то совсем непростой задачей. Поэтому структура расчета заработной платы привязана напрямую к отработанному времени или количеству оформленных сделок или проектов.

Большинство предприятий в настоящее время оценивают действия персонала либо вручную (что крайне трудоемко и требует содержания большого отдельного штата сотрудников для данных целей), либо с применением фиксированных ставок оплаты труда, слабо подверженных положительной или отрицательной динамике [6].

На основании уровня средней заработной платы в компании и специалистов исследуемого направления в целом обычно определяется заработная плата конкретного специалиста. Оценка производится «на глаз», зачастую является субъективным видением эффективности работы сотрудника его непосредственным руководителем. Такая система оплаты труда специалиста не может претендовать на объективность, что зачастую подмечают сами работники. Как следствие, данная система оплаты труда служит почвой для конфликтов между сотрудниками и непосредственным руководством, мешая продуктивной деятельности.

Серьезные аналитические методы для корректировки производительности труда персонала и повышения его заинтересованности в результатах своего труда используют исключительно крупные корпорации или ИТ-компании [12].

Одним из важнейших бизнес-процессов в компании является бизнес-процесс финансовой мотивации сотрудников. Правильно сформулировать и

реализовать правила, которые действительно будут положительно сказываться на производительности труда сотрудников (как количественно, так и качественно) – задача неочевидная и совсем не простая.

Существует ряд элементов комплекса управления компанией, некоторые из них имеют прямое отношение к процессам финансовой мотивации персонала. Среди них выделяют:

– управление финансами – в рамках данного элемента формируются бухгалтерские отчетности и определяются необходимые для проекта размеры финансирования. Финансирование, в том числе, необходимо для формирования бюджета финансовой мотивации персонала;

– контроллинг – в рамках данного элемента ведется мониторинг всех доходов и расходов компании. В отношении финансовой мотивации играет роль с точки зрения анализа расходов на финансовую мотивацию сотрудников;

– управление проектами – в рамках данного элемента выполняется мониторинг проектов, контролируются финансовые, трудовые, временные ресурсы;

– управление качеством – в рамках данного элемента осуществляется контроль за качеством и соответствием стандартам всей производимой продукции и оказываемых услуг. Оценку качества можно использовать в качестве одного из факторов финансовой мотивации сотрудников;

– управление персоналом – в рамках данного элемента осуществляется контроль за работниками, их квалификацией, уровнем профессионализма.

Финансовая мотивация оказывает значительное влияние на многие аспекты трудовой деятельности. В зависимости от количества выделенных ресурсов можно регулировать требуемый уровень производительности труда, сохранить ценные на текущий момент кадры [20].

Структура финансовой мотивации персонала включает множество сложных бизнес-процессов. Стоит подробнее остановиться на наиболее значимых из них:

– развитие пресонала – в долгосрочной перспективе позволяет получить высококвалифицированных кадров в случае грамотной мотивации;

– потребности в персонале – финансовая мотивация как делает работу менее привлекательной для некомпетентного сотрудника, не справляющегося со своими обязанностями и получающего заработную плату в виде чистого оклада, так и привлечь специалистов, заинтересованных в высокой оплате своих компетенций и навыков;

– необходимый уровень квалификации персонала – позволяет регулировать средний и медианный уровень квалификации персонала в зависимости от назначаемого размера финансовой мотивации;

– производительность труда персонала – в процессе распределения вознаграждения собираются статистические данные по производительности труда в организации в целом за отчетный период;

– теоретические объемы финансирования мотивации;

– оценка влияния финансовой мотивации на производительность труда;

– расчет необходимых объемов финансовой мотивации для персонала согласно квалификации;

– расчет необходимых объемов финансовой мотивации для персонала;

– оценка результатов применения финансовой мотивации персонала.

Сотрудники отдела кадров должны быть в курсе всех текущих и планируемых проектов. Им необходимо своевременное информирование от линейных руководителей о количестве текущих проектов, о сроках и масштабах проектов, готовящихся к открытию. Также необходимо иметь

постоянный доступ к прогнозированию нагрузки на проект со стороны заказчика и дайджестам (ежедневным планам нагрузки), составляемым менеджерами младшего звена (супервайзерами, бригадирами и т.д.).

Как правило, работы, определенные таким образом, строго ограничены по бюджету. Однако с помощью грамотного управления мотивацией сотрудников можно свести к минимуму расходы на реализуемый проект (Рисунок 1).

Грамотное управление финансовой мотивацией позволяет поддерживать стабильный уровень производительности труда, не повышая расходы компании, если ситуация на проектах устраивает руководителей и заказчика. Также в случае отрицательной динамики по выполняемым на проекте задачам имеется возможность корректировки ситуации в положительную сторону. Однако для мониторинга ситуации и внесения корректировок требуются дополнительные сотрудники. В случае автоматизации процесса финансовой мотивации необходимый штат для поддержки этой системы можно сократить до 3-4 экспертов в области, которыми в случае необходимости могут выступить и опытные сотрудники.

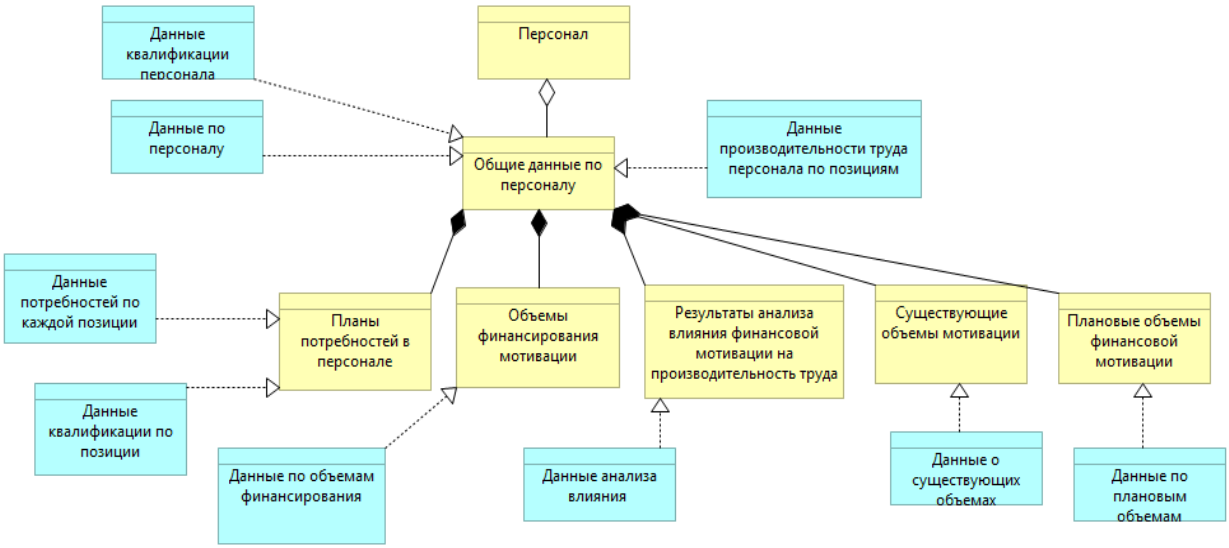


Рисунок 1 – Диаграмма информационной архитектуры финансовой мотивации персонала

Начальные бизнес-требования к системе автоматизации процесса расчета заработной платы с учетом необходимости проведения финансовой мотивации персонала:

a) должны быть улучшены стратегии и методы расчета необходимых объемов ресурсов, выделяемых на организацию финансовой мотивации (в разрезе каждой должности);

b) должны быть улучшены методы анализа результатов использования стратегии финансовой мотивации;

c) коэффициенты должны быть подвержены автоматическому анализу в соответствии с изменением объема ресурсов, выделенных на организацию финансовой мотивации;

d) процесс корректировки коэффициентов должен быть максимально ускорен.

Условные надбавки, которые призваны повысить заинтересованность сотрудников, имеют часто субъективный характер, так как назначаются начальниками подразделений [18].

В условиях расширяющейся технологии управления персоналом на удаленной работе возникает необходимость проведения объективного анализа деятельности сотрудников. Большинство компаний предпочитают классические схемы работы, однако расширяется и аудитория последователей применения новых схем оценки качества труда и его производительности и расчета на этом основании заработной платы сотрудников.

1.2 Модели оплаты труда сотрудников

Наиболее популярной до сих пор является модель расчета заработной платы в зависимости от занимаемой должности (тарифного разряда), а также объемов выполненной работы. Расчет заработной платы сотрудников предполагает предварительную обработку информации о сотрудниках предприятия, которая включает личную информацию о сотруднике, а также

необходимые данные для расчета заработной платы. Обязательной является информация о занимаемой должности и соответствующем тарифном разряде. Соответствие занимаемым должностям определяет документ штатное расписание [9].

Формирование штатного расписания и ведение личного досье персонала системы управления производственной организации является достаточно трудоемкой задачей уже даже для среднего предприятия.

Штатное расписание – чрезвычайно важный документ, который имеется в каждой компании, осуществляющей найм работников. В нем указываются должностные обязанности сотрудника. Информация в трудовую книжку вносится в соответствии с информацией, которая указана в штатном расписании. Оно оформляется по особой унифицированной форме.

Решение поставленной задачи сотрудниками отдела кадров или службой по работе с персоналом может отводиться разным должностям в зависимости от комплектации отдела и размеров предприятия [15].

Составление штатного расписания проводится менеджерами по работе с кадрами, однако при необходимости в его составление подключается главный бухгалтер и дополнительно сотрудники планово-экономического отдела, юридического отдела и т.п. Утверждение штатного расписания относится к полномочиям руководителя организации. Документооборот, связанный с поставленной задачей, затрагивает все аспекты деятельности предприятия.

Для формирования штатного расписания и ведения личных досье персонала необходимо решение следующих задач. Внесение общей информации по предприятию, создание справочников по должностям на предприятии, определяющим основной кадровый контур. Для каждой должности определяется круг компетенций, расположение в структуре управления организации, включая подчинения и управление.

Кроме этого, включается информация, связанная с требованиями к работнику, видами оплаты труда и вариантами ставок, а также графиком

работы. Требования, предъявляемые к работнику, например, раздел «Должен знать», определяются типовой инструкцией, основанной на тарифно-квалификационных характеристиках.

В случае, если сотрудник работает по временной форме оплаты труда, расчет осуществляется по определенным законодательством алгоритмам относительно повременной оплаты труда.

Из-за слабого стимулирующего действия повременная оплата труда имеет незначительную область применения, например, использование повременной оплаты работников в бригадах с оплатой по единому наряду.

При ведении повременной оплаты труда необходимо соблюдение следующих требований:

- ведение строгого учета за фактически отработанным количеством часов каждым работником с отображением времени простоя;
- корректное распределение повременных тарифов согласно квалификации и реальной сложности выполняемых работ;
- разработка и обоснование норм обслуживания, нормирование заданий и численности по каждой категории работников;
- оптимизация организации труда на каждом рабочем месте для эффективного использования рабочего времени.

Расчет заработной платы сотрудников, работающих в сдельной системе оплаты труда, осуществляется по следующей формуле (1):

$$ЗП_{пов} = С \cdot В \quad (1)$$

где С – тарифная ставка работника определенного разряда квалификации в установленную единицу времени (как правило, час), руб.;

Ф – фактически отработанное сотрудником время (как правило, в часах).

Расчет производится с использованием тарифного разряда работника. Как правило, это происходит по тому, что выполняемая работа слишком

разнообразна, чтобы как-либо классифицировать ее по физическим или моральным затратам на выполнение, необходимости наличия опыта или специальных знаний.

Основным документом, подтверждающим факт отработки сотрудником конкретного количества часов или дней, является табель, который ведется и подписывается непосредственным руководителем сотрудника (бригадиры, руководители группы, супервайзеры, мастера и т.д.). Заработная плата может устанавливаться за отработанный месяц (оклад), день или час. Затем для почасовой или подневной форме оплаты труда умножается количество отработанных дней (часов) на ставку оплаты за один день (час). В случае же с окладом высчитывается процент отработанных дней и умножается на месячный оклад.

Корректировка для сотрудников, находящихся на конкретной тарифной ставке, производится исключительно в виде надбавок на основании дополнительных условий, связанных с наличием опыта работы, отмеченными достижениями.

При сдельной или иной системе оплаты труда корректировка связана с объемом выполненных работ или же, в случае использования новой системы планирования, от полученного конечного дохода компании.

1.3 Особенности грейдовой системы оплаты труда и мотивации

Квалификационные характеристики, являющиеся основными параметрами для расчёта заработной платы, формируются только соответственно в результате оценки профессиональной и трудовой ценности выполненных работ. Коэффициенты межквалификационных соотношений формируются исходя из результатов оценки ценности должностей специально отобранными профессионалами – экспертами.

Одним из нестандартных инструментов, связанных с расчетом заработной платы на основании определения ценности должности, является система грейдов.

Система грейдов нацелена на решение следующего комплекса задач:

– определение меры ценности каждой должности относительно общего вклада в структуру стратегического развития компании;

– построение более объективной и оптимизированной оплаты труда;

– оценивание деятельности сотрудника с учетом занимаемой должности в компании;

– получение серьезных регулируемых инструментов финансовой мотивации персонала.

Несмотря на особенности работы по принципу выделения грейдов конечная система оплаты может являться комплексной и включать разбиение на тарифные разряды и работу по грейдам в зависимости от необходимости регулирования и типов выполняемых работ.

С одной стороны, одним из обязательных условий назначения работника на определенную должность является оценка его квалификации, которая проводится с использованием справочника квалификационных характеристик профессий (СКХП). Однако без этих элементов невозможно построение и грейдовой системы.

С другой стороны, основой обоих систем является вилковый принцип, который предполагает установку не точных коэффициентов для расчета заработной платы, а выбор их из некоторого заданного интервала на основании решения задачи оптимизации общего объема заработной платы или же точечного «ручного» распределения.

Построение грейдовой системы оплаты можно определить шагами следующего алгоритма:

1) описание должностей, которые будут рассматриваться в общей грейдовой системе;

2) оценивание каждой должности с использованием выбранной методики оценивания;

3) непосредственная разработка системы грейдов;

4) задание размера оплаты труда согласно проведенному делению ценности должностей для организации;

5) введение грейдовой системы.

С точки зрения адекватности использования выработанной технологии определения оплаты труда всегда после этапа внедрения возникает вопрос о частичной корректировке в зависимости от изменившихся условий работы должностей или ситуации на рынке.

Процесс описания должностей связан с их изучением, поэтому могут применяться следующие методики:

– анкетирование;

– интервьюирование;

– наблюдение.

В ходе интервьюирования происходит получение описания должности на основании понимания ее самими работниками.

Анкетирование позволяет достаточно быстро, часто в автоматизированном режиме получать необходимые сведения и проводить предварительный анализ полученных данных.

Вариант наблюдения за процессом работы предполагает присутствие аналитика на рабочем месте сотрудника и получение данных о выполняемых действиях путем непосредственного наблюдения.

Каждый из этих методов несет определенную практическую ценность при формировании представления о должности у наблюдателя. Прямые анонимные ответы сотрудников, осознанные рассказы о должности и ежедневные алгоритмы действий – все это позволяет сложить целостную картину.

Алгоритм описания должности графически представлен на Рисунке 2.

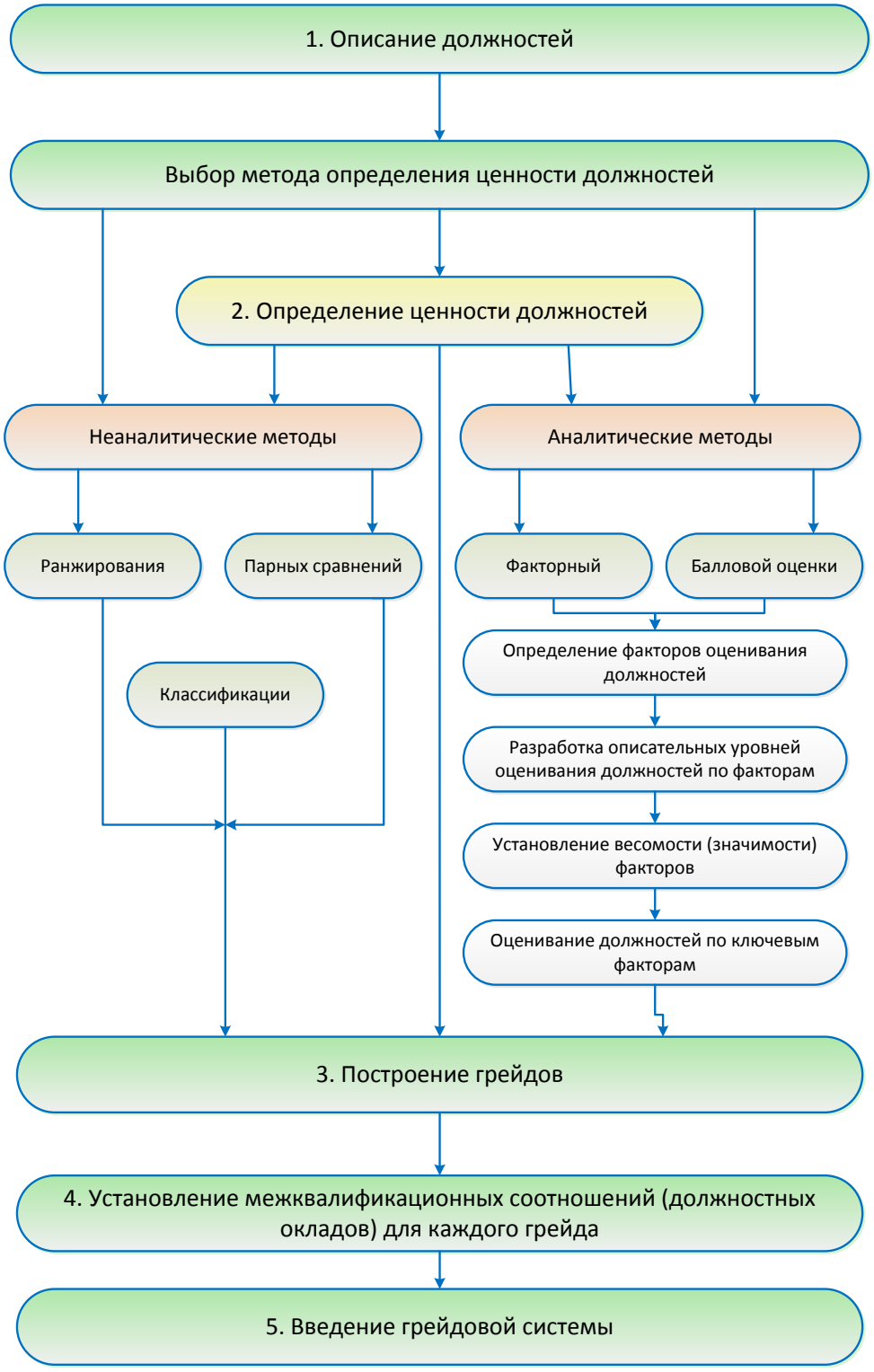


Рисунок 2 – Описание этапов разработки грейдовой системы оплаты труда

Ценность должности может быть определена с использованием нескольких методик, однако все методики условно можно разделить на:

– аналитические;

– неаналитические.

Использование неаналитических методик предполагает применение в ходе оценивания знаний экспертов, в качестве которых могут выступать:

– опытные специалисты, выполняющие работу на оцениваемых должностях;

– руководители подразделений;

– специалисты стратегического планирования.

Метод ранжирования носит максимально субъективный характер. Суть его заключается в том, что специальная комиссия, состоящая из наиболее авторитетных профессионалов, руководителей подразделений (среднего и высшего уровней) распределяет должности в порядке уменьшения их пользы и практической ценности для организации. Ввиду субъективности таких оценок результаты зачастую довольно сильно расходятся в значениях, и для их обработки и корректировки используются статистические методы.

Несколько иной подход предлагает метод классификации. Его суть заключается в группировке сотрудников по некоторым выделенным характеристикам, например:

– категория персонала (руководители, профессионалы, специалисты, технические служащие, рабочие)

– уровень управления (руководители высшего, среднего и низшего звеньев);

– квалификационная категория (ведущий, первой или второй категории, без категории) и т. д.

В завершение должности необходимо распределить по грейдам (грейды определяются только в пределах выделенной ранее группы).Недостаток заключается в том, что оценки экспертов снова не лишены некоторой субъективности, однако метод классификации может быть вспомогательным для метода ранжирования.

Существует также вариант оценивания не всех должностей сразу, а каждой должности с каждой другой. В таком случае распределение баллов производится согласно следующим условиям:

– должность, обладающая большей ценностью, получает один балл;

– должность, обладающая меньшей ценностью, не получает баллов;

– должность сама с собой не сравнивается.

После этого необходимо определить сумму для каждой должности (Таблица 1). Провести попарное сравнение эксперту значительно проще, чем производить ранжирование целого ряда должностей.

Таблица 1 – Пример матрицы попарного сравнения

Должность	**Менеджер проекта**	**Системный аналитик**	**Веб разработчик**	**Тестировщик**	**Программист**
Менеджер проекта		0	0	0	0
Системный аналитик	1		1	0	0
Веб разработчик	1	1		0	0
Тестировщик	1	0	1		0
Программист	1	1	1	1	
Сумма	4	2	3	1	0
Ранг	1	3	2	4	5

Среди **аналитических методов** выделяются факторный метод и метод баллового оценивания.

Способы деления на грейды могут быть самыми разными. Если пользоваться неаналитическими методами, то они формируются при помощи рангов, которые были разработаны для каждой из должностей. Ранги же делятся на диапазоны, которые определяются руководителями и специалистами по системе оплаты труда, используя их субъективное мнение.

Также используется деление должностей на такие группы, которые применяются на данном предприятии (Таблица 2).

Таблица 2 – Сформированные грейды согласно рангам

Категория	**Название должности**	**Ранг должности**	**Грейд**
Руководители высшего звена	Генеральный директор	12	4-й
	Руководитель филиала	11	4-й
	Проектный менеджер	11	4-й
	Операционный менеджер	11	4-й
Руководители среднего звена	Руководитель группы	10	3-й
	Координатор проекта	10	3-й
Руководители низшего звена	Супервайзер	9	2-й
	Сотрудник отдела контроля качества	8	2-й
Профессионалы	Сотрудник службы мониторинга	7	1-й
	Старший оператор	6	1-й
	Оператор логистической поддержки	5	1-й
	Оператор	4	1-й

По результатам формирования грейдов устанавливаются диапазоны для оценки основной заработной платы и технологий расчета дополнительной заработной платы. Грейд становится группой, для которой и устанавливается диапазон, при этом в грейд могут входить разного уровня специалисты в случае, если для данной стратегии развития компании они вносят одинаковый вклад с точки зрения оценки экспертов.

Установка вилок заработной платы как диапазона расчета предполагает дифференциацию, которую можно проводить и внутри грейда. Такое разделение позволяет организовать деление по специалистам.

Расчет параметров диапазона для расчета оклада производится с учетом оценочных данных по рынку труда для подобных специалистов. Технология использования рыночных данных может быть организована несколькими способами:

– поднятие нижнего параметра диапазона вилки до среднерыночного значения оклада для подобных специалистов, а верхний параметр поднимается на определенный процент от этого значения;

– среднерыночная «цена» специалиста такого типа становится средней в диапазоне, определенном для грейда, а верхняя граница также поднимается на определенный процент от среднерыночной.

Такие технологии дают возможность градировать оклады в зависимости от возможностей компании на данный момент с помощью выбора процента превышения выплат над среднерыночными.

Выбор размера выплачиваемой заработной платы, таким образом, должен быть привязан к рыночной стоимости подобного труда. Однако необходимо производить оценку и внутренней конъюнктуры, которая создает микроклимат в коллективе, не допустить возникновения у остальных сотрудников условного понятия «внутренней справедливости».

Разработка диапазонов может осуществляться разными методами:

– назначение диапазона для каждого грейда в отдельности;

– использование квалификационных коэффициентов, которые связывают грейдовую заработную плату с минимальной оплатой труда в компании на основании пропорциональности.

В компаниях предпочтение обычно отдается первой методике, однако второй метод скорее отвечает интересам компании и при необходимости изменения ситуации на рынке возникает справедливое пропорциональное изменение уровня зарплат для всех сотрудников.

Ключевыми характеристиками диапазонов являются:

– соотношение между средним коэффициентом наиболее низкого грейда и средним коэффициентом наивысшего грейда;

– тип роста средних коэффициентов в диапазоне;

– ширина диапазона — разность между максимальным и минимальным коэффициентами в диапазоне;

– перекрытие в диапазоне.

Выводы по главе

В первой главе были рассмотрены проблемы мотивации персонала. Были изучены существующие модели труда, рассмотрены их сильные и слабые стороны с мотивационной и правовой точек зрения. По результатам исследования было принято решение остановиться на использовании грейдовой системы оплаты труда.

Для данной системы оплаты труда были рассмотрены различные аналитические и неаналитические методы оценивания ценности должностей. Были рассмотрены варианты использования рыночных данных для решения поставленных задач (поднятие нижнего параметра диапазоны вилки до среднерыночного значения оклада или же назначение среднерыночной цены специалиста средней в диапазоне грейда).

Также проведен анализ существующих методик разработки диапазонов: назначение диапазона для каждого грейда в отдельности или же использования коэффициентов, связывающих каждую группу сотрудников с минимальной оплатой труда в компании на основании пропорциональности).

2 Алгоритмы, используемые для ранжирования

2.1 Стандартные методики классификации объектов

Главной задачей алгоритма кластеризации является создание кластеров, которые обладают свойством однородности внутри и имеют существенные отличия от объектов, входящих в другие кластеры. Множества, определяющие кластеры, строятся на основе анализа самих данных или векторов характеристик объектов, содержащих наборы данных. При этом элементы, образующие эти множества, должны одновременно быть максимально похожими на себе подобных и в значительной мере отличаться от множеств других кластеров.

По сути, основная идея кластерного анализа – выявление близости групп объектов, предполагающее «естественное» разбиение всей совокупности на компактные скопления элементов, поэтому ключевым понятием кластерного анализа становится определение этого «расстояния», которое в теории множеств называется метрикой. Данные, используемые для проведения кластерного анализа, сразу могут быть представлены в виде матрицы расстояний в случае использования количественных, а не качественных характеристик. Однако если возникает необходимость задания такого расстояния (метрики), то ее выбор может кардинальным образом изменить результаты кластерного анализа.

Общая формулировка задачи кластерного анализа не зависит от рассматриваемой предметной области и в большинстве классических работ имеет следующий вид [17].

Пусть задано некоторое множество $I = \{I_1, I_2,..., I_n\}$ состоящее из nобъектов. При этом j-ая характеристика объекта I_jопределена как x_{kj}, а набор характеристик объекта j определяется вектором $X_j = [x_{ij}]$. Следовательно, множеству объектов $I = \{I_1, I_2,..., I_n\}$ ставится в соответствие множество измерений $X = \{X_1, X_2,..., X_n\}$, полностью описывающее актуальные характеристики объектов в данной предметной области.

Введенные множества с точки зрения Евклидова пространства порождают некоторую область.

Целью проведения кластеризации является выделение некоторого числа групп на указанном множестве $I = \{I_1, I_2,..., I_n\}$ с учетом имеющихся характеристик объектов $X = \{X_1, X_2,..., X_n\}$, при условии, что число групп $m, m < n$. Основным условием является непересекаемость созданных групп $\pi = \{\pi_1, \pi_2,..., \pi_m\}$.

Решение задачи кластерного анализа может быть представлено как поиск некоторого разбиения, обладающего свойством оптимальности. Евклидово пространство позволяет задавать оптимальный критерий на основе определения целевого значения функционала, выражающего уровни возможного построения разбиения и группировки. Таким образом, задача кластерного анализа может быть сформулирована как поиск оптимального решения при наличии набора ограничений, т.е. фактически поиск условного экстремума функционала. В качестве функционала может выступать сумма квадратов отклонений по всем кластерам, и тогда решение задачи сводится к минимизации среднеквадратичного отклонения.

Следовательно, если имеется N измерений, то характеристики объектов $X = \{X_1, X_2,..., X_n\}$ составляют матрицу (2):

$$X = \{X_1, X_2,..., X_n\} = \begin{bmatrix} x_{11} & x_{12} & ... & x_{1N} \\ x_{21} & x_{22} & .. & x_{2N} \\ .. & ... & ... & ... \\ x_{n1} & x_{2n} & ... & x_{nN} \end{bmatrix} \quad (2)$$

Для каждой пары векторов заданного пространства может быть определено расстояние $d(X_i, X_j)$(3):

$$\Delta = \begin{bmatrix} 0 & d_{12} & ... & d_{1N} \\ d_{21} & 0 & .. & d_{2N} \\ .. & ... & ... & ... \\ d_{n1} & d_{2n} & ... & 0 \end{bmatrix} \quad (3)$$

где $d(X_i, X_i) = d_{ii} = 0\ i = \overline{1, n}$.

Очевидно, что расстояние между характеристиками можно определять только в случае использования одинаковых единиц измерения, поэтому при необходимости может быть проведена нормировка (4):

$$z1 = \frac{x - \bar{x}}{\sigma}, z2 = \frac{x}{\bar{x}}, z3 = \frac{x}{x_{\max}}, z4 = \frac{x - \bar{x}}{x_{\max} - x_{\min}} \quad (4)$$

Если для количественных признаков наиболее часто используют расстояния, то близость качественных свойств проще определять при помощи понятия меры. Мера задает близость между объектами и определяется как функция, удовлетворяющая условиям (5)

$$\begin{gathered} 0 \le \mu(X_i, X_j) \le 1 \\ \mu(X_i, X_i) = 1 \\ \mu(X_i, X_j) = \mu(X_{\text{ij}}, X_i) \end{gathered} \quad (5)$$

Тогда матрица близости имеет вид (6):

$$\mu = \begin{bmatrix} 1 & \mu_{12} & \dots & \mu_{1N} \\ \mu_{21} & 1 & .. & \mu_{2N} \\ .. & \dots & \dots & \dots \\ \mu_{n1} & \mu_{2n} & \dots & 1 \end{bmatrix} \quad (6)$$

где $\mu(X_i, X_i) = \mu_{\text{ii}} = 1\ i = \overline{1, n}.$

Примером линейной меры может служить функция определяющая коэффициент корреляции, расстояние может быть задано различным образом с соблюдением свойств определения расстояния для метрического пространства (7) - (9):

$d(X_i, X_j) = \sum_{k=1}^{N} \left| x_{\text{ki}} - x_{\text{kj}} \right|$ – линейное расстояние (7)

$d(X_i, X_j) = \left[\sum_{k=1}^{N} (x_{\text{ki}} - x_{\text{kj}})^2 \right]^{1/2}$ - евклидово расстояние (8)

$d(X_i, X_j) = \left[\sum_{k=1}^{N} (x_{\text{ki}} - x_{\text{kj}})^p \right]^{1/p}$ - метрика Минковского (9)

В рамках реальной прикладной задачи матрицы расстояний Δ или близостей μ могут быть получены в результате проведения экспертных оценок или осуществления прямых измерений, например, число взаимных ссылок авторов, набор ключевых слов на странице, предпочтения

посетителей сайтов, рассчитанные путем оценки числа возвращений на определенные страницы и т.д.

Общая схема проведения кластерного анализа может быть представлена в виде схемы (Рисунок 3).

Рисунок 3 – Последовательность проведения кластерного анализа [21]

Сам процесс проведения кластерного анализа можно представить в виде следующей последовательности шагов:

Шаг 1. Построение основного кластеризируемого множества.

Шаг 2. Выделение свойств, на основе которых будет проводиться оценка близости объектов в группах.

Шаг 3. Нормализация количественных характеристик и задание значений качественным характеристикам.

Шаг 4. Выбор формулы задания метрики. Расчет элементов матрицы расстояний или близости.

Шаг 5. Использование методики кластерного анализа с целью формирования групп с использованием алгоритмов кластеризации.

Шаг 6. Представление результатов анализа.

После оценки адекватности полученных результатов может быть осуществлено возвращение к шагу 4 и смена метрики.

2.2 Технологии машинного обучения

Для деления по классам необходимо организовать работу механизмов, которые отвечают за деление на группы по определенным признакам. На входе мы задаем набор необходимых нам признаков, на выходе получаем сформированные механизмами группы.

Так как область применения очень разнообразна, то возникает необходимость выделения каких-либо одинаковых единообразных свойств,

которые используются в поставленной задаче. Деление на группы при помощи машинного обучения может происходить как опираясь на их предпочтения (т.е. сам пользователь делает свой выбор), так и оценку деятельности пользователей (например, сотрудников на основании анализа результатов их деятельности).

То есть нам необходимо подобрать такой признак, который наиболее точно поможет поделить наши объекты на группы.

Характеристики тех признаков, которые мы используем, должны отражать суть объекта при использовании каждой конкретной задачи. Необходимо преобразовать качественные оценки в количественные для того, чтобы использовать разнообразные метрики и нормы в многомерных пространствах. Мы можем провести оценку рисков и ожидаемых потерь при использовании данной классификации.

Необходимо найти что-то индивидуальное, что будет характеризовать все элементы нашей группы при помощи выбранных признаков.

Системы машинного обучения ищет такой феномен, отслеживая принадлежность объекта данной группе, используя наши заданные признаки на входе. Процесс обучения – это поиск зависимости между близкими объектами. В результате получаем деление на группы и некоторую метку, которая будет присваиваться новым объектам исследования при причислении их к выбранной группе на основании их похожести.

Целью машинного обучения является изучение методов построения алгоритмов, которые будут способны обучаться. Обычно выделяют два типа обучения:

– по прецедентам (индуктивное обучение), основой которого являются эмпирические данные и выявленные на их основе общие закономерности;

– дедуктивное обучение, продуцируемое экспертами и организованное в виде базы знаний.

Обычно используют обучение по прецедентам, а дедуктивное обучение – это скорее экспертная система. Машинное обучение – это смесь математической статистики, технологии эффективности вычислений и методов оптимизации. Машинное обучение - это не только теоретическая дисциплина, ее часто используют для решения инженерных и других практических задач, используя данные об экспериментах, которые проводились ранее или были смоделированы.

Рассмотрим самые распространенные алгоритмы машинного обучения.

Алгоритм обучения с учителем «случайный лес» используется для задач классификации и восстановления регрессии. Суть метода в том, что обучение нескольких веток дерева ведется параллельно и независимо. После этого в результате голосования ветвей дерева складывается конечный результат.

У алгоритма имеются практические преимущества: данные с огромным объемом признаков и категорий эффективно обрабатываются, достигается крайне высокая точность предсказания класса. Алгоритм легко масштабируется, ввиду большого количества ветвей легко поддается распараллеливанию на несколько потоков. Также алгоритм сохраняет все преимущества стандартных деревьев решений: исходные данные не нужно предварительно обрабатывать, работать можно как с категориальными, так и с вещественными признаками. Отсутствующие значения также не являются проблемой – алгоритм поддерживает работу с ними.

Для построения дерева решения используется алгоритм CART. Этим алгоритмом выполняется построение разбиения пространства признаков на непересекающиеся области за счет его рекурсивного разбиения. Иными словами, каждому из узлов дерева ставится в соответствие определенная область пространства признаков вместе с правилом, согласно которому производится ее разделение на две области.

При обучении модели случайного леса из начальной выборки выделяется случайная подвыборка, для которой строится дерево решений,

такое, что переменная в каждом из новых узлов дерева выбирается из случайного подмножества признаков.

Алгоритм обучения с учителем «машина опорных векторов» используется для решения задач классификации с большим количеством классов и восстановления регрессии. В процессе обучения алгоритма осуществляется построение линейного порогового классификатора $\mathrm{sgn}\left(\sum w_j \cdot x_j - w_0\right)$, где уравнение $\langle w, x\rangle = w_0$ служит для описания гиперплоскости, которая разделяет классы. Сущность метода состоит в оптимальном (с точки зрения определенного критерия) способе выбора w и w_0. При максимизации зазора ($m\mathrm{argin}$) между классами повышается адекватность классификации. В дальнейшем данный принцип был полностью теоретически обоснован.

Для возможности применения метода опорных векторов для задачи классификации с количеством классов $K > 2$ разработано две стратегии:

– стратегия «каждый против каждого». Выполняется обучение $\frac{K\cdot(K-1)}{2}$ различных моделей на всех доступных подзадачах, относящихся к бинарной классификации. Классификация нового объекта осуществляется с учетом всех построенных моделей, после чего делается выбор превалирующего класса;

– стратегия «один против всех». K моделей обучаются на задачах бинарной классификации, имеющих тип «один против всех оставшихся». Выбор класса нового объекта производится в соответствии с максимальным значением $m\mathrm{argin}$.

Наиболее эффективным при ранжировании является метод обучения учителем. Алгоритмы кластеризации являются разновидностью так называемого «спонтанного обучения» (unsupervised learning), основной задачей которого является выявление структуры в рядах данных, на первый взгляд кажущихся случайными (или немаркированными).

В общем случае базой для подобного алгоритма является выявление степени сходства между элементами (например, между сотрудниками,

хорошо выполняющими свою работу) путем расчета их расстояния от других элементов в пространстве признаков (feature space). В виде признака в пространстве признаков может быть выбрано, например, количество эффективно выполненных проектов или средняя производительность труда сотрудников в определенном классе.

Размерность пространства признаков определяется именно количеством независимых признаков. В том случае, если какие-либо элементы достаточно «близки» друг к другу, они могут быть объединены в один кластер.

Разработано большое количество алгоритмов кластеризации. Одним из самых простых среди них можно считать алгоритм *k* средних (*k* - means), выполняющий разделение элементов на *k* кластеров.

Первоначально распределение элементов по этим кластерам осуществляется в произвольном порядке. После этого для каждого из кластеров производится вычисление центра масс (или просто центра) как функции его членов.

Далее осуществляется проверка расстояния каждого из членов кластера от центра данного кластера.

Если в соответствии с результатами этой проверки член окажется ближе к другому кластеру, то происходит его перемещение в данный кластер. Проверив все расстояния для всех членов необходимо вычислить заново центры кластеров.

После того, как достигнуто стабильное состояние (процесс очередной итерации не вызвал перемещения членов), совокупность считается кластеризованной требуемым образом и происходит остановка алгоритма [11].

Процесс вычисления расстояния между двумя объектами может быть достаточно трудоемким для визуализации. Для этого в наиболее наглядном варианте каждый член кластера рассматривается в виде многомерного вектора. Для данных векторов далее вычисляется Евклидово расстояние.

Евклидово расстояние является геометрическим расстоянием в многомерном пространстве. Для вычисления евклидова расстояния между точками x и y в n - мерном пространстве используется следующее выражение (10):

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}, \quad (10)$$

Следует отметить, что евклидово расстояние (как и его квадрат) вычисляется на основании исходных, а не стандартизованных данных.

Для небольших объемов информации алгоритмы классического типа подходят отлично. Однако при росте количества обрабатываемой информации могут проявляться проблемы в получении корректных результатов на приемлемом уровне качества. При оффлайновой обработке данная трудность не может быть отнесена к критичной. Поэтому при выполнении сценариев реального времени требуются принципиально иные, специализированные подходы.

2.3 Учет общих рисков и ожидаемых потерь

Проблема оценки точности используемого классификатора при выделении групп объектов сводится к оценке:

– точности проведенной классификации;

– логарифмических потерь;

– площади ROC кривой;

– матицы неточностей.

Указанные оценки могут применяться не для всех используемых классификаторов, однако наиболее часто используется предварительная оценка точности классификации.

Точность классификации может быть рассчитана классическим способом в виде относительной частоты появления корректного прогноза. Значение точности может быть определено в виде частного:

– числитель – число прогнозов в выборке по результатам соответствующее реалиям предметной области;

– знаменатель – общее число проведенных прогнозов.

Оценка данной характеристики не всегда в полной мере отражает корректность кластеризации и чаще всего используется в случае равновеликих классов.

К тому же не всегда в реальной прикладной задаче можно без мнения эксперта оценить точность выдаваемой классификации, в таких случаях возможно применение иных методов оценки.

Например, логарифмические потери определяют вероятность принадлежности исследуемого объекта к выбранному классу. Само значение логарифмической потери определяется на интервале (0;1), где:

– близость к единице оценивается как уверенность в сделанном прогнозе принадлежности к классу;

– близость к нулю выражает сомнение в выполненном прогнозе.

Функция логарифмических потерь определяется через штрафную функцию при определении некорректной классификации.

В случае оценки для одного класса можно принять классификацию как испытания Бернулли, в этом случае (11):

$$a_i = a(x_i|w) \text{ - ответ алгоритма на -м объекте} \quad (11)$$

$$p(y|X,w) = \prod_i p(y_i|x,w) = \prod_i a_i^{y_i}(1-a_i)^{1-y_i} \to \max$$

$$\sum_i (-y_i \log a_i - 1(1-y_i)\log(1-a_i)) \to \min$$

Определяя вероятность попадания корректного объекта в класс как p, можно определить оптимальную оценку корректности определения класса как (12):

$$-p\log(a_i) - (1-p)\log(1-a_i) \quad (12)$$

$$a_i = p$$

Минимизирующий фукционал (13):

$$-p\log p - (1-p)\log p \quad (13)$$

Для сведения указанной функции к оценке логарифмических потерь необходимо определять логарифм по основанию 2, тогда доверительный интервал при сбалансированной выборке включается в отрезок [0;1].

При использовании нескольких классов получается следующая ситуация. Пусть есть некоторые N выборок, тогда штрафная функция логарифмических потерь для M классов определяется как (14):

$$\text{LogLoss} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{\text{ij}} \log(p_{\text{ij}}), \quad (14)$$

где

y_{ij} - определяет принадлежность выборки i классу j;

p_{ij} - определяет вероятность принадлежности выборки i классу j.

В случае использования бинарной классификации потери могут быть оценены с помощью функции, определяющей площадь ROC-кривой (AUC).

Предварительно рассмотрим разделение множества на два класса, при этом определяемая оценка принадлежности расположена на зарытом интервале [0;1]. Визуализация корректности такого алгоритма может быть произведена при помощи ROC-кривой (ROC = receiver operating characteristic, иначе – «кривая ошибок»).

Качество может быть ценено на основании расчёта площади под этой кривой– AUC (AUC = area under the curve).

ROC-кривая строится в виде ломаной на прямоугольном поле площади $m \times n$, где:

– m - число корректных определений класса, в этом случае используется метка 1;

– n - число некорректных определений класса, в этом случае используется метка 0.

ROC-кривая выходит из начала координат и дальше происходит движение по правилу (Рисунок 4):

– правильное определение класса – вверх на 1;

– неправильное определение класса на один вправо.

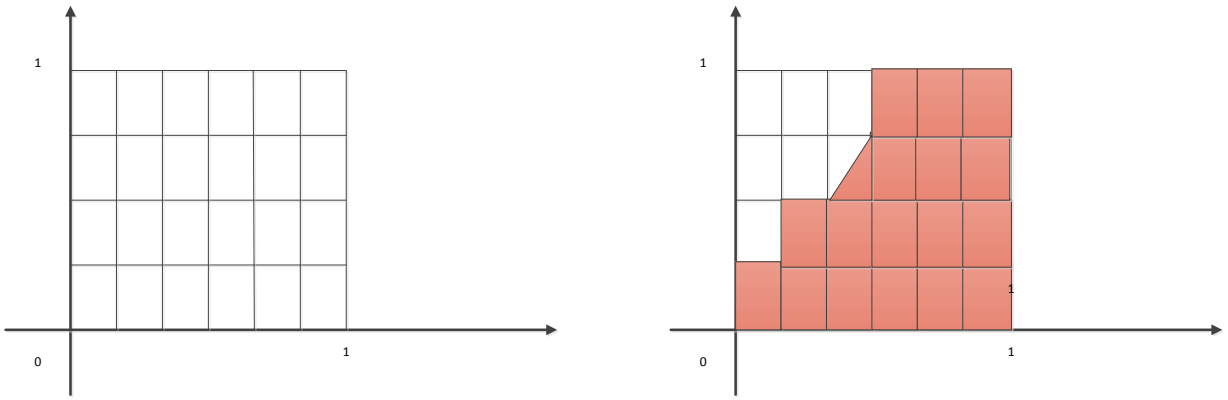


Рисунок 4 – Вариант представления ROC-кривой

Площадь под образованной кривой позволяет оценить уровень точности проведенной классификации.

Определение точности проводимой классификации может фиксироваться с использованием метрик:

- точность (precision);
- полнота (recall).

Или же применяются производные этих метрик в виде F-меры или производной метрики R-Precision.

Точность характеризует долю действительно относящихся к конкретному классу объектов, которые в результате кластеризации были отнесены верно.

Полнота характеризует долю найденных объектов конкретного класса по отношению ко всем существующим.

Представление таких параметров производится при помощи матрицы контингентности, составляемой для каждого класса индивидуально (Рисунок 5).

Категория		Экспертная оценка	
		Положительная	Отрицательная
Оценка системы	Положительная	TP	FP
	Отрицательная	FN	TN

Рисунок 5 – Определение матрицы контингентности

Заполнение таблицы контингентности осуществляется после расчета всех значений элементов:

- TP — истинно-положительное решение;
- TN — истинно-отрицательное решение;
- FP — ложно-положительное решение;
- FN — ложно-отрицательное решение.

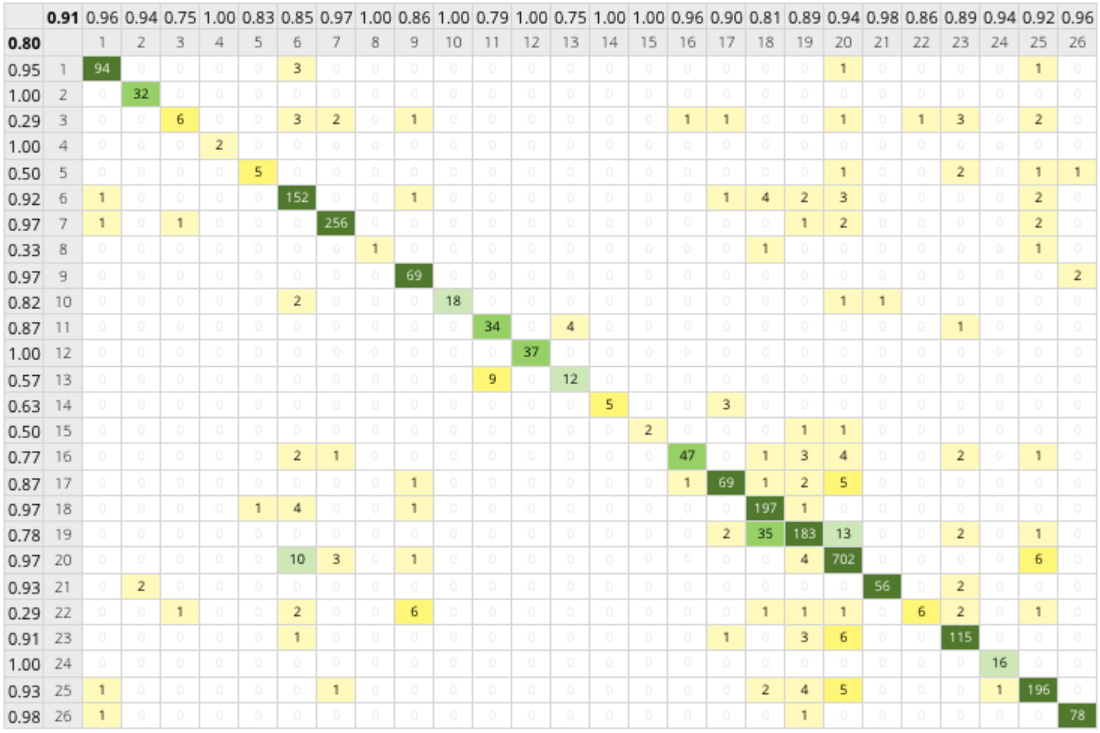
На основании рассчитанных коэффициентов матрицы определяются точность и полнота (15):

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{15}$$

Показатели точности и полноты могут быть представлены различными способами. Так, для небольшого количества классов наиболее простой, удобной и наглядной моделью является матрица неточности.

Матрица неточностей – является квадратной матрицей размера N на N, где N - общее число классов. Сравнение проходит по всем решениям, которые имеют экспертную оценку. Столбцы этой матрицы резервируются за экспертными решениями, а строки за решениями классификатора. Заполнение матрицы производится по результатам анализа (Рисунок 6).

Таким образом, матрица неточностей позволяет как достаточно точно с учетом статистики, так и наглядно, визуально, оценить степень неточности реализуемого классификатора.



	0.91	0.96	0.94	0.75	1.00	0.83	0.85	0.97	1.00	0.86	1.00	0.79	1.00	0.75	1.00	1.00	0.96	0.90	0.81	0.89	0.94	0.98	0.86	0.89	0.94	0.92	0.96
0.80		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
0.95	1	94	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0
1.00	2	0	32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.29	3	0	0	6	0	0	3	2	0	1	0	0	0	0	0	0	1	1	0	0	1	0	1	3	0	2	0
1.00	4	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.50	5	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	2	0	1	1
0.92	6	1	0	0	0	0	152	0	0	1	0	0	0	0	0	0	0	1	4	2	3	0	0	0	0	2	0
0.97	7	1	0	1	0	0	0	256	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	2	0
0.33	8	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0
0.97	9	0	0	0	0	0	0	0	0	69	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
0.82	10	0	0	0	0	0	2	0	0	0	18	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
0.87	11	0	0	0	0	0	0	0	0	0	0	34	0	4	0	0	0	0	0	0	0	0	0	1	0	0	0
1.00	12	0	0	0	0	0	0	0	0	0	0	0	37	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.57	13	0	0	0	0	0	0	0	0	0	0	9	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0
0.63	14	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	3	0	0	0	0	0	0	0	0	0
0.50	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	1	1	0	0	0	0	0	0
0.77	16	0	0	0	0	0	2	1	0	0	0	0	0	0	0	0	47	0	1	3	4	0	0	2	0	1	0
0.87	17	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	69	1	2	5	0	0	0	0	0	0
0.97	18	0	0	0	0	1	4	0	0	1	0	0	0	0	0	0	0	0	197	1	0	0	0	0	0	0	0
0.78	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	35	183	13	0	0	2	0	1	0
0.97	20	0	0	0	0	0	10	3	0	1	0	0	0	0	0	0	0	0	0	4	702	0	0	0	0	6	0
0.93	21	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	56	0	2	0	0	0
0.29	22	0	0	1	0	0	2	0	0	6	0	0	0	0	0	0	0	0	1	1	1	0	6	2	0	1	0
0.91	23	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	3	6	0	0	115	0	0	0
1.00	24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0
0.93	25	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2	4	5	0	0	0	1	196	0
0.98	26	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	78

Рисунок 6 – Пример заполнения матрицы неточности

Согласно представленным данным примера большинство объектов классификатор распределил верно (матрицу можно считать в основном диагональной). Однако есть и классы с низкой точностью, где большое число элементов вне главной диагонали.

Расчет точности и полноты производится достаточно просто по элементам матрицы, используя следующие формулы (16):

$$\text{Precision}_c = \frac{A_{c,c}}{\sum_{i=1}^{n} A_{c,i}} \qquad (16)$$

$$\text{Recall}_c = \frac{A_{c,c}}{\sum_{i=1}^{n} A_{i,c}},$$

Общая точность и полнота определяется как среднее арифметическое точности по всем определенным классам.

2.4 Выбор инструментов разработки системы

Выбор среды разработки связан со спектром решаемых задач и возможностей использования различных технологий реализации. Для проведения сравнительного анализа языков программирования в рамках выбранной предметной области выбираются следующие параметры (Таблица 3):

– наличие библиотек машинного обучения, которые позволяют не только организовать процесс обучения, но и провести оценки его точности;

– возможность построения сложного интерфейса, так как пользователи системы должны иметь возможности для построения сложных аналитических отчетов;

– простота использования, которая предполагает оценку сложности и громоздкости используемых технологий для реализации.

Таблица 3 – Сравнительный анализ языков программирования

Параметры	**Язык программирования**			
	Java	Visual C++ 12	Visual C# 4.0	Python
Возможность построения сложного графического интерфейса	Средняя	Средняя	Высокая	Низкая
Библиотеки машинного обучения	Spark MLlib, Weka,	Shogun, Azure Machine Learning, Microsoft Cognitive Toolkit, TensorFlow	Shogun, Azure Machine Learning	Shogun, TensorFlow, Theano, PyTorch, Scikit-learn, Keras
Простота использования	Низкая	Низкая	Высокая	Средняя
Наличие бесплатного программного обеспечения	NetBeans, Eclipse.	Только версия VS Community	Только версия VS Community	Eclipse+PyDev, Spyder

Указанные выше сравнительные характеристики (Таблица 3), позволяют говорить о том, что в рамках решаемой задачи возможность использования наибольшего числа библиотек и инструментов имеет язык Python.

TensorFlow – нейронная сеть, которая показывает отличные результаты благодаря тому, что обработка данных осуществляется не на одном, а сразу на нескольких уровнях, что способствует получению более достоверного результата.

Открытость исходного кода библиотеки машинного обучения TensorFlow в Google способствует упрощению процесса разработки и развертывания нейронных сетей повышенной сложности. У TensorFlow не предусмотрена возможность предоставления каждому из разработчиков прав пожинать плоды машинного обучения, однако данный инструмент делает возможным подключение к программе разработчика с помощью интерфейсов API для языков программирования $C/C++$ и Python.

Выводы по главе

Во второй главе были рассмотрены стандартные методики классификации объектов, определен и описан пошагово алгоритм кластеризации объектов.

Проведен анализ методов машинного обучения (как по прецедентам, так и индуктивных). Были рассмотрены отдельные алгоритмы машинного обучения (случайный лес, машина опорных векторов).

Также были оценены плюсы и минусы алгоритмов кластеризации, выделены наиболее подходящие для использования в разрабатываемой программе (алгоритм k-means, Байесовские цепи доверия, цепи Маркова, метод Роккио).

В качестве средства реализации после сравнения был выбран язык программирования Python. Предварительно были выделены его достоинства и недостатки, проведено сравнение с конкурентами (Java, Visual C++12, Visual C# 4.0).

3 Разработка системы классификации сотрудников

3.1 Формирование набора данных

Выделение грейдов с использование технологии машинного обучения, как описано ранее, предполагает определение ключевых признаков, которые кладут в основу анализа «похожести» объектов.

Выделение грейдов с этой точки зрения - это выбор близких по уровню и интенсивности работы сотрудников, оплата труда которых должна быть если не одинаковой, то быть определена в рамках некоторой «вилки». Суммарный фонд оплаты труда будет более «справедливо» распределен в случае выделения наибольших активов сотрудникам, вносящим максимальный вклад в получение дохода компанией и определение оплаты труда «не очень эффективных» сотрудников по остаточному принципу.

Грейдовая система оплаты предполагает более корректное разделение фонда заработной платы, и, следовательно, предполагает постоянную его корректировку на основании изменений, вызванных финансовой мотивацией сотрудников.

Таким образом, рассматриваемая технология классификации определяет подбор данных за определенный период:

– до месяца для малых компаний, в которых изменение дохода может происходить достаточно часто;

– квартал или иной период, выбранный в качестве промежутка реализации тактических планов.

Более мелкое изменение для крупных компаний может привести к несбалансированности бюджета, а также некорректной оценке результатов труда сотрудников. Обязательным является исключение из анализа или корректировка сезонных изменений и последних месяцев года, связанных с закрытием проектов на стратегическом уровне.

В качестве основных признаков, оказывающих влияние на общий вклад сотрудника в деятельность компании можно выделить следующие:

– уровень полученной квалификации на данный момент, определяемый экспертной оценкой или на основании квалификационного тестирования;

– средний уровень выполняемых и оконченных проектов, в которых сотрудник принимает непосредственное участие;

– производительность труда, рассчитанная на основании определенных задач в рамках выполненных проектов и проведённого план-факторного анализа по всем проектам, выполненным сотрудником за рассматриваемый период;

– сложность выполнимых работ, оценка которых проводится экспертами или на основании расчета используемых трудозатрат в зависимости от уровня работающего специалиста.

Параметр сравнения в виде квалификации не предполагает исключительно учет уровня образования или же опыта работы. Оценочный интервал для уровня квалификации определяется от уровня самого квалифицированного специалиста до уровня стажера.

В структуру оценки квалификации обязательно должны быть включены:

– уровень образования;

– опыт работы;

– оценка квалификационного тестирования;

– оценка экспертов, полученная на основании последней проведённой аттестации.

Указанные характеристики могут быть объединены с некоторыми весами, с учетом вида деятельности компании и расширены дополнительными категориями, такими как специализированные навыки, необходимые специалисту на рабочем месте:

– уровень использования информационных технологий;

– знание иностранных языков;

– применение современных методов управления и т.п.

Результатом оценки квалификации становиться взвешенное среднее полученных оценок.

При использовании проектного подхода оценка уровня исполняемого сотрудников проекта не составляет труда, как так определяется на основании статистического анализа уже закрытых проектов или экспертной оценки сложности проектов на основании сравнения с аналогами.

Расчет производительности труда осуществляется на основании определения следующих характеристик:

– числа решаемых задач за выбранный период по завершенным проектам;

– сложность решаемых задач, определяемая на основании оценки трудозатрат или сравнительного анализа стоимости подобных работ на рынке труда.

Общая сложность работы характеризуется мерой участия сотрудника во всех закрытых за период проектах компании.

Получая следующий набор характеристик, группа экспертов выделяет принадлежность конкретного сотрудника грейду:

– квалификация или уровень набора навыков, необходимых для выполнения должностных функций (skill);

– средний уровень закрытых за период проектов (projct_level);

– производительность труда за выбранный период (labor_capacity);

– сложность выполняемых работ в закрытых за период проектах (job_complexity).

Для предварительного обучения возможно проведение оценки несколькими экспертами, а далее определение конечной оценки в виде принадлежности грейду стандартной методикой, например, простым усреднением или методом иерархий.

В результате сформированный набор данных импортирован в табличный процессор MS Excel (Рисунок 7).

В качестве идентификатора для определения конкретного сотрудника выбирается его номер, присвоенный в кадровой службе.

	A	B	C	D	E	F	G	H
1	Number	FIO	skill	project_level	labor_capacity	job_complexity	grade	
2	1	Петров А.В.	6	10	10	7	3	
3	2	Иванов Г.А.	4	9	8	9	3	
4	3	Колпышев А.М.	6	9	3	2	2	
5	4	Заяров А.К.	2	5	8	8	2	
6	5	Туров Е.Б.	9	8	7	6	4	
7	6	Синеров А.А.	9	4	2	5	2	
8	7	Умаров К.Е.	9	4	10	1	3	
9	8	Багуев О.П.	8	10	8	4	3	
10	9	Парамонов А.В.	3	5	10	4	2	
11	10	Варнава А.С.	10	9	5	9	4	
12	11	Караваев В.П.	6	10	1	2	2	
13	12	Буров К.Е.	3	2	2	9	1	
14	13	Самсонов А.К.	8	10	8	6	3	
15	14	Буреев А.К.	2	7	9	3	1	
16	15	Загаров Е.К.	6	8	10	9	3	
17	16	Манаев А.В.	8	9	3	5	3	

Рисунок 7 – Пример сформированного файла предварительных данных

С целью использования полученных предварительных данных для обучения с использованием библиотеки Scikit-learn данные преобразованы к следующему виду (Рисунок 8).

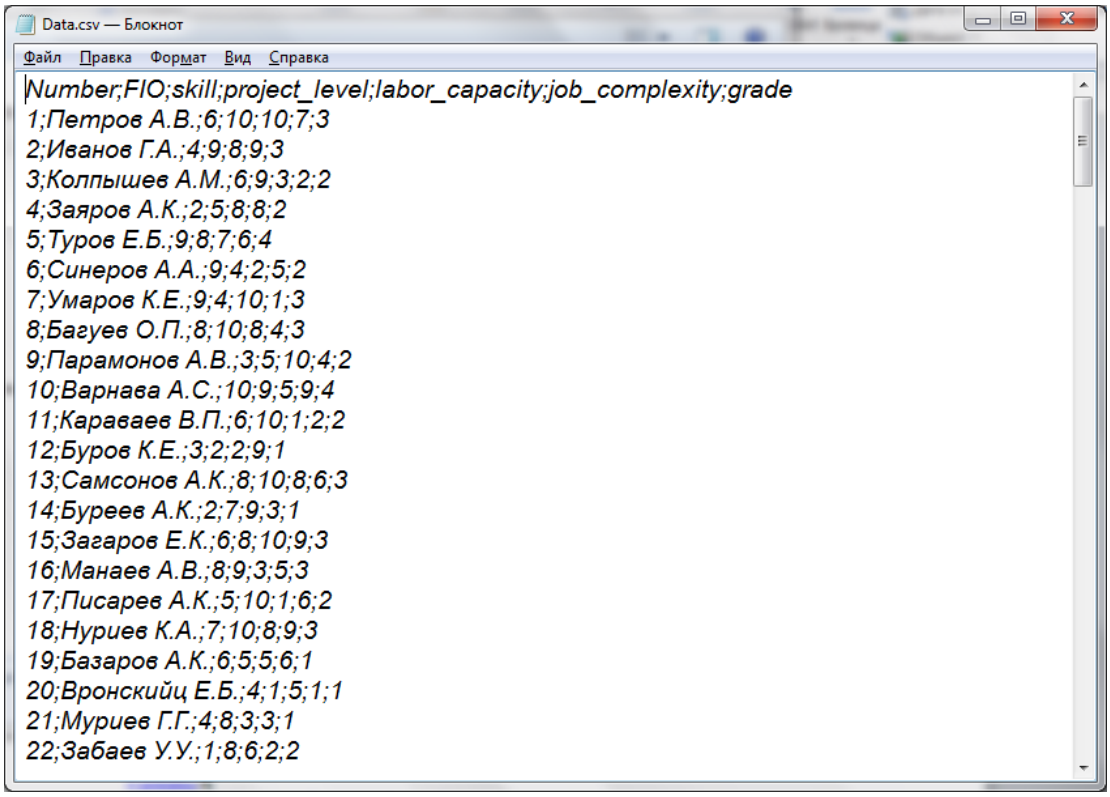


Рисунок 8 – Пример сформированного файла предварительных данных

3.2 Описание используемых библиотек

В ходе исследования применяются инструменты библиотеки Scikit-learn 0.24.1 [24], которая включает множество технологий машинного обучения как с учителем, так и без.

Scikit-Learn представляет собой библиотеку машинного обучения на языке программирования Python с открытым исходным кодом. Содержит реализации практически всех возможных преобразований [25].

Для организации работы с выбранной библиотекой необходимо подключение следующих библиотек для устойчивой работы:

– NumPy (математические операции и тензорное исчисление);

– SciPy (реализация научно-технических вычислений);

– Matplotlib (инструменты для визуализация данных);

– IPython (интерактивная консоль для Python);

– SymPy (символьная математика);

– Pandas (обработка, манипуляции и анализ данных).

Специализация библиотеки Scikit-Learn на алгоритмах машинного обучения предполагает реализацию исключительно механизмов решения следующих задач:

– обучение с учителем;

– обучение без учителя.

В рамках модели обучения с учителем используются следующие компоненты для поддержки:

– классификации (предсказание признака, множество допустимых значений которого ограничено);

– регрессии (предсказание признака с вещественными значениями).

Однако модель обучения без учителя также является необходимой. Она реализует такие важные компоненты, как:

– кластеризация (разбиение данных по классам, которые модель определит самостоятельно),

– понижение размерности (представление данных в пространстве меньшей размерности с минимальными потерями полезной информации);

– детектирование аномалий.

Методы, используемые для проведения классификации и кластеризации в библиотеке Scikit-Learn очень разнообразны: от линейных до сложных нейронных сетей.

Линейные модели предполагают ограничения для построения классификации в виде гиперплоскостей, которые разделяют или же аппроксимируют исследуемые данные.

Технологии оценки расстояний между объектами исследуемого множества используют метрические модели, в рамках которых применяются методы оценки (К ближайших соседей).

Задачи с большим числом условий могут быть адаптированы под модели в виде деревьев решений, в рамках которых определяется оптимальный вариант решения задачи.

Расширением моделей деревьев решений являются ансамблевые методы, в рамках которых комбинируются возможности нескольких методик. Такие модели также предполагают применение отбора признаков (бустинг, бэггинг, случайный лес, мажоритарное голосование).

Могут быть использованы также нейронные сети для решения регрессионных задач и задач классификации.

Из нелинейных методов выделяется:

– SVM – обучение определению границ принятия решений;

– вероятностное моделирование для задач классификации Наивный Байес;

– t-SNE – метод понижения размерности;

– K-средних – методика кластеризации, требущая предварительную оценку числа будущих кластеров для распределения данных;

– Кросс-валидация, использующая полную выборку без выделения тестовой, так как обучение проводится несколько раз в качестве обучающей выборки выступают части полного множества данных. Окончательный результата определяется усреднением.

Наиболее популярные методы, инструменты для использования которых встроены в библиотеку Scikit-Learn:

– k-ближайших соседей (K-Nearest Neighbors);

– опорных векторов (Support Vector Machines);

– классификатор дерева решений (Decision Tree Classifier) / Случайный лес (Random Forests);

– Наивный байесовский метод (Naive Bayes);

– линейный дискриминантный анализ (Linear Discriminant Analysis);

– логистическая регрессия (Logistic Regression).

Проведение классификации при помощи определения К-ближайших соседей предполагает использование алгоритмов поиска кратчайшего расстояния между уже распределенными по множествам объектами в ходе использования обучающего набора и исследуемым объектом (Рисунок 9).

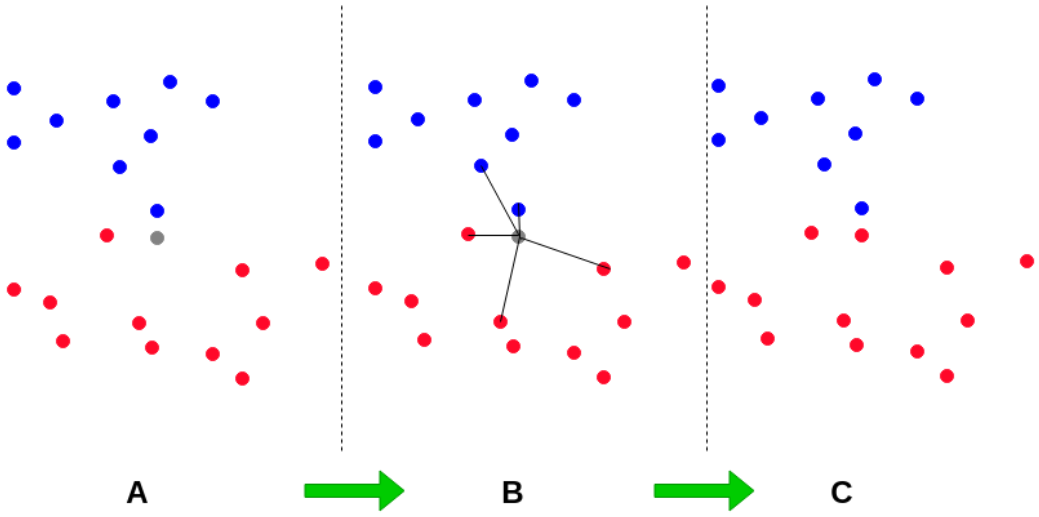


Рисунок 9 – Визуализация метода К-ближайших соседей

Классификация с использованием дерева решений предполагает последовательное разбиение множества на подмножества путем применения различных критериев, при этом каждое множество может обладать своей сортирующей категорией.

Продолжение деления уменьшает число элементов подмножества. Окончание деления проходит при создании групп из одного элемента. Объединение нескольких таких деревьев определяет случайный лес (Рисунок 10).

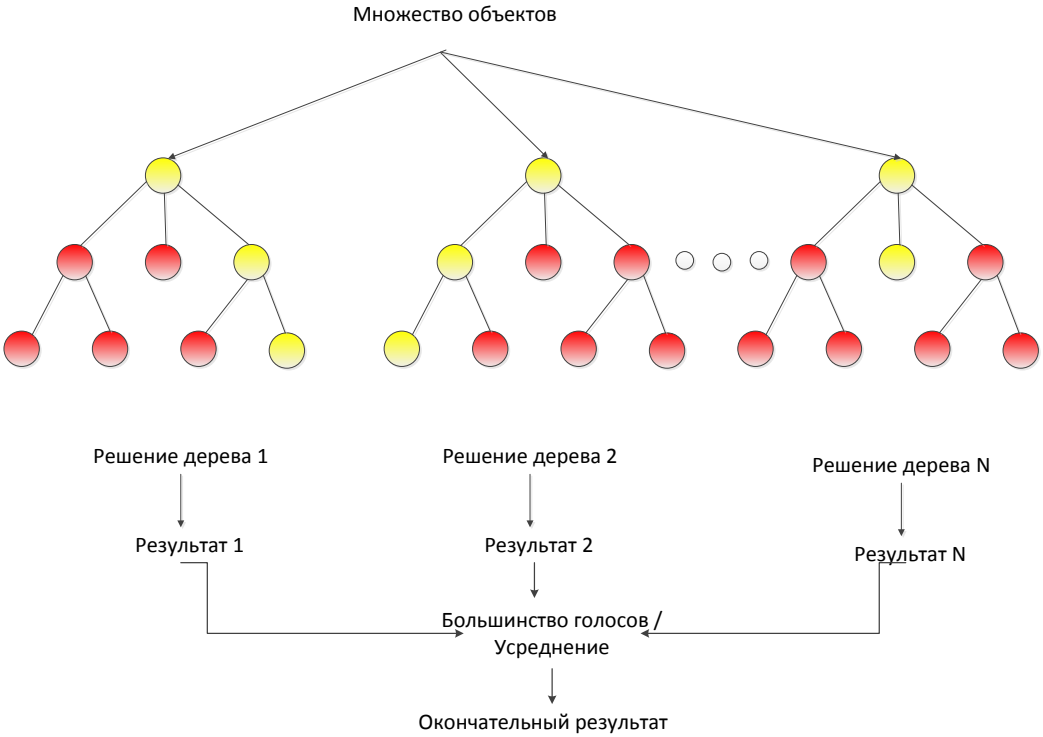


Рисунок 10 – Классификатор дерева решений

Метод с использованием Naive Bayes основывается на принципах расчета вероятности попадания объекта в подмножество, с учетом уже известных вероятностей наступления других событий. В ходе анализа все используемые параметры объекта считаются независимыми, и наступление события одновременного выполнения нескольких параметров предполагает оценку произведения вероятностей наступления отдельных событий.

Уменьшение размерности набора данных для сведения решения задачи к более простой реализуется при помощи линейного дискриминантного

анализа. Переход к меньшей размерности проводит путем проецирования на линию и комбинации точек в класс с учетом удаления от центральной точки.

Метод опорных векторов (SVM) также использует расстояния, однако выделяет целую линию разделения между подмножествами, выступающих в виде разных классов.

Цель классификатора состоит в увеличении расстояния межу подмножествами. В результате образованные разделенные подмножества и становятся классами (Рисунок 11).

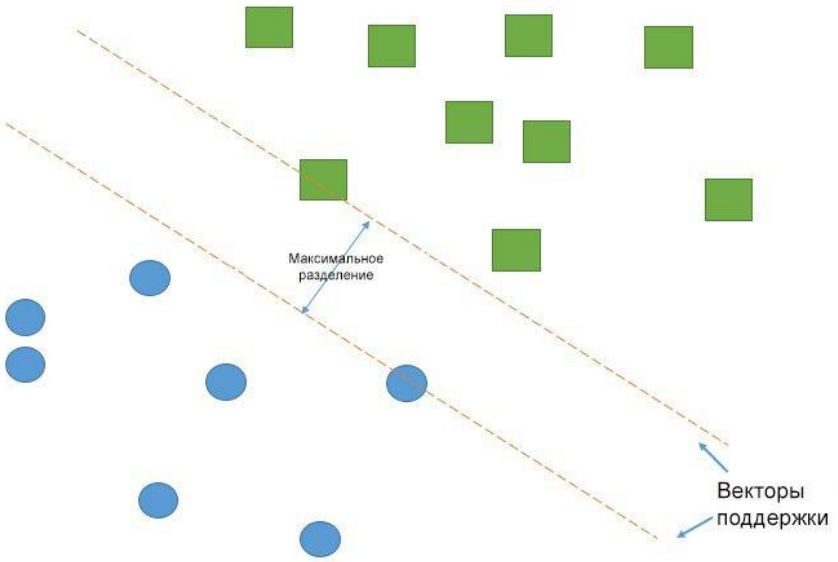


Рисунок 11 – Метод опорных векторов

3.3 Реализация проектного решения

Разработанное приложение для проведения сравнения выделенных алгоритмов построено с использованием графического интерфейса средствами языка Python (библиотека Tkinter).

Главное окно приложения включает главное меню, которое является основной структурой управления и включает несколько подменю: исходные данные, создание грейдов, сравнение эффективности по ROC и кноку «Выход» (Рисунок 12).

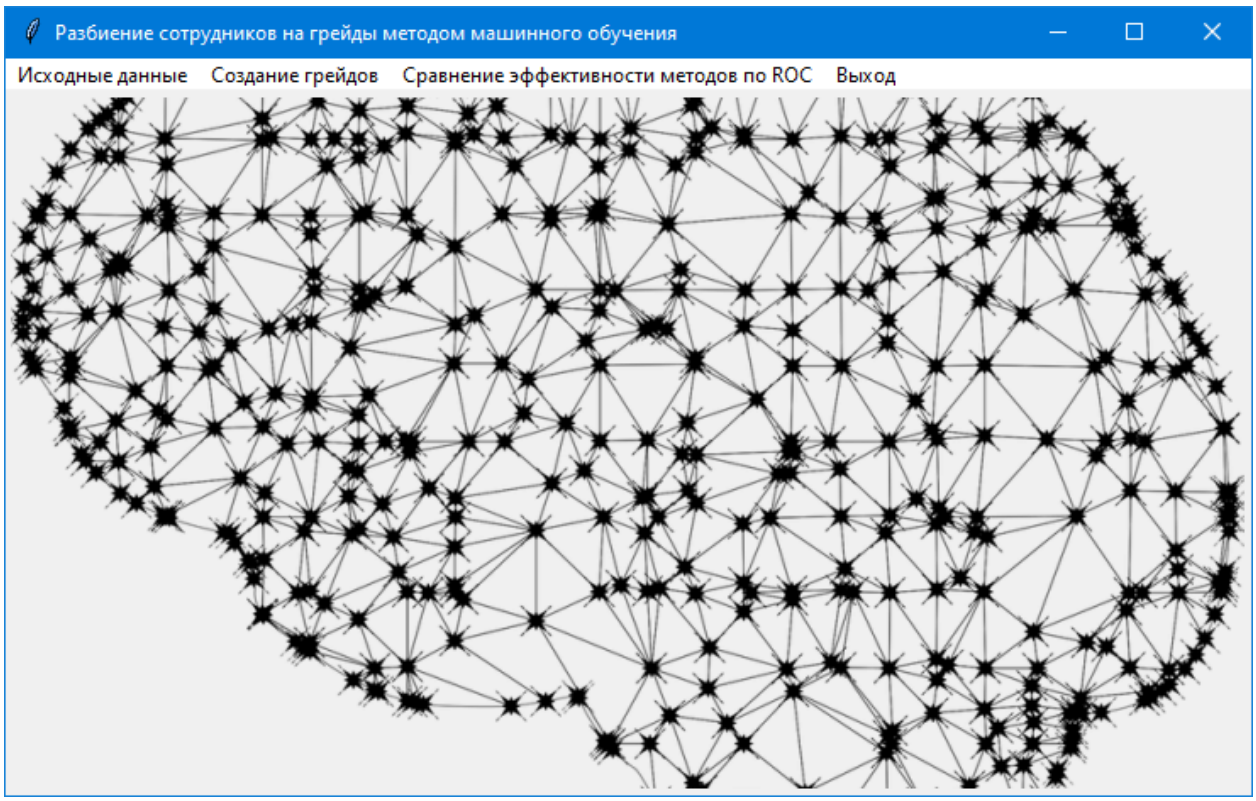


Рисунок 12 – Пример главного окна приложения

Рисунок 13 представляет подменю для обзора данных и получения результатов обучения, которое выпадает при клике по кнопке «Исходные данные».

Подменю включает следующие элементы:

– Просмотр данных – отображение данных, используемых в качестве выборки для обучения и деления на грейды. ;

– Результаты обучения – отображение разделения на грейды по результатам обучения выбранными методами и оценки их эффективности.

Результат обучения фиксируется и выводится в консоль (Рисунок 14).

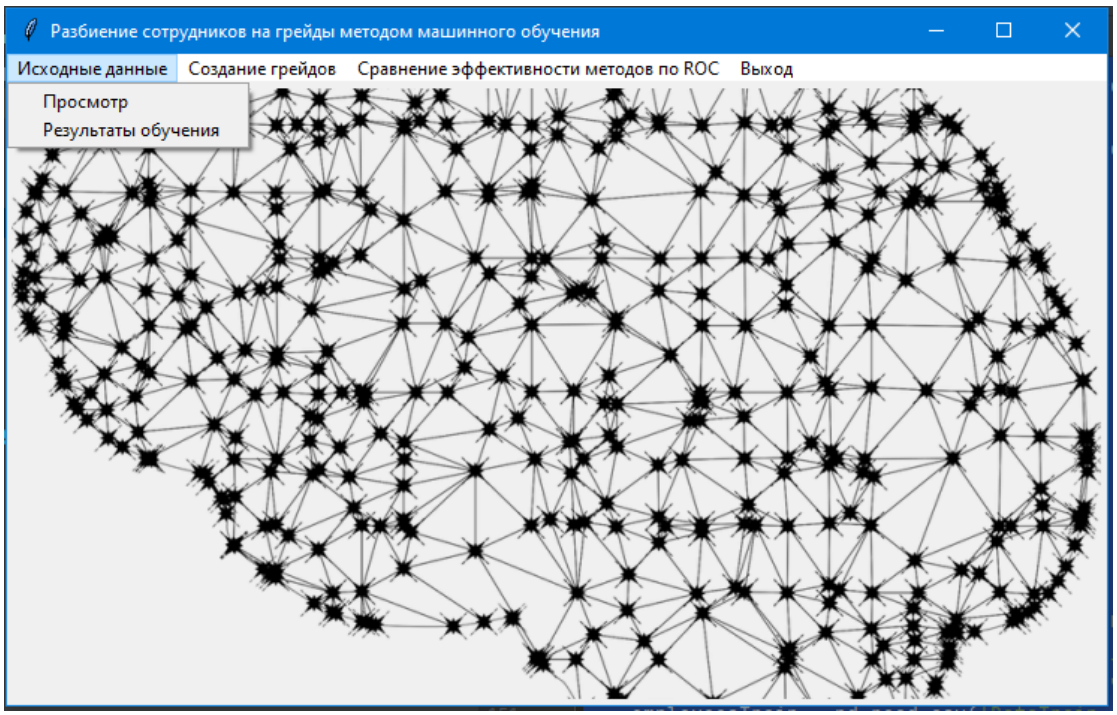


Рисунок 13 – Варианты проведения тестирования нейронной сети с использованием различных алгоритмов оптимизации

```
E:\Python_Project\Grades\venv\Scripts\python.exe E:/Python_Project/Grades/DataProcessing.py
Оценка точности классификаторов


Метод опорных векторов

0.55
Метод ближайших соседей

0.525
Случайный лес

0.675

Матрицы неточности классификаторов


Метод опорных векторов

[[8 1 0 0]
 [8 9 0 0]
 [0 2 2 1]
 [2 2 2 3]]
Метод ближайших соседей

[[11  2  1  0]
 [ 5  6  0  0]
 [ 0  2  1  1]
 [ 2  4  2  3]]
Случайный лес
```

Рисунок 14 – Пример отображения результатов оценки проведенного обучения

Рисунок 15 демонстрирует варианты по выбираемым алгоритмам обучения для создания грейдов:

– Метод опорных векторов;

– Метод ближайших соседей;

– Случайный лес.

При переключении на соответствующий пункт подменю производится выполнение выбранного действия.

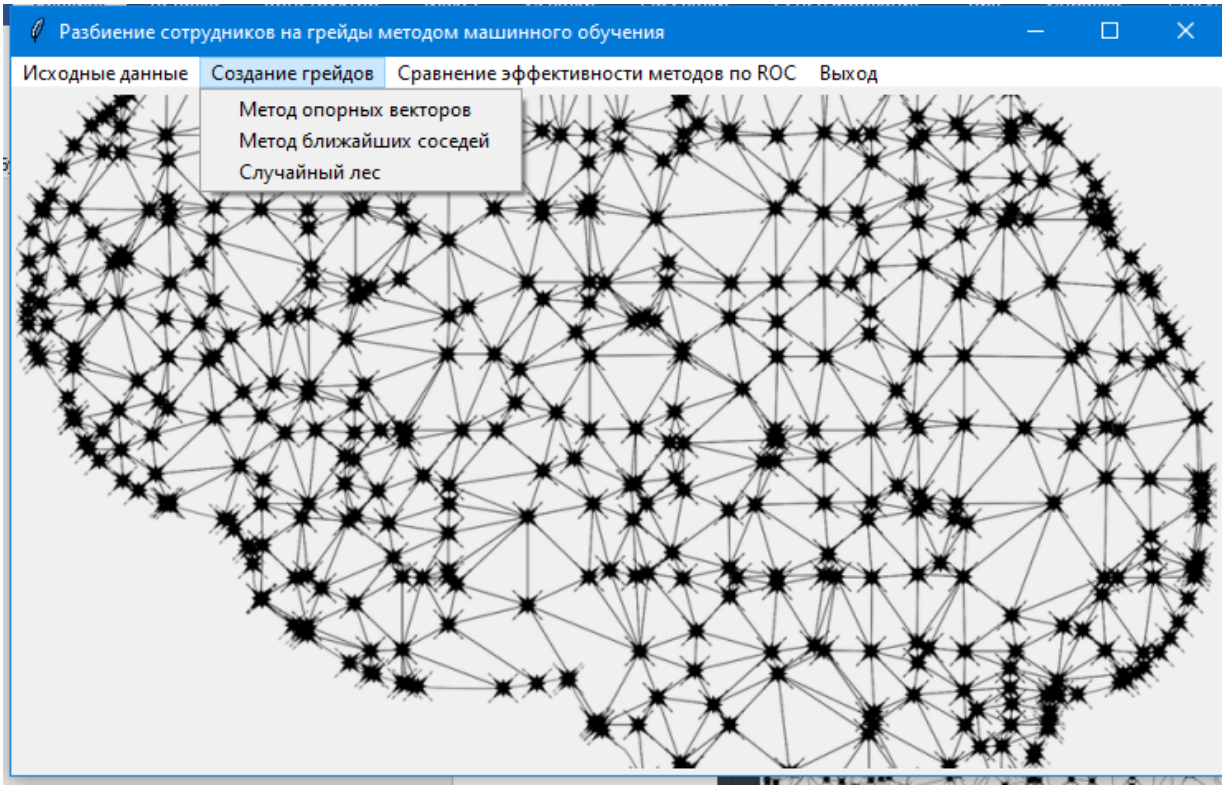


Рисунок 15 – Вариант использования меню для создания грейдов по методам обучения

Код программы, реализующей классификацию сотрудников, представлен в приложении А.

Фрагмент кода, реализующий графический интерфейс , представлен ниже.

```
root = Tk()
    root.title("Разбиение сотрудников на грейды методом машинного обучения")
    mainmenu = Menu(root)
    root.config(menu=mainmenu)
    sourcemenu = Menu(mainmenu, tearoff=0)
    sourcemenu.add_command(label="Просмотр", command=sourcedata)
    sourcemenu.add_command(label="Результаты обучения",command=view_quantity)
    grademenu = Menu(mainmenu, tearoff=0)
```

```
    grademenu.add_command(label="Метод опорных векторов", command=view_SVC)
    grademenu.add_command(label="Метод ближайших соседей", command=view_KNN)
    grademenu.add_command(label="Случайный лес", command=view_RFC)
    mainmenu.add_cascade(label="Исходные данные",menu=sourcemenu)
    mainmenu.add_cascade(label="Создание грейдов",menu=grademenu)
    mainmenu.add_command(label="Сравнение эффективности методов по ROC",
command=comparemethods)
    mainmenu.add_command(label="Выход", command=root.destroy)
    # Вывод картинки
    # создание рабочей области
    frame = Frame(root)
    frame.grid()
    # добавление изображения
    canvas = Canvas(root, height=410, width=730)
    img = PhotoImage(file='wall.png')
    image = canvas.create_image(0, 0, anchor='nw', image=img)
    canvas.grid(row=1, column=1)
    root.mainloop()

if __name__ == '__main__':
    main()
```

Кроме того, в программе реализован многооконный интерфейс. Для вывода дочернего окна использована следующая функция.

```
# Просмотр исходных данных
def sourcedata():
    data = pd.read_csv('DataSource.csv')
    sw=tk.Toplevel(root)
    sw.title('Исходные данные для машинного обучения')
    sw.text = data.columns.values
   # sw.parent = root
    sw.tree = ttk.Treeview(sw, columns=sw.text[1:])
    sw.vsb = tk.Scrollbar(sw, orient="vertical", command=sw.tree.yview)
    sw.tree.configure(yscrollcommand=sw.vsb.set)

    sw.vsb.pack(side="right", fill="y")
    for i, j in enumerate(sw.text):
        sw.tree.heading(f"#{i}", text=j)
    for i in range(len(data[sw.text[0]])):
        sw.tree.insert('', 'end', text=data[sw.text[0]][i],
values=list(map(lambda x: data[x][i], sw.text[1:])))

    sw.tree.pack()
```

Целесообразно кратко остановиться на ключевых шагах реализации поставленной задачи классификации.

Реализация любого классификатора начинается с его импорта в Python. Для представленных в программе методов (метод опорных векторов, метод ближайших соседей, случайный лес) данная операция выглядит следующим образом:

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
```

Следует отметить, что представленными классификаторами их перечень в библиотеке Scikit-Learn далеко не ограничивается. С остальными классификаторами можно ознакомиться в соответствующей документации по данной библиотеке.

Далее осуществляется создание экземпляра классификатора. Для этого создается соответствующая переменная и вызывается функция, которая связана с классификатором. Пример подобного действия представлен ниже:

```
# Модели
SVC_model = SVC()
RFC_model = RandomForestClassifier(n_estimators = 70) # случайный лес (в
параметре передается кол-во деревьев)
KNN_model = KNeighborsClassifier(n_neighbors = 18) # модель ближайших соседей
(параметр - число соседей)
```

После этого открыт путь к обучению классификатора. До начала такой процедуры выполняется его «адаптация» в соответствии с обучающими данными.

Для помещения признаков и меток, участвующих в обучении, в классификатор предназначена функция fit. В реализованной программе обращение к ней выглядит следующим образом:

```
SVC_model.fit(PrArgs_Train,PrRes_Train)
KNN_model.fit(PrArgs_Train,PrRes_Train)
RFC_model.fit(PrArgs_Train,PrRes_Train)
```

Данные модели, прошедшие обучение, пригодны для подачи в классификатор. Для этого существует соответствующая функция классификатора, как predict. В нее передается в качестве параметра соответствующий признак, принимающий участие в прогнозировании.

```
SVC_model.predict(PrArgs_Res)
KNN_model.predict(PrArgs_Res)
RFC_model.predict(PrArgs_Res)
```

Вышеописанные этапы (создание экземпляра, обучение и, как итог, классификация), могут быть отнесены к ключевым при работе с классификаторами в библиотеке Scikit-Learn. Вместе с тем, в данной библиотеке предусмотрена возможность управления не только классификаторами, но и, непосредственно, данными. Для более четкого понимания принципа совместного взаимодействия данных и классификатора в процессе решения задачи классификации целесообразно уделить

повышенное внимание процессам, происходящим в ходе машинного обучения в целом.

Весь процесс машинного обучения состоит из следующих этапов: подготовка данных, формирование наборов данных для обучения, создание классификатора, обучение классификатора, формирование прогноза в соответствии с полученными результатами классификации, а также последующая оценка производительности классификатора и настройка параметров, участвующих в классификации.

Прежде всего, необходимой является подготовка совокупности данных для классификатора. Под этим в широком смысле понимается преобразование данных в пригодную и удобную форму для классификации с одновременной обработкой любых выявленных аномалий в этих данных. К аномалиям относятся, например, отсутствие значений в данных, отклонения любого другого типа. Все они подразумевают необходимость обработки. Если этого не сделать, высока вероятность их отрицательного влияния на производительность соответствующего классификатора. Такой этап получил название «предварительная обработка данных» (data-preprocessing).

Следующий этап заключается в разделении данных на совокупности обучающего и тестового характера. Чтобы максимально упростить данный процесс для пользователя в библиотеке Scikit-Learn предусмотрена такая функция, как traintestsplit.

Как уже было упомянуто, для создания и обучения классификатора используется обучающая совокупность (набор) данных. Выполнив эти этапы, можно утверждать о способности модели к выдаче определенных прогнозов. Путем сравнения результатов работы классификатора с уже известными фактическими данными, делаются выводы, касающиеся точности предсказания классификатора.

В практических исследованиях часто возникает необходимость определенной «коррекции» параметров классификатора. Этот процесс обычно продолжается до достижения классификатором требуемой точности

(поскольку точное соответствие показателей классификатора требованиям пользователя сразу с первого раза представляется маловероятным).

3.4 Проведение эксперимента

После запуска приложения в среде PyCharm на экране появляется главное окно программы (Рисунок 16).

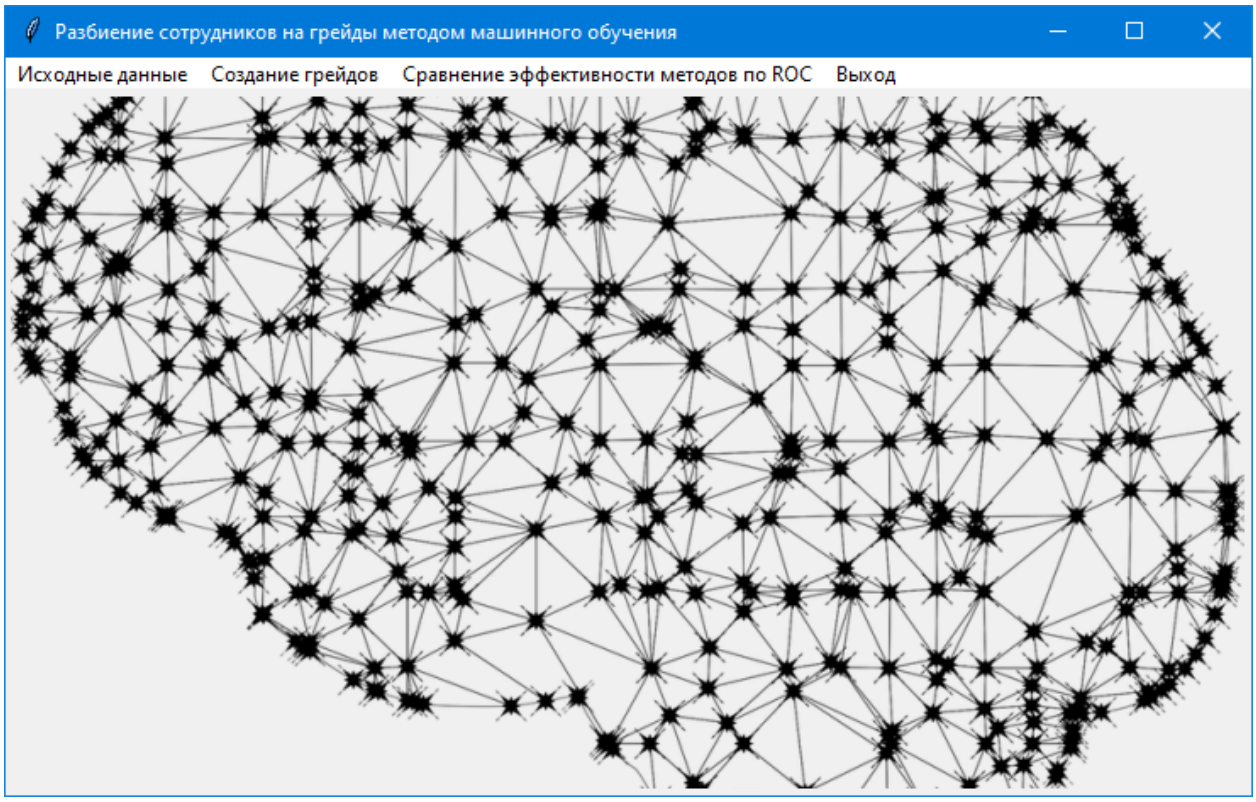
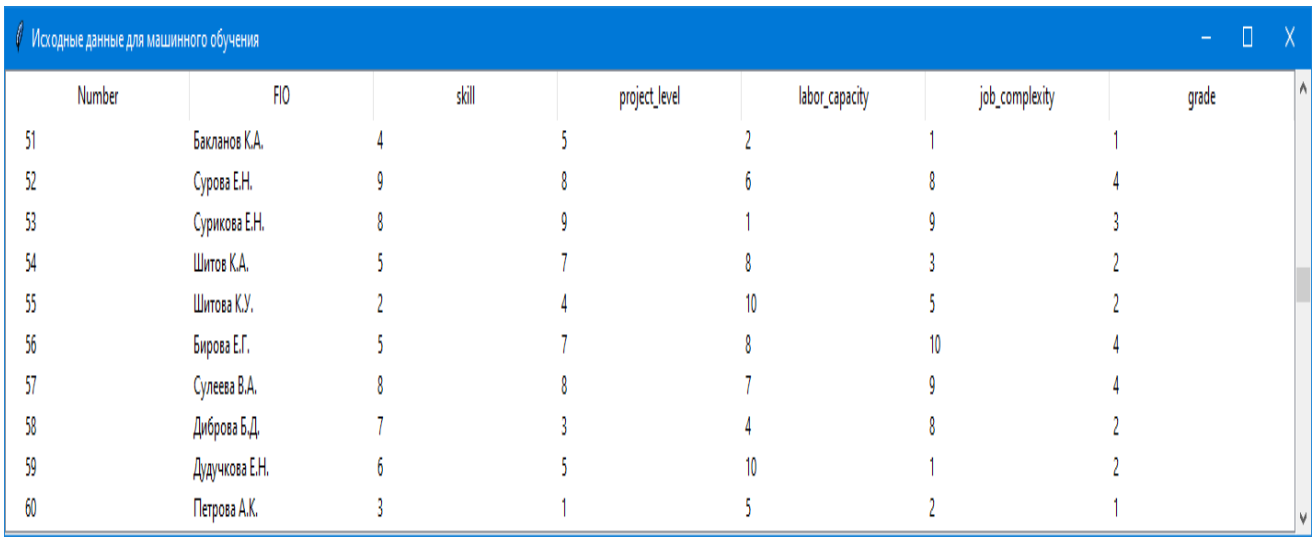


Рисунок 16 – Главное окно приложения

Пользователь может выбрать необходимое ему действие, выбрав соответствующий пункт в главном меню формы, расположенном в ее верхней части.

Выбор пункта меню «Исходные данные -> Просмотр» приводит к выдаче на экран набора данных, предназначенного для обучения классификатора (Рисунок 17).



Number	FIO	skill	project_level	labor_capacity	job_complexity	grade
51	Бакланов К.А.	4	5	2	1	1
52	Сурова Е.Н.	9	8	6	8	4
53	Сурикова Е.Н.	8	9	1	9	3
54	Шитов К.А.	5	7	8	3	2
55	Шитова К.У.	2	4	10	5	2
56	Бирова Е.Г.	5	7	8	10	4
57	Сулеева В.А.	8	8	7	9	4
58	Диброва Б.Д.	7	3	4	8	2
59	Дудучкова Е.Н.	6	5	10	1	2
60	Петрова А.К.	3	1	5	2	1

Рисунок 17 – Набор исходных данных для обучения

Так как обучение производится для трех методов (метод опорных векторов, метод ближайших соседей, случайный лес), то целесообразно оценить эффективность обучения каждого из методов. Для этого необходимо выбрать пункт меню «Сравнение эффективности методов по ROC», после чего на экран будет выведена гистограмма эффективности указанных методов (Рисунок 18).

После этого можно переходить к прогнозированию с помощью созданных моделей для каждого из методов.

Разбиение сотрудников по грейдам в соответствии с методом опорных векторов представлено на Рисунок 19.

Разбиение сотрудников по грейдам в соответствии с методом ближайших соседей представлено на Рисунок 20.

Разбиение сотрудников по грейдам в соответствии со «случайным лесом» представлено на Рисунок 21.

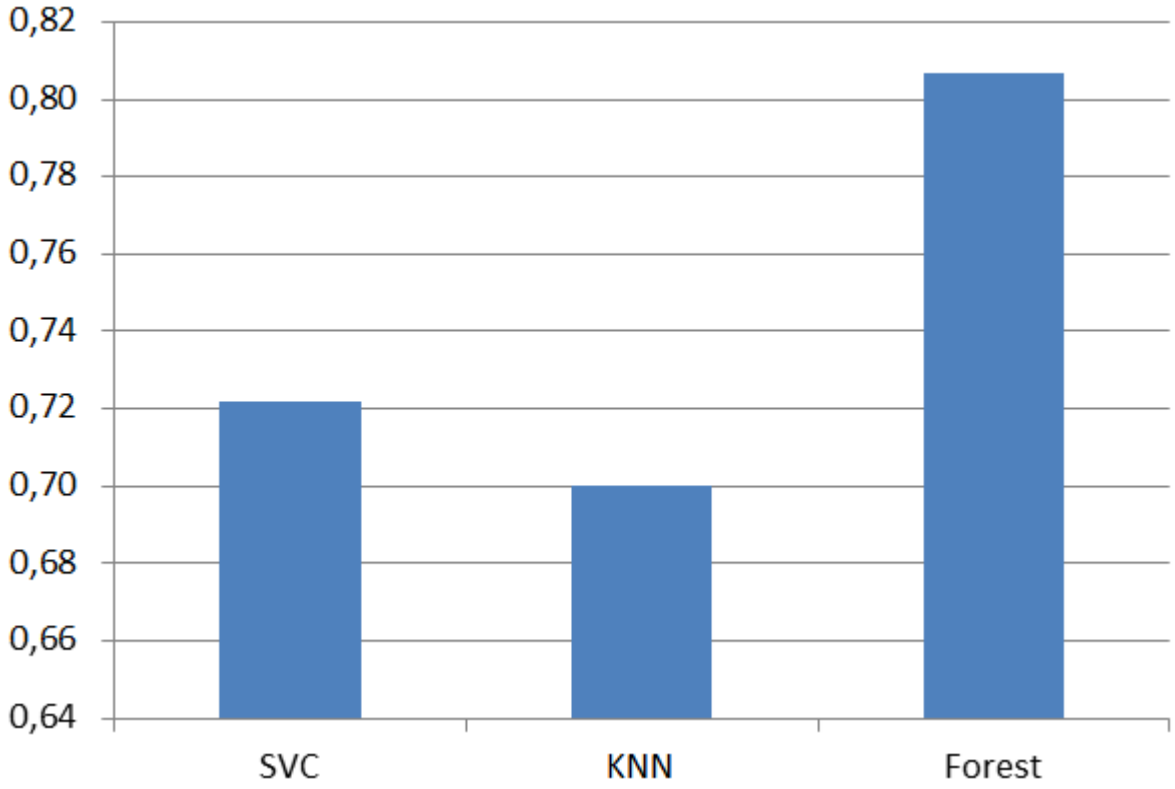
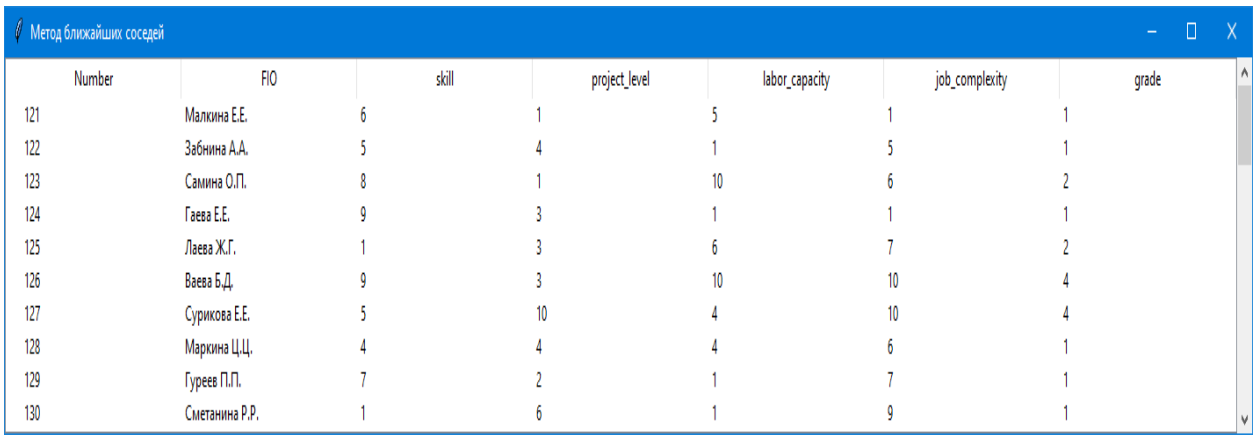


Рисунок 18 – Гистограмма эффективности методов машинного обучения

Метод опорных векторов

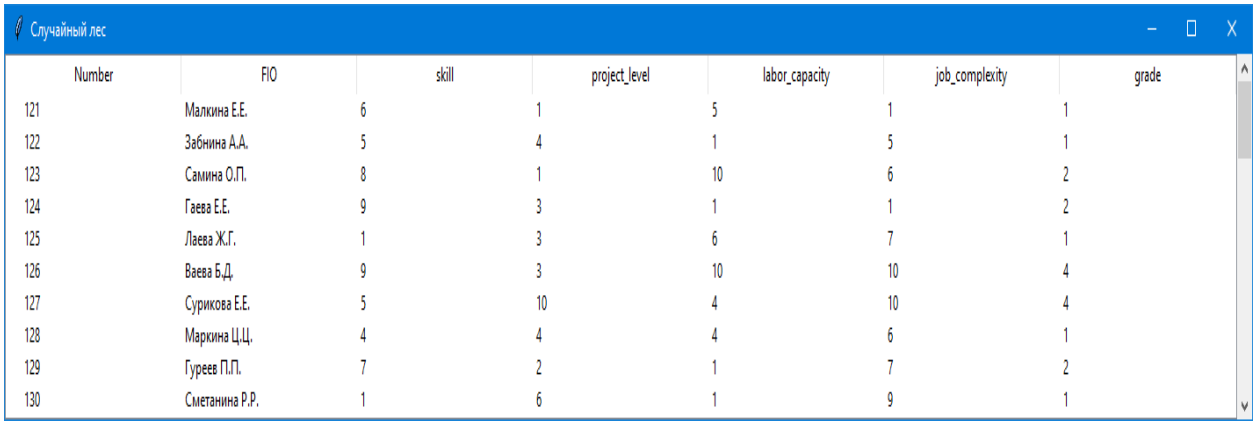
Number	FIO	skill	project_level	labor_capacity	job_complexity	grade
121	Малкина Е.Е.	6	1	5	1	1
122	Забнина А.А.	5	4	1	5	1
123	Самина О.П.	8	1	10	6	2
124	Гаева Е.Е.	9	3	1	1	2
125	Лаева Ж.Г.	1	3	6	7	1
126	Ваева Б.Д.	9	3	10	10	4
127	Сурикова Е.Е.	5	10	4	10	4
128	Маркина Ц.Ц.	4	4	4	6	1
129	Гуреев П.П.	7	2	1	7	2
130	Сметанина Р.Р.	1	6	1	9	1

Рисунок 19 – Разбиение по грейдам методом опорных векторов



Number	FIO	skill	project_level	labor_capacity	job_complexity	grade
121	Малкина Е.Е.	6	1	5	1	1
122	Забнина А.А.	5	4	1	5	1
123	Самина О.П.	8	1	10	6	2
124	Гаева Е.Е.	9	3	1	1	1
125	Лаева Ж.Г.	1	3	6	7	2
126	Ваева Б.Д.	9	3	10	10	4
127	Сурикова Е.Е.	5	10	4	10	4
128	Маркина Ц.Ц.	4	4	4	6	1
129	Гуреев П.П.	7	2	1	7	1
130	Сметанина Р.Р.	1	6	1	9	1

Рисунок 20 – Разбиение по грейдам методом ближайших соседей



Number	FIO	skill	project_level	labor_capacity	job_complexity	grade
121	Малкина Е.Е.	6	1	5	1	1
122	Забнина А.А.	5	4	1	5	1
123	Самина О.П.	8	1	10	6	2
124	Гаева Е.Е.	9	3	1	1	2
125	Лаева Ж.Г.	1	3	6	7	1
126	Ваева Б.Д.	9	3	10	10	4
127	Сурикова Е.Е.	5	10	4	10	4
128	Маркина Ц.Ц.	4	4	4	6	1
129	Гуреев П.П.	7	2	1	7	2
130	Сметанина Р.Р.	1	6	1	9	1

Рисунок 21 – Разбиение по грейдам с помощью «случайного леса»

3.5 Оценка эффективности проведенного ранжирования

Оценку результатов работы выбранных оптимизационных алгоритмов в ходе машинного обучения можно провести также на основании данных точности вычислений и скорости выполнения обработки данных. При этом оценка включает расчет:

– оценки точности классификатора для каждого метода;

– определение элементов матриц неточности.

При этом для каждого метода с целью оценки общих рисков и ожидаемых потерь производится расчет параметров:

– accuracy – общая точность проведенной классификации в виде относительно частоты выявления правильного класса после сравнения с данными экспертов;

– precision – точность модели, метрика характеризует уровень доверия к модели, которое определяется числом ложных срабатываний, вычисляется как отношение числа выборов класса, которые модель признает как правильные и выбраны они корректно после сравнения с экспертными оценками, к общему числу вариантов, которые система признает правильными;

– recall – полнота, отражающая возможность выявления корректного определения класса на основании численного значения равного отношению вариантов, которые модель считает правильными, и они действительно были правильными к числу всех правильных вариантов распределения по классам.

А также F-меры и общего числа случаев корректного определения класса после сравнения с экспертными оценками.

Результаты, полученные в ходе тестирования для разных методов представлены ниже.

Оценка точности классификаторов

Метод опорных векторов
0.72
Метод ближайших соседей
0.70
Случайный лес
0.81

Отчеты о классификации

Метод опорных векторов

	precision	recall	f1-score	support
1	0.64	1.00	0.78	19
2	0.84	0.73	0.78	27
3	0.70	0.60	0.65	15
4	0.95	0.53	0.68	19
accuracy			0.72	80
macro avg	0.78	0.72	0.72	80
weighted avg	0.79	0.72	0.73	80

Метод ближайших соседей

	precision	recall	f1-score	support
1	0.81	0.99	0.89	24
2	0.63	0.75	0.68	21
3	0.45	0.45	0.45	14
4	0.95	0.47	0.63	21
accuracy			0.70	80
macro avg	0.71	0.67	0.66	80
weighted avg	0.74	0.70	0.69	80

Случайный лес

	precision	recall	f1-score	support

1	0.64	1.00	0.78	18
2	1.00	0.77	0.87	31
3	1.00	0.87	0.93	16
4	0.70	0.60	0.65	15
accuracy			0.81	80
macro avg	0.84	0.81	0.81	80
weighted avg	0.86	0.81	0.82	80

В результате проведенного ранжирования и оценки его результатов можно сказать о достаточной эффективности используемых методик с учетом предметной области используемой задачи. Рост эффективности и снижение общих рисков и ожидаемых потерь возможно путем наращивания числа элементов выборки для обучения.

Выводы по главе

В финальной главе был сформирован набор данных, который включает в себя:

- квалификация или уровень набора навыков, необходимых для выполнения должностных функций (skill);
- средний уровень закрытых за период проектов (projct_level);
- производительность труда за выбранный период (labor_capacity);
- сложность выполняемых работ в закрытых за период проектах (job_complexity).

Также были описаны все используемые библиотеки и подробно разобраны этапы реализации проекта.

После разработки проект был протестирован с использованием различных методов для выделения грейдов. Наиболее эффективным и точным оказался метод k-means.

Заключение

В ходе работы, целью которой является разработка системы классификации эффективности сотрудников на предприятии ООО "Гран Лимитед" с применением технологий машинного обучения, были решены все поставленные задачи исследования:

– проанализирована проблема организации финансовой мотивации сотрудников;

– рассмотрены различные форм оплаты труда и отмечены особенности грейдовой системы;

– проведен анализ существующих алгоритмов, используемых для ранжирования объектов;

– рассмотрены различные методы машинного обучения с учителем для проведения классификации в рамках выбранной прикладной задачи;

– разработан проект системы и выбраны эффективные инструменты реализации;

– проведено тестирование разработанного решения;

– оценена эффективность проведенного ранжирования в ходе эксперимента.

Результатом работы стала система, выполняющая разделение на грейды сотрудников с использованием машинного обучения с учителем. Система разработана средствами языка программирования Python и современных инструментов библиотеки Scikit-learn, которая включает множество технологий машинного обучения как с учителем, так и без.

В результате проведенного ранжирования и оценки его результатов можно сказать о достаточной эффективности используемых методик с учетом предметной области используемой задачи. Рост эффективности и снижение общих рисков и ожидаемых потерь возможно путем наращивания числа элементов выборки для обучения.

Список используемой литературы

1. Басова А.А. Устойчивое развитие территорий: кластерный подход // Материалы международных научных конференций 20–21 апреля 2017 г. СПб.: Скифия-принт, 2017. С. 270-271.
2. Бейдер Д. Чистый Python. Тонкости программирования для профи. СПб.: Питер, 2018. 288 с.
3. Берри П. Изучаем программирование на Python. М.: Эксмо, 2017. 611 с.
4. Вакула А. И., Валуйскова Е. А. Анализ теоретических и практических основ выплаты заработной платы // ЮП. 2016. №2 (76).
5. Ватутин Э.И., Титов В.С., Емельянов С.Г. Основы дискретной комбинаторной оптимизации. М.: АРГАМАК-МЕДИА, 2016. 270 с.
6. Виноградова Ю.А. Современные проблемы мотивации и оплата труда в организациях // Символ науки. 2016. №10-1.
7. Гоник Г.Г., Мельник А.Р. Оптимизация оплаты труда на основе технологии грейдирования // Энигма. 2020. № 19. С. 48-52.
8. Гребнева М. Е., Подтуркина О. А., Савченко Ю. С. Актуальные вопросы организации учета расчетов с персоналом по оплате труда // Научный вестник Крыма. 2018. №2 (13).
9. Епифанова М. А. Основные понятия, виды, формы и системы оплаты труда в современных условиях хозяйствования // Вопросы науки и образования. 2018. №14 (26).
10. Ершов И. А., Стукач О.В. Повышение устойчивости решения задач классификации методами кластерного анализа с корректным нормированием данных // Управление развитием. 2016. Т.185. №3. С.120-129.
11. Карпенко А. П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой. М.: МГТУ, 2017. 447 с.
12. Карпова Т.П. Развитие кадрового потенциала финансовой организации через внедрение системы грейдов // Вестник современных исследований. 2018. № 7.3 (22). С. 451-456.

13. Любанович Б. Простой Python. Современный стиль программирования. СПб.: Питер ,2016. 480 с.

14. Моцная О. В., Чиканова Л. А. Некоторые проблемы правового регулирования заработной платы в Российской Федерации // Журнал российского права. 2016. №6 (234).

15. Пудовкина Д.А., Богатырева И.В. Оптимизация оплаты труда на основе технологии грейдирования // В сборнике: Российская наука: актуальные исследования и разработки. Сборник научных статей VII Всероссийской научно-практической конференции. В 2-х частях. Редколлегия: Г.Р. Хасаев, С.И. Ашмарина [и др.]. 2019. С. 230-234/

16. Слаткин Б. Секреты Python: 59 рекомендаций по написанию эффективного кода. М.: Вильямс, 2016. 274 с.

17. Снегурова В., Кочуренко Н. Основы математической обработки информации. Учебник и практикум. М.: Юрайт, 2017. 220 с.

18. Соколова А. П., Дуборкина И. А. Система оплаты труда в коммерческих организациях // Сервис в России и за рубежом. 2017. №2 (72).

19. Степанова К.А., Дариенко О.Л. Грейдовая система оплаты труда как фактор повышения эффективности мотивации персонала промышленного предприятия // Материалы Четвертой международной научно-практической конференции. 2019 «актуальные вопросы экономики и управления: теоретические и прикладные аспекты». С 113-119.

20. Сулимова М.А. Актуальные проблемы учета расчетов по оплате труда // «Молодежь и наука» Международный аграрный научный журнал, 2015.

21. Суркова А.С., Буденков С.С. Построение модели и алгоритма кластеризации в интеллектуальном анализе данных // Вестник Нижегородского университета им. Н.И. Лобачевского. 2012. №2 (1). С.198-202.

22. Хафизова О. Оплата труда методом грейдирования: системный анализ ценностного обмена // Экономика и управление: научно-практический журнал. 2019. С. 135-139.

23. Шолле Ф. Глубокое обучение на Python. СПб.: Питер, 2018. 486 с.

24. Scikit-learn 0.24.1. URL: https://scikit-learn.org/stable/supervised_learning.html

25. Введение в Scikit-learn. URL: https://neurohive.io/ru/osnovy-data-science/vvedenie-v-scikit-learn/

26. Suzuki S, Abe K. Topological Structural Analysis of Digitized Binary Images by Border Following // CVGIP. 1985. Vol. 1. P. 32-46.

27. Samuel Burns. Python Deep learning: Develop your first Neural Network in Python Using TensorFlow, Keras, and PyTorch (Step-by-Step Tutorial for Beginners). –

28. Ross Girshick Jeff Donahue Trevor Darrell Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation Tech report (v5) [Электронный ресурс]. – Режим доступа: https://arxiv.org/pdf/1311.2524.pdf (дата обращения: 10.02.21)

29. OpenCV: API Documentation [Электронный ресурс]. – Режим доступа: https://docs.opencv.org/2.4/modules/refman.html (дата обращения: 15.02.20)

30. Keras Documentation [Электронный ресурс]. – Режим доступа: https://keras.io/api/ (дата обращения: 15.02.21)

31. Tensorflow API Documentation [Электронный ресурс]. – Режим доступа: https://www.tensorflow.org/api_docs/python/tf (дата обращения: 15.02.21)

Приложение А

Ссылка на проект

https://cloud.mail.ru/public/K55T/ifAXGRKLF