



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2020년05월11일
(11) 등록번호 10-2109187
(24) 등록일자 2020년05월04일

(51) 국제특허분류(Int. Cl.)
G06Q 10/06 (2012.01) G06Q 10/10 (2012.01)
(52) CPC특허분류
G06Q 10/06375 (2013.01)
G06Q 10/063114 (2013.01)
(21) 출원번호 10-2019-0141175
(22) 출원일자 2019년11월06일
심사청구일자 2019년11월06일
(56) 선행기술조사문헌
JP2019191778 A*
KR1020190114581 A*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
(주)위숍
서울특별시 마포구 백범로 35, 아루페관 3층(신수동)
(72) 발명자
김규한
경기도 의정부시 평화로290번길 11, 102동 1603호(호원동, 한주아파트)
(74) 대리인
유상무

전체 청구항 수 : 총 3 항

심사관 : 이명진

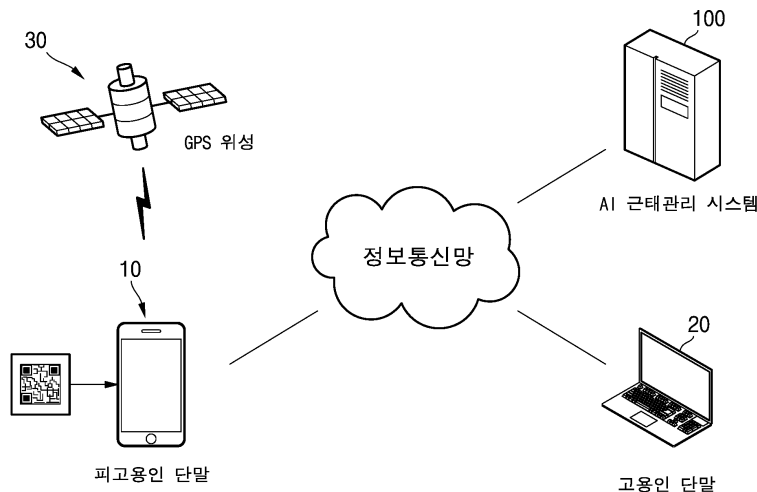
(54) 발명의 명칭 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템

(57) 요약

본 발명은 AI 근태예측 시스템을 공개한다. 보다 상세하게는, 직장내 출퇴근 시간을 관리하고, 평소 출퇴근 행동 패턴에 기반하여 각 피고용인의 전반적인 직장생활과 관련된 유의미한 정보를 제공하는 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템에 관한 것이다.

본 발명의 실시예에 따르면, 피고용인 단말기 및 QR 코드를 이용하여 피고용인의 근태를 관리할 수 있고, 피고용인의 근태정보에 기초하여 급여 산출 및 혜택을 지급할 수 있는 효과가 있다. 또한, 본 발명은 수집된 근태정보를 분석하여 피고용인의 지각 예측, 장기 근속 예측 및 피고용인평가를 산출함으로써, 피고용인 관리에 있어서 유의미한 정보를 제공할 수 있다.

대표도 - 도1



(52) CPC특허분류

G06Q 10/06398 (2013.01)

G06Q 10/105 (2013.01)

명세서

청구범위

청구항 1

피고용인의 근태정보를 수집, 분석 및 저장하고, 근태정보의 분석결과에 따라 피고용인에게 혜택을 지급하는 근태 처리단; 및

저장된 근태정보를 전처리하고, AI 기술에 기반하여 전처리 데이터를 이용한 피고용인별 전반적인 직장생활과 관련된 성향정보를 도출하여 평가하며, 피고용인의 이후 출퇴근 패턴을 예측하는 AI 운영단을 포함하고,

상기 AI 운영단은,

상기 근태 처리단에 저장된 근태정보에 기초하여 기 정의된 포맷에 따라 데이터 전처리를 수행하여 로우 데이터로 변환하는 데이터 처리부;

하나 이상의 학습 모델이 정의되고, 입력되는 로우 데이터를 상기 학습 모델에 적용하는 AI 엔진; 및

상기 AI 엔진의 출력 데이터를 가공하여 평가지수를 산출하는 데이터 가공부를 포함하며,

상기 로우 데이터는,

피고용인의 지각에 관한 데이터를 시간, 기간 및 확률을 포함하는 복수의 항목(지각횟수, 조기출근, 주간 지각횟수, 확률값으로 표현되는 연속지각횟수 및 일정시간이상 지각횟수)으로 세분화한 제1 칼럼;

피고용인의 근속시간에 관한 데이터를 복수의 항목으로 세분화한 제2 칼럼;

고용인이 각 피고용인을 평가한 정성적인 평가점수;

AI 분석을 위한 피고용인의 성별, 연령, 근무시간, 급여, 근무요일, 재직중인 사업장 규모 및 업종 중, 하나 이상을 포함하는 변수별 데이터; 및

상기 피고용인의 출퇴근 행동 패턴에 영향을 주는 하나 이상의 요소를 포함하는 기타 데이터를 포함하고,

상기 데이터 가공부는,

상기 제1 칼럼, 제2 칼럼, 변수별 데이터 및 기타 데이터를 가공하여 피고용인 개인별 종합 평가지수를 산출한 개별정보를 산출하거나,

상기 변수별 데이터를 기준으로 하여, 상기 제1 칼럼, 제2 칼럼 및 기타 데이터를 가공하여 케이스별 평가지수를 산출한 평균 정보를 생성하며,

상기 데이터 처리부는 원핫인 코딩기법을 사용하여 더미변수를 생성하고,

상기 AI 엔진에는 예측결과를 도출하기 위한 지각 예측 모델 및 장기근무 예측 모델이 정의되고, 이를 위한 데이터 전처리 절차에서는 먼저 지각 예측 모델에서 무일정 피고용인은 삭제하며, 장기근무 예측 모델에서 근무중인 피고용인은 삭제하는 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템.

청구항 2

삭제

청구항 3

삭제

청구항 4

삭제

청구항 5

삭제

청구항 6

제 1 항에 있어서,

상기 학습 모델은,

랜덤 포레스트 모형, 로지스틱 회귀 모형, 스토캐스틱 그래디언트 하강 모형 및 그래디언트 부스팅 모형 중 적어도 하나에 기반한, 지각 예측 모델 및 장기근속 예측모델을 포함하는 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템.

청구항 7

제 1 항에 있어서,

상기 근태 처리단은,

피고용인의 근태정보를 수집하고 분석하는 근태정보 처리부;

피고용인의 수집한 근태정보 및 분석 결과를 저장하는 근태정보 저장부; 및

상기 근태정보에 기초하여 피고용인에게 혜택을 지급하는 혜택 지급부

를 포함하는 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템.

발명의 설명

기술 분야

[0001] 본 발명은 AI 근태예측 시스템에 관한 것으로, 특히 직장내 출퇴근 시간을 관리하고, 평소 출퇴근 행동 패턴에 기반하여 각 피고용인의 전반적인 직장생활과 관련된 유의미한 정보를 제공하는 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템에 관한 것이다.

배경 기술

[0002] 현재의 소규모 업소에서의 근태 관리는 별도의 시스템을 마련하지 않고 고용인과 피고용인의 직접 대면에 의하여 출퇴근을 파악하는 방식을 사용하고 있고, 근태정보 또한 별도의 시스템에 저장되는 것이 아니라 고용인이 장부에 수기로 기재하여 관리하는 것이 일반적이다.

[0003] 이는, 소규모 업소에서는 최소 수백만원에서 수천만원에 달하는 근태 관리 시스템을 구매하여 활용한다는 것은 사실상 불가능하기 때문인데, 이와 같이 근태관리가 주먹구구 식으로 이루어짐에 따라 고용인과 피고용인 사이에 실제 근로시간에 대하여 의견이 일치하지 않는 경우도 있고, 근로시간에 대한 급여 또는 수당의 지급에 관해서도 의견이 달라져 분쟁이 발생하는 경우가 종종 있다.

[0004] 이러한 문제점으로 인하여 근태 관리는 자동 시스템을 이용하여 체계적으로 이루어지는 것이 바람직하기에, 소규모 업체에서 저렴한 비용으로 좀 더 쉽게 구현할 수 있는 시스템이 필요하다.

[0005] 따라서 많은 사람들이 사용하고 있는 스마트폰을 이용하여 근태정보를 수집하고, 이를 관리하며 분석할 수 있는 시스템의 필요성이 증대하고 있다.

[0006] 관련 선행기술로 특허공개공보 제10-2003-0034704호(발명의 명칭: 블루투스를 이용한 출입/근태관리시스템 및 방법, 공개일자: 2003.05.09.)가 있다.

[0007] 상기의 선행기술에 따르면, 소정의 카드에 기록된 사원정보를 읽어드려 이를 토대로 출입가능한지 여부를 판단하여 출입 가능한 경우, 출입문 개폐후 출입한 출입정보를 보안 시스템 전송하고, 블루투스 모듈을 통해 보안시스템으로부터 전송된 출입정보를 이용하여 각 피고용인별로 분류시켜 출입/근태정보DB를 구축하고, 출입/근태정보DB에 저장된 출입정보를 통해 각 피고용인별 근태상황이 분석되도록 지원하는 구성을 개시하고 있다.

[0008] 그러나, 종래기술의 경우 단지 출퇴근 시간에 대한 정보만을 이용하여 사원들의 근태관리 기능만을 제공할 뿐이며, 각 피고용인들에 대한 근태정보에 기초한 유용한 정보를 제공하는 정도에는 이르지 못한다는 한계가 있다.

선행기술문헌

특허문헌

[0009] (특허문헌 0001) 등록특허공보 제10-1496646호(공고일자: 2015.03.02.)

발명의 내용

해결하려는 과제

[0010] 본 발명은 전술한 한계를 극복하기 위해 안출된 것으로, 스마트폰과 같은 피고용인 단말을 활용하여 사업장 등에 구비된 QR 코드를 피고용인 단말에서 인식하고, 인식된 QR 코드 정보와 단말의 위치 정보를 근거로 해당 장소에 대한 출입 여부를 확인하여 허위 출퇴근 여부를 확인하고, 용이하게 처리할 수 있도록 하는 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템을 제공하는 데 과제가 있다.

[0011] 또한, 본 발명은 수집된 근태정보를 분석하여 피고용인의 지각 예측, 장기 근속 예측 및 피고용인평가를 산출할 수 있는 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템을 제공하는 데 다른 과제가 있다.

[0012] 또한, 본 발명은 수집된 정보에 대한 분석을 통해 지각을 세분화하고, 근속기간을 세분화하고, 평가점수, 기타 평가 지수 등을 근거로 개별 피고용인에 대한 평가 지수를 산출하고, 산출된 평가 지수를 다양하게 활용할 수 있도록 하는 데 또 다른 과제가 있다.

과제의 해결 수단

[0013] 전술한 과제를 해결하기 위해, 본 발명의 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템은, 피고용인의 근태정보를 수집, 분석 및 저장하고, 근태정보의 분석결과에 따라 피고용인에게 혜택을 지급하는 근태처리단 및 저장된 근태정보를 전처리하고, AI 기술에 기반하여 전처리 데이터를 이용한 피고용인별 전반적인 직장생활과 관련된 성향정보를 도출하여 평가하며, 피고용인의 이후 출퇴근 패턴을 예측하는 AI 운영단을 포함할 수 있다.

[0014] 상기 AI 운영단은, 상기 근태 처리단에 저장된 근태정보에 기초하여 기 정의된 포맷에 따라 데이터 전처리를 수행하여 로우 데이터로 변환하는 데이터 처리부, 하나 이상의 학습 모델이 정의되고, 입력되는 로우 데이터를 상기 학습 모델에 적용하는 AI 엔진 및 상기 AI 엔진의 출력 데이터를 가공하여 평가지수를 산출하는 데이터 가공부를 포함할 수 있다.

[0015] 상기 로우 데이터는, 피고용인의 지각에 관한 데이터를 시간, 기간 및 확률을 포함하는 복수의 항목으로 세분화한 제1 칼럼, 피고용인의 근속시간에 관한 데이터를 복수의 항목으로 세분화한 제2 칼럼, 고용인이 각 피고용인을 평가한 정성적인 평가점수, AI 분석을 위한 피고용인의 성별, 연령, 근무시간, 급여, 근무요일, 재직중인 사업장 규모 및 업종 중, 하나 이상을 포함하는 변수별 데이터 및 상기 피고용인의 출퇴근 행동 패턴에 영향을 주는 하나 이상의 요소를 포함하는 기타 데이터를 포함할 수 있다.

[0016] 상기 데이터 가공부는, 상기 제1 칼럼, 제2 칼럼, 변수별 데이터 및 기타 데이터를 가공하여 피고용인 개인별 종합 평가지수를 산출한 개별정보를 생성할 수 있다.

[0017] 상기 데이터 가공부는, 상기 변수별 데이터를 기준으로 하여, 상기 제1 칼럼, 제2 칼럼 및 기타 데이터를 가공하여 케이스별 평가지수를 산출한 평균 정보를 생성할 수 있다.

[0018] 상기 학습 모델은, 랜덤 포레스트 모형, 로지스틱 회귀 모형, 스토캐스틱 그래디언트 하강 모형 및 그래디언트 부스팅 모형 중 적어도 하나에 기반한, 지각 예측 모델 및 장기근속 예측모형을 포함할 수 있다.

[0019] 상기 근태 처리단은, 피고용인의 근태정보를 수집하고 분석하는 근태정보 처리부, 피고용인의 수집한 근태정보 및 분석 결과를 저장하는 근태정보 저장부, 및 상기 근태정보에 기초하여 피고용인에게 혜택을 지급하는 혜택지급부를 포함할 수 있다.

발명의 효과

[0020] 본 발명의 실시예에 따르면, 피고용인 단말기 및 QR 코드를 이용하여 피고용인의 근태를 관리할 수 있고, 피고

용인의 근태정보에 기초하여 급여 산출 및 혜택을 지급할 수 있는 효과가 있다.

[0021] 또한, 본 발명의 실시예에 따르면, 수집된 근태정보를 분석하여 피고용인의 지각 예측, 장기 근속 예측 및 피고용인평가를 산출함으로써, 피고용인 관리에 있어서 유의미한 정보를 제공할 수 있는 효과가 있다.

[0022] 또한, 본 발명의 실시예에 따르면, 수집된 정보에 대한 분석을 통해 지각을 세분화하고, 근속기간을 세분화하고, 평가점수, 기타 평가 지수 등을 근거로 개별 피고용인에 대한 평가 지수를 산출하고, 산출된 평가 지수를 다양하게 활용할 수 있는 효과가 있다.

도면의 간단한 설명

[0023] 도 1은 본 발명의 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템 및 이와 연결되는 단말을 포함하는 전체 구조를 나타낸 도면이다.

도 2는 본 발명의 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템의 내부 구조를 나타낸 도면이다.

도 3은 본 발명의 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템의 근태정보 처리부의 구조를 나타낸 도면이다.

도 4는 본 발명의 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템의 출퇴근 정보 수집부의 구조를 나타낸 도면이다.

도 5는 본 발명의 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템의 근태 처리단의 구조를 나타낸 도면이다.

도 6은 본 발명의 다른 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템의 근태 처리단의 구조를 나타낸 도면이다.

도 7은 본 발명의 또 다른 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템의 근태 처리단의 구조를 나타낸 도면이다.

도 8은 본 발명의 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 방법을 나타낸 도면이다.

도 9 내지 도 13은 본 발명의 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템에 적용된 지각 예측 모델에 대한 코드 및 이에 따른 예측결과를 나타낸 도면이다.

도 14 내지 도 16은 본 발명의 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템에 적용된 장기근속 예측 모델에 대한 코드 및 이에 따른 예측결과를 나타낸 도면이다.

도 17 내지 도 18은 본 발명의 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템에 적용된 피고용인평가를 도출하기 위한 코드 및 도출결과를 나타낸 도면이다.

도 19는 본 발명의 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템이 제공하는 결과 대시보드 화면을 예시한 도면이다.

발명을 실시하기 위한 구체적인 내용

[0024] 상기한 바와 같은 본 발명을 첨부된 도면들과 실시예들을 통해 상세히 설명하도록 한다.

[0025] 본 발명에서 사용되는 기술적 용어는 단지 특정한 실시 예를 설명하기 위해 사용된 것으로, 본 발명을 한정하려는 의도가 아님을 유의해야 한다. 또한, 본 발명에서 사용되는 기술적 용어는 본 발명에서 특별히 다른 의미로 정의되지 않는 한, 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 의미로 해석되어야 하며, 과도하게 포괄적인 의미로 해석되거나, 과도하게 축소된 의미로 해석되지 않아야 한다. 또한, 본 발명에서 사용되는 기술적인 용어가 본 발명의 사상을 정확하게 표현하지 못하는 잘못된 기술적 용어일 때에는, 당업자가 올바르게 이해할 수 있는 기술적 용어로 대체되어 이해되어야 할 것이다. 또한, 본 발명에서 사용되는 일반적인 용어는 사전에 정의되어 있는 바에 따라, 또는 전후 문맥상에 따라 해석되어야 하며, 과도하게 축소된 의미로 해석되지 않아야 한다.

[0026] 또한, 본 발명에서 사용되는 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한 복수의 표현을 포함한다. 본 발명에서, "구성된다" 또는 "포함한다" 등의 용어는 발명에 기재된 여러 구성 요소들, 또는 여러 단계를 반

드시 모두 포함하는 것으로 해석되지 않아야 하며, 그 중 일부 구성 요소들 또는 일부 단계들은 포함되지 않을 수도 있고, 또는 추가적인 구성 요소 또는 단계들을 더 포함할 수 있는 것으로 해석되어야 한다.

[0027] 또한, 본 발명에서 사용되는 제 1, 제 2 등과 같이 서수를 포함하는 용어는 구성 요소들을 설명하는데 사용될 수 있지만, 구성 요소들은 용어들에 의해 한정되어서는 안 된다. 용어들은 하나의 구성 요소를 다른 구성 요소로부터 구별하는 목적으로만 사용된다. 예를 들어, 본 발명의 권리 범위를 벗어나지 않으면서 제 1 구성 요소는 제 2 구성 요소로 명명될 수 있고, 유사하게 제 2 구성 요소도 제 1 구성 요소로 명명될 수 있다.

[0028] 이하, 첨부된 도면을 참조하여 본 발명에 따른 바람직한 실시 예를 상세히 설명하되, 도면 부호에 관계없이 동일하거나 유사한 구성 요소는 동일한 참조 번호를 부여하고 이에 대한 중복되는 설명은 생략하기로 한다.

[0029] 또한, 본 발명을 설명함에 있어서 관련된 공지 기술에 대한 구체적인 설명이 본 발명의 요지를 흐릴 수 있다고 판단되는 경우 그 상세한 설명을 생략한다. 또한, 첨부된 도면은 본 발명의 사상을 쉽게 이해할 수 있도록 하기 위한 것일 뿐, 첨부된 도면에 의해 본 발명의 사상이 제한되는 것으로 해석되어서는 아니 됨을 유의해야 한다.

[0030] 도 1은 본 발명의 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템 및 이와 연결되는 단말을 포함하는 전체 구조를 나타낸 도면이다.

[0031] 도 1을 참조하면, 본 발명의 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템(100)은 각 지역의 사업장에 설치된 QR 코드를 피고용인 단말(10)에 설치된 전용 어플리케이션으로부터 GPS 위성(30) 등에 의해 획득한 피고용인 단말(10)의 현재 위도 및 경도와, QR 코드에 정의된 사업장의 위도 및 경도를 매칭하여 출퇴근 인증을 진행할 수 있다.

[0032] 상세하게는, 본 발명의 실시예에 따른 AI 근태예측 시스템(100)은 QR 코드 내에 사업장의 위도, 경도의 정보가 저장됨에 따라, 피고용인이 QR 코드를 복사하여 지정된 사업장이 아닌 외부에서 출근 인증을 시도할 경우 출퇴근 범위 이탈로 출퇴근이 불가능하게 된다. 또한, 피고용인 단말(10)의 위도 및 경도와 사업장의 위도 및 경도와 의 실시간 거리측정을 통해 일정거리에 도달할 경우에만 출퇴근이 활성화되며, 피고용인 단말(10) 내 설정되어 있는 고유식별번호 즉, 디바이스 토큰값을 통해 타인의 스마트폰으로 출퇴근이 불가능하게 된다.

[0033] 또한, 본 발명의 실시예에 따른 AI 근태예측 시스템(100)은 하나 이상의 기계학습을 위한 모형에 기반한 학습 모델이 정의된 AI 엔진이 탑재될 수 있고, 이러한 AI 엔진에 의해 현재까지 누적 수집된 근태정보를 이용하여 각 피고용인의 단순 지각 현황이 아닌 이후의 지각 발생과, 장기 근속 여부를 예측할 수 있고, 이에 기초한 피고용인 평가지수를 산출 및 제공할 수 있다.

[0034] 즉, 본 발명의 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템(100)은 피고용인의 근태정보를 수집하여 고용인이 근태관리 작업을 용이하게 수행할 수 있도록 지원할 뿐만 아니라, AI 기술을 도입하여 근태정보에서 유용한 예측정보를 추출하여 활용할 수 있도록 하는 것을 특징으로 한다.

[0035] 이를 위해, 본 발명의 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템(100)은 테스트된 하나 이상의 기계학습을 위한 알고리즘에 의한 최적의 학습 모델을 탑재할 수 있고, 근태정보에 대한 데이터 전처리 수단, AI 엔진 및 데이터 가공수단을 탑재할 수 있다.

[0036] 이하, 도면을 참조하여 전술한 기능을 구현하기 위한 AI 근태예측 시스템(100)의 구성을 상세히 설명한다.

[0037] 도 2는 본 발명의 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템의 내부 구조를 나타낸 도면이다.

[0038] 도 2를 참조하면, 본 발명의 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템(100)은, 피고용인의 근태정보를 수집, 분석 및 저장하고, 근태정보의 분석결과에 따라 피고용인에게 혜택을 지급하는 근태 처리단(101) 및 저장된 근태정보를 전처리하고, AI 기술에 기반하여 전처리 데이터를 이용한 피고용인별 전반적인 직장생활과 관련된 성향정보를 도출하여 평가하며, 피고용인의 이후 출퇴근 패턴을 예측하는 AI 운영단(201)을 포함할 수 있다.

[0039] 상세하게는, 근태 처리단(101)은 근태정보 처리부(110), 근태정보 저장부(120) 및 혜택 지급부(130)를 포함할 수 있다.

[0040] 근태정보 처리부(110)는 피고용인의 근태정보를 수집하고 분석할 수 있다. 여기서, 근태정보는 출근 및 퇴근에 관한 정보를 의미한다. 출근 및 퇴근 시간 정보에 기초하여 일일 근로 시간 및 주간, 월간 근로 시간을 산출할 수 있고 또한 근태 정보를 기초로 분석하여 지각, 조퇴 및 결근 등의 정보를 생성할 수 있다.

- [0041] 근태정보 저장부(120)는 피고용인의 근태정보를 저장할 수 있다. 피고용인의 출근, 퇴근 시간 및 일정 기간 동안의 총 근로시간뿐만 아니라 지각, 조퇴 및 결근에 관한 정보도 저장할 수 있다.
- [0042] 혜택 지급부(130)는 근태정보에 기초하여 피고용인에게 혜택을 지급하기 위한 절차를 수행할 수 있다. 여기서, 혜택은 소정의 근로시간을 충족한 피고용인에게 법률상 또는 계약상 지급하여야 하는 급여 이외의 수당을 의미하거나 복지 포인트와 같은 지급 의무없는 어드벤처지를 의미한다.
- [0043] 전술한 혜택은, 일례로서 소정의 근로시간을 충족한 피고용인에게 식사를 할 수 있는 포인트 및 현금 등을 지급하는 형태로 제공될 수 있다.
- [0044] AI 운영단(201)은 데이터 처리부(210), AI 엔진(220) 및 데이터 가공부(230)를 포함할 수 있다.
- [0045] 데이터 처리부(210)는 근태 처리단(101)에 의해 누적, 수집된 각 피고용인의 근태정보를 기계학습 모델에 입력할 수 있도록 전처리 절차를 수행할 수 있다. 근태정보는 공지의 데이터 베이스 시스템에 따른 테이블 형식으로 구성될 수 있고, 데이터 처리부(210)는 데이터 베이스 테이블에서 학습 모델에 요구되는 데이터만을 추출 및 정렬하여 로우 데이터(raw data)로 전처리할 수 있다.
- [0046] 이러한 로우 데이터에 포함되는 내용으로는 피고용인의 지각에 관한 데이터를 시간, 기간 및 확률을 포함하는 복수의 항목으로 세분화한 제1 칼럼, 피고용인의 근속시간에 관한 데이터를 복수의 항목으로 세분화한 제2 칼럼, 고용인이 각 피고용인을 평가한 정성적인 평가점수, AI 분석을 위한 피고용인의 성별, 연령, 근무시간, 급여, 근무요일, 재직중인 사업장 규모 및 업종 중, 하나 이상을 포함하는 변수별 데이터 및 피고용인의 출퇴근 행동 패턴에 영향을 주는 하나 이상의 요소를 포함하는 기타 데이터 등이 있다.
- [0047] 상기의 제1 칼럼은 시간별, 기간별, 확률별로 세분화된 칼럼값으로서, 예를 들면, 5분 또는 5분 이상의 지각횟수, 5분 또는 5분 이상의 조기출근, 주간 지각횟수, 확률값으로 표현되는 연속지각횟수 및 30분이상 지각횟수 등일 수 있다.
- [0048] 제2 칼럼은 근속기간을 세분화한 칼럼값으로서, 예를 들면 1주일 근무후 퇴사, 3개월 근무 등일 수 있다.
- [0049] 평가점수는, 사업주가 피고용인을 평가한 정성적인 점수로서, 업무수행능력, 친화력 등 일 수 있다.
- [0050] 변수별 데이터는, 성별, 연령, 근무시간대, 급여, 근무요일, 사업장규모, 업종 등일 수 있다.
- [0051] 기타 데이터는 각 피고용인의 출퇴근 행동 패턴에 영향을 줄 수 있는 하나 이상의 요소, 예를들면, 집과 사업장간의 거리, 자녀유무, 자녀수 등의 부가적인 데이터 일 수 있다.
- [0052] 데이터 처리부(210)는 전술한 항목으로 데이터를 전처리 한다.
- [0053] 또한, 데이터 처리부(210)는 전처리 절차 중 하나로서, 'NA값'의 경우 데이터를 보완 또는 삭제할 수 있고, 탐색기반으로 아웃라이어로 추정되는 데이터를 삭제할 수 있다.
- [0054] 또한, 후술하는 AI 엔진(220)에는 예측결과를 도출하기 위한 지각 예측 모델 및 장기근무 예측 모델이 정의될 수 있고, 이를 위한 데이터 전처리 절차에서는 먼저 지각 예측 모델에서 무일정 피고용인은 하루에 일정과 상관없이 근무시간만 채우는 피고용인으로 지각예측과는 상관없이 없으므로 삭제할 수 있고, 장기근무 예측 모델에서 근무중인 피고용인은 근무일수를 계산할 수 없으므로 테스트, 검증데이터에 적합하지 않으므로 삭제할 수 있다.
- [0055] 특히, 데이터 처리부(210)는 급여유형 같은 범주형 데이터에 대하여 모델에 바로 적용할 수 없으므로 원핫인코딩기법을 사용하여 더미변수를 생성할 수 있다. 상세하게는, 범주형 변수를 표현하는데 가장 널리 쓰이는 방법은 원핫인코딩(one-hot-encoding), 원아웃오브엔코딩 또는 가변수가 있다.
- [0056] 원핫인코딩이란, 해당되는 하나의 데이터만 1로 변경해 주고 나머지는 0으로 채워주는 것을 뜻한다. 0과 1로 표현된 변수는 이진 분류공식에 적용할 수 있어서 다음과 같이 개수에 상관없이 범주마다 하나의 특성으로 표현할 수 있다.
- [0057] AI 엔진(220)은 전처리 데이터를 정의된 학습 모델에 입력하여 그 결과를 도출할 수 있다. AI 엔진(220)에 따르면, 피고용인의 지각패턴, 지각비율, 장기근무비율 등 수치를 다양한 기법에 따라 탐색할 수 있다. 예를 들면, 변수에서 지각과 관련이 되는 변수를 탐색하거나, 변수에서 장기근무와 관련이 있는 변수를 탐색하거나, 지각, 장기근무와 관련이 있는 추가 데이터를 탐색할 수 있다.
- [0058] 또한, 본 발명의 AI 엔진(220)은 5분지각을 지각으로 정의하거나, 90일 이상 근무를 장기근무로 정의할 수

있다.

- [0059] 이러한 AI 엔진(220)에 정의되는 학습 모델을 구축하기 위해, 랜덤 포레스트(Random Forest), 로지스틱 회귀(Logistic Regression), 그래디언트 부스팅 분류기(Gradient Boosting Classifier) 및 서포트 벡터 머신(Support Vector Machine) 등 공지의 기계학습 모형에 기반하여 복수의 학습 모델을 구현하고, 각 학습 모델별 예측 소요시간, 예측 정확도를 비교함으로써 최적의 지각 예측, 장기근무 예측 모델을 선정할 수 있다.
- [0060] 또한, 선정된 학습 모델에 대해서는 최적화 과정이 요구된다. 통상적으로 기계학습 알고리즘에는 지정해야 할 설정들이 다수 존재한다. 이러한 튜닝 옵션은 하이퍼 파라미터(hyperparameter)라고 불리우며, 성능을 최적화하거나 편향(bias)과 분산(variance) 사이의 균형을 맞추에 있어서 알고리즘을 조절하기 위해 사용된다. 모델의 성능 최적화를 위해 하이퍼파라미터를 튜닝은 필수 사항이라 할 수 있다. 이에 따라, 학습 모델의 정확도를 기준으로 하여 하이퍼파라미터를 조정하여 모델의 정확도가 최적화 되도록 한다.
- [0061] 이후, 구축된 예측모델에 실제 데이터를 적재하여 최종 검증을 진행할 수 있다. 그 기준으로는 지각 예측 모델의 경우 5분을 지각 기준으로 점원의 지각확률 예측의 정확도를 65%이상으로 설정할 수 있고, 장기근무 예측 모델의 경우 90일 기준으로 점원의 장기근무확률 예측의 정확도를 65%이상으로 설정할 수 있다.
- [0062] 데이터 가공부(230)는 AI 엔진(220)이 출력하는 각 피고용인의 지각 예측 결과, 장기 근속 예측 결과 및 평가 지수를 유의미한 정보로 가공할 수 있다. 전술한 AI 운영단(201)의 구성에 따라 획득된 지각 예측 결과, 장기 근속 예측결과 및 피고용인 평가 지수 등은 다양한 형태로 활용될 수 있다.
- [0063] 예를 들면, 현재 상용화된 채용 플랫폼은 구직자의 과거 근태정보를 확인할 수 없음에 따라, 단순히 구직자의 자소서 내용에만 의존하는 상황이라고 할 수 있다. 이는 불필요한 면접비용 발생, 채용 후 발생하는 비용 발생의 원인이 된다. 이에 구직자에 대한 본 발명의 예측 결과를 채용데이터를 활용한다면 채용 필터링 가능하게 된다.
- [0064] 또한, 본 발명의 예측 결과 및 평가 지수는 신용평가 보조지표로 활용될 수 있다. 현재 금융기관의 신용평가모델 기본데이터는 신용카드 갯수, 최근 3개월간 개설 신용카드수, 연체건, 연체액수, 6개월 내 연체건수 등의 금융거래 실적에만 국한되어 있다.
- [0065] 현재, 각 금융사의 경우 개인회생 등을 통해 삭제된 신용거래 미보유자를 필터링하기 위해 '신용거래 미비자'의 데이터를 활용하고 있지만 사회초년생이며 신용거래 등이 없는 선량한 신용거래 미비자를 구분하기에는 어려운 상황이며, 선량한 신용거래 미비자의 경우에도 수집데이터가 부족함에 따라 강제적으로 신용등급을 하락시키는 경우가 종종 발생하고 있다.
- [0066] 이에 따라, 데이터 가공부(230)는 AI 엔진이 제공하는 예측 결과 등을 기반으로 하여 사회초년생의 불리한 신용평가에 대해 보조지표로 활용할 수 있는 형태로 데이터를 가공 및 출력할 수 있다.
- [0067] 이러한 데이터 가공부(230)는 개별정보 및 평균정보로 구분되도록 데이터를 가공할 수 있고, 개별정보는 전술한 개인별 데이터의 제1 컬럼, 제2 컬럼, 변수데이터 및 기타 데이터를 가공하여 개인별 종합적인 평가지수를 산출한 것으로, 일례로서, '1등급은 상위 5%, 2등급은 상위 10% 형태'의 정보이다.
- [0068] 또한, 평균정보는 변수별 데이터 기준으로 하여 상기 제1 컬럼, 제2 컬럼 및 기타 데이터를 가공하여 케이스별 평가지수를 산출한 것으로, 일례로서 '특정 피고용인 남자, 오전시간 및 주말근무인 경우, 지각확률은 5 %로 전체평균보다 지각확률이 10% 더 낮다'는 형태의 정보이다.
- [0069] 도 3은 본 발명의 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템의 근태정보 처리부의 구조를 나타낸 도면이다.
- [0070] 도 3을 참조하면, 본 발명의 따른 실시예에 따른 근태정보 처리부(110)는, 출퇴근 정보 수집부(111) 및 전송부(115)를 포함할 수 있다.
- [0071] 출퇴근 정보 수집부(111)는 피고용인의 출근 및 퇴근 시간을 수집할 수 있다. 출퇴근 정보는 다양한 형태로 수집할 수 있다. 예를 들어, 피고용인이 출퇴근시에 직접 근무지의 현관문에 설치된 지문 인식기에 지문을 인식함으로써 출퇴근 시간을 인증하거나, 카드를 이용하여 인증할 수 있고, 피고용인이 단말기를 통하여 직접 출퇴근 시간을 입력함으로써 수집하는 방식이 적용될 수 있다. 단, 본 발명의 실시예에 따르면, 피고용인 단말을 통해 각 근무지에 설치된 출퇴근 QR 코드를 인식시켜 사용자 단말이 현재 위치정보 및 출퇴근 정보를 시스템에 전송하는 방식이 적용되는 것이 바람직하다.

- [0072] 전송부(115)는 출퇴근 정보 수집부(111)에서 수집한 출근 및 퇴근 정보를 고용인의 단말기로 전송할 수 있다.
- [0073] 고용인은 피고용인의 근태정보를 알아야 할 뿐만 아니라, 피고용인이 출퇴근 시간을 준수하고 있는지를 확인할 수 있는데, 이를 위하여 실시간으로 피고용인의 출퇴근시간을 고용인 단말로 전송되도록 할 수 있다.
- [0074] 도 4는 본 발명의 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템의 출퇴근 정보 수집부의 구조를 나타낸 도면이다.
- [0075] 도 4를 참조하면, 본 발명의 또 다른 실시예에 따른 출퇴근 정보 수집부(111)는 QR 코드 감지부(111a) 및 위치 정보 비교부(111b)를 포함할 수 있다.
- [0076] QR 코드 감지부(111a)는 피고용인의 모바일 단말기에 위치하면서 출퇴근QR 코드를 감지할 수 있다. 일반적으로, 출퇴근 확인은 지문인식 또는 카드키 인식을 통해서 인증하는 방식이 널리 이용되고 있으나, 지문인식은 인식 성공률이 낮고 스캔 및 식별과정에 비교적 긴 시간이 소요된다. 또한, 카드키는 항상 소지를 해야 하는 불편함이 있다. 따라서, 인식 속도가 빠르고, 대부분이 갖고 있는 스마트폰을 이용하여 QR 코드(QR Code, Quick Response Code)를 스캔함으로써 출퇴근을 인증하면 편리할 수 있다.
- [0077] 그러나 QR 코드는 이를 이미지로 보관하여 아무곳에서나 스캔을 할 수 있다는 문제점이 있다. 즉, 피고용인이 이미지를 갖고서 아무 장소에서나 스캔함으로써 출퇴근을 인증할 수도 있는 것이다.
- [0078] 이러한 문제점을 해결하기 위하여, 본 발명의 실시예에 따르면, 출퇴근 QR 코드는 피고용인의 근무지에 위치하고, 출퇴근 승인 정보와 함께 근무지의 위치정보를 포함할 수 있다. QR 코드에 근무지의 위치정보를 포함함으로써 위치정보 비교부(111b)를 이용하여 피고용인의 올바르게 출퇴근 인증을 수행 했는지 확인할 수 있다.
- [0079] 여기서, 위치정보 비교부(111b)는 QR 코드 감지부(111a)가 출퇴근 QR 코드를 감지할 때의 피고용인의 모바일 단말기의 위치정보와 출퇴근 QR 코드에 포함된 근무지의 위치정보를 비교할 수 있다.
- [0080] 만약, 출퇴근을 인증한 장소의 위치정보가 QR 코드에 포함된 근무지의 위치정보와 다른 경우에는 부정하게 출퇴근 인증을 한 것으로 보아 패널티를 주거나, 해당 시간에 출퇴근을 하지 않은 것으로 간주할 수도 있다.
- [0081] 도 5는 본 발명의 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템의 근태 처리단의 구조를 나타낸 도면이다.
- [0082] 도 5를 참조하면 본 발명의 또 다른 실시예에 따른 근태 처리단(101)은 근태정보 처리부(110), 근태정보 저장부(120), 혜택 지급부(130), 급여 산출부(140) 및 가불 처리부(145)를 포함할 수 있다.
- [0083] 도시된 바와 같이, 급여 산출부(140)를 제외한 나머지 구성요소(110, 120, 130)가 전술한 근태 처리단(101)과 동일하다. 따라서, 본 실시예에서는 급여 산출부(140) 및 가불 처리부(145)에 대해서만 자세히 설명하기로 한다.
- [0084] 급여 산출부(140)는 특정일을 기준으로 지급하여야 할 급여를 산출하고, 급여의 산출 내역을 피고용인 즉, 피고용인 단말 및 고용인 단말로 전송한다.
- [0085] 피고용인이 시간제 근로자인 경우에 급여는 총 근로시간에 시간당 임금을 곱한 금액이 될 것이지만, 연장근로수당, 주휴수당 등 법률 및 계약에 따라 임금 지급 산출 기준은 다양하게 있을 수 있다.
- [0086] 각각의 피고용인 별로 다양한 기준에 따라 산출기준을 달리하여 피고용인 별로 다른 산출식에 따라 급여를 산출할 수 있다.
- [0087] 가불 처리부(145)는 피고용인의 요청에 따라 급여에 대한 가불 서비스를 제공할 수 있다.
- [0088] 상세하게는 가불 서비스는 급여 지급일보다 일정시점 이전에 현재까지의 누적된 일일 급여를 정산하여 피고용인에게 지급하는 서비스로서, 피고용인으로부터 급여에 대한 가불이 요청되면, 출퇴근정보에 기초하여 급여 산출부(140)가 해당 피고용인의 현재 시점까지의 누적 급여를 산출하고, AI 엔진의 근태평가에 따라 가불 지급 여부 및 가불액을 결정하면 피고용인에게 선지급할 수 있다.
- [0089] 여기서, 가불액은 AI엔진에 의한 피고용인 성실도, 장기근무기록 및 점장평가 등의 항목이 반영되어 증감될 수 있다.
- [0090] 가불 서비스 제공 이후, 급여일이 도래하면 피고용인에게 지급되는 급여액은 원 임금 지급 예상액에서 가불액을 차감한 금액이 된다.

- [0091] 이러한 가불 서비스는, 현재 급여 지급 시스템에서 근무중인 피고용인이 급여를 지급받기 위해서는 매월 정해진 지급일까지 기다릴 수 밖에 없음에 따라, 개인 사정 등의 이유로 급히 돈이 필요한 피고용인을 위한 기능이다.
- [0092] 도 6은 본 발명의 다른 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템의 근태 처리단의 구조를 나타낸 도면이다.
- [0093] 도 6을 참조하면 본 발명의 또 다른 실시예에 따른 근태 처리단(101)은 근태정보 처리부(110), 근태정보 저장부(120), 혜택 지급부(130) 및 피고용인 추천부(150)를 포함할 수 있다.
- [0094] 근태정보 처리부(110), 근태정보 저장부(120) 및 혜택 지급부(130)의 설명을 생략하고 피고용인 추천부(150)에 대해서 설명하면, 피고용인 추천부(150)는 피고용인의 근태정보가 소정의 기준치 이상에 해당하는 피고용인을 구인 희망 고용인에게 추천한다.
- [0095] 즉, 피고용인 추천부(150)는 피고용인의 지각 또는 결근의 횟수 또는 비율에 대한 기준치 이상에 해당하는 피고용인을 선별하여 피고용인을 구인 희망 고용인에게 추천한다.
- [0096] 여기서 구인 희망 고용인이란 고용할 피고용인을 찾고 있는 고용인을 의미하고, 한 가지 실시예로서, 적절한 구인 신청 프로세스를 거쳐 구인 희망 그룹에 포함될 수 있도록 할 수 있다.
- [0097] 구인 희망 고용인은 구인 신청을 할 때 피고용인에 대한 자격에 대한 조건을 기재할 수 있고, 그 조건은 피고용인의 근태정보에 대한 분석 결과에 대한 기준일 수 있다.
- [0098] 예를 들어, 희망하는 피고용인은 이전 근무지에서의 총 근로일수에 대한 지각, 조퇴 또는 결근의 비율이 일정수 이하인 피고용인일 것이라는 조건을 기재할 수 있다. 만약 지각의 비율이 1% 이하인 피고용인이라면 300일을 일하는 동안 지각 횟수가 3번 이하인 경우를 의미한다.
- [0099] 따라서 피고용인 추천부(150)는 구인 희망 고용인의 조건에 따라서 지각 비율이 구인 희망 고용인이 요구하는 조건을 충족하는 피고용인만을 선별하여 추천할 수 있다.
- [0100] 도 7은 본 발명의 또 다른 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템의 근태 처리단의 구조를 나타낸 도면이다.
- [0101] 도 7을 참조하면, 본 발명의 또 다른 실시예에 따른 근태 처리단(101)은 근태정보 처리부(110), 근태정보 저장부(120), 경력정보 저장부(160), 혜택 지급부(130) 및 피고용인 추천부(150)를 포함할 수 있다.
- [0102] 근태정보 처리부(110), 근태정보 저장부(120) 및 혜택 지급부(130)의 설명을 생략하고 경력정보 저장부(160)에 대해서 설명하면, 경력정보 저장부(160)는 피고용인 또는 고용인 중 적어도 한쪽에 의하여 입력되는 피고용인의 경력정보를 저장한다.
- [0103] 예를 들어 피고용인은 이전에 근무하였던 근무지 이름, 근무 기간, 직책 및 담당 업무 등을 기재할 수 있다. 그런데 이러한 경력정보는 피고용인이 스스로 기재를 하는 경우에는 허위로 기재할 문제점도 있기에 피고용인이 근무하였던 고용인의 승인이 있는 경우에만 저장이 되도록 할 수도 있다.
- [0104] 그리고 본 실시예에서는 피고용인 추천부(150)는 경력정보 저장부(160)의 피고용인의 경력정보에 더 기초하여 피고용인을 추천한다.
- [0105] 앞선 도 5의 실시예에서는 피고용인의 근태정보에 기초해서만 구인 희망 고용인에게 피고용인을 추천하였는데, 본 실시예에서는 피고용인의 경력정보에 더 기초하여 피고용인을 추천할 수 있다.
- [0106] 예를 들어, 고용인은 특정 업무를 최소 6개월 이상 해본 경험이 있는 피고용인을 고용하고자 하는 조건을 기재한 경우에는 그 조건을 충족하는 피고용인을 추천할 수 있다.
- [0107] 도 8은 본 발명의 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 방법을 나타낸 도면이다.
- [0108] 도 7을 참조하면, 단계 710에서는, 스마트 근태 관리 시스템(100)이 피고용인의 근태정보를 수집하고 분석한다.
- [0109] 단계 720에서는, 스마트 근태 관리 시스템(100)이 피고용인의 근태정보를 저장한다.
- [0110] 단계 730에서는, 스마트 근태 관리 시스템(100)이 근태정보에 기초하여 피고용인에게 혜택을 지급한다.
- [0111] 도 9 내지 도 13은 본 발명의 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템에 적용된 지각 예측 모델에 대한 코드 및 이에 따른 예측결과를 나타낸 도면이다.

- [0112] 본 발명의 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템은, 랜덤 포레스트 모형, 로지스틱 회귀 모형, 스토캐스틱 그래디언트 하강 모형 및 그래디언트 부스팅 모형 중 적어도 하나에 기반하여 지각 예측 모델 및 장기근속 예측모델을 구축함으로써 피고용인에 대한 출퇴근 행동 패턴을 예측하는 것을 특징으로 한다.
- [0113] 전술한 기계학습 모형 중, 랜덤포레스트(Random Forest)는 부트스트랩 (bootstrap)표본을 이용할 뿐 아니라 변수에 임의성을 더한 방법을 말하며 앙상블 이론이 갖는 장점을 극대화하여 예측 및 분류 정확도가 높다는 특징이 있다.
- [0114] 로지스틱 회귀(logistic regression)는 확률 모델로서 독립 변수의 선형 결합을 이용하여 사건의 발생 가능성을 예측하는데 사용되는 통계 기법이다. 흔히 로지스틱 회귀는 종속변수가 이항형 문제(즉, 유효한 범주의 개수가 두개인 경우)를 지칭할 때 사용된다.
- [0115] 스토캐스틱 그래디언트(stochastic gradient descent)는 함수가 어렵고 복잡하여 수학적 접근 방법으로 풀기 어려운 문제에 안정적인 동작을 보여준다.
- [0116] 그래디언트 부스팅(Gradient Boosting; GBM)은 회귀분석 또는 분류 분석을 수행할 수 있는 기계학습 알고리즘 중에서도 가장 예측 성능이 높다고 알려진 알고리즘이다.
- [0117] 본 발명에서는 과거, 미래 기준으로 8:2로 전처리된 데이터를 학습용 데이터와 검증용 데이터를 분리하여 AI 엔진을 설정할 수 있다.
- [0118] 이하, 전술한 기계학습 모형에 기반한 본 발명의 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템의 구현형태를 설명한다.
- [0119] 도 9을 참조하면, 본 발명의 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템에 적용된 지각 예측 모델에 대한 소스코드의 일부로서, 데이터 적재 및 전처리 분석과 예측에 필요한 공지의 파이썬 라이브러리가 이용될 수 있다. 특히, Pandas 및 numpy 라이브러리를 사용함에 따라 데이터 객체를 이용해서 데이터를 쉽게 가공이 가능하며, 더 나아가 통계연산이 용이하고, 그 외 데이터 변수 사이의 연관성, 그룹, 선택 및 조인 등의 다양한 함수를 통해 매트릭스 형의 데이터를 효율적이고 쉽게 가공할 수 있다. 또한, 예측모델인 그래디언트 부스팅 알고리즘 및 ROC(Receiver Operating Characteristic)이 적용될 수 있다.
- [0120] 도 10은 본 발명의 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템의 데이터 처리부에 대한 소스코드를 예시하고 있으나, 데이터 전처리를 위한 소스코드가 이에 한정되는 것인 아니다.
- [0121] 도 11은 본 발명의 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템을 구현하기 위해, 랜덤 포레스트 모형, 로지스틱 회귀 모형, 스토캐스틱 그래디언트 하강 모형 및 그래디언트 부스팅 모형에 기반하여 지각 예측 모델을 구현하기 위한 소스코드 및 그 테스트 결과를 나타내고 있다.
- [0122] 도 11을 참조하면, 4개의 모델 예측 결과값을 확인 결과, 정확도(acc; accuracy) 및 수신자 조작 곡선(auc; Area Under ROC Curve ROC(Receiver operating Characteristic Curve) 곡선을 산출할 수 있다. 여기서, auc는 모든 분류 임계값에서 분류 모델의 성능을 보여주는 그래프의 곡선 아래 영역이며, recall은 지각을 지각으로 정확히 분류한 비율을 가리킨다. 상기의 acc, auc 및 recall의 3개의 값을 비교한 결과, 그래디언트 부스팅이 지각 예측모델에 제일 적절함을 알 수 있다.
- [0123] 테스트 결과, 전체적인 정확도는 약 76%를 나타내고, 그 중 지각을 정확히 분류한 비율은 10%이며, 2중 분류 모델의 성능을 나타내는 auc값은 0.483인 것을 확인할 수 있다..
- [0124] 도 12는 전술한 그래디언트 부스팅에 기반한 학습 모델을 이용한 지각 예측 결과를 생성하기 위한 소스코드의 일부를 나타내고 있고, 도 13은 도 12의 전처리 및 학습 모델에 의해 생성된 예측결과를 나타내고 있다.
- [0125] 도 12 및 도 13을 참조하면, 예측결과로서, 가게ID, 피고용인ID, 출근날짜, 출근 시간을 AI 엔진에 입력하면, 예측한 날짜의 지각확률을 소수점 2자리 단위로 확인할 수 있다. 도 13에서 무일정인 피고용인은 'CALENDAR'로 표시된다. 그리고, 예측결과는 콤마를 기준으로 분리하는 'csv'형식으로 저장될 수 있다.
- [0126] 도 14 내지 도 16은 본 발명의 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템에 적용된 장기근속 예측 모델에 대한 코드 및 이에 따른 예측결과를 나타낸 도면이다.
- [0127] 도 14를 참조하면, 본 발명의 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템에 적용된 장기근속 예측 모델은 파이썬이 지원하는 라이브러리를 적재하고, 근태정보에서 필요한 로우 데이터, 일레로서, 가게ID,

피고용인ID, 출근날짜, 출근 시간 등에 대한 컬럼만을 추출하여 데이터 셋을 준비한다.

[0128] 이후, 도 15에 나타난 소스코드에 따라, 그래디언트 부스팅 모형을 이용하여 예측을 수행하면, 도 16의 예측결과를 얻을 수 있다. 이러한 장기근속 예측결과 또한 가게ID, 피고용인ID, 근무시작일을 입력하면 해당피고용인의 장기근속확률을 소수점 2자리 단위로 보여주며, CSV 형식으로 저장될 수 있다.

[0129] 도 17 및 도 18은 본 발명의 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템에 적용된 피고용인 평가를 도출하기 위한 코드 및 도출결과를 나타낸 도면이다.

[0130] 본 발명의 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템은 피고용인에 대하여 전술한 지각 예측 결과 및 장기 근속 예측 결과를 이용한 피고용인과 관련된 피고용인평가를 도출할 수 있다.

[0131] 도 17은 피고용인평가를 위한 시스템으로의 데이터 적재 즉, 데이터 처리부의 소스코드 일부를 나타내고 있고, 도 18은 피고용인별 평가결과를 나타내고 있다.

[0132] 도 17의 소스코드에 의하면, 피고용인의 평가지수에 필요한 데이터를 탐색하게 되며, 그 항목으로는 피고용인 성실도(비지각확률), 장기근무기록, 장기근무확률, 점장평가 등이 포함될 수 있다. 또한, 탐색된 데이터를 사용하여 평가지수를 산출하게 되는데, 그 산출방법을 수식으로 나타내면 이하의 수학적 식 1,

수학적 식 1

[0133]
$$\text{평가점수} = (a \times \text{비지각확률}) + (b \times \text{장기근속확률}) + (c \times \text{점장평가})$$

[0135] 로 정의 될 수 있다. 여기서, a,b,c는 조정된 가중치일 수 있다.

[0136] 또한, 피고용인ID와, 도출된 지각확률 및 장기근무확률을 입력하면 피고용인별 평가지수를 소수점 2자리 단위로 확인할 수 있다. 평가지수는 0점 내지 5점 사이의 값으로 표시된다. 입력 데이터 형식은 CSV가 이용된다.

[0137] 도 19는 본 발명의 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템이 제공하는 결과 대시보드 화면을 예시한 도면이다.

[0138] 도 19를 참조하면, 본 발명의 실시예에 따른 출퇴근 행동 패턴에 기반한 AI 근태예측 시스템은 지각 예측결과를 대시보드 형식으로 제공할 수 있다. 이러한 대시보드는 가게별 및 점원별 지각확률을 수치 및 그래프 형태로 동시에 표시할 수 있다. 특히, 관리자의 시인성을 높이기 위해, 지각 확률이 높은 피고용인의 그래프의 원을 상대적으로 크게 표시할 수 있다.

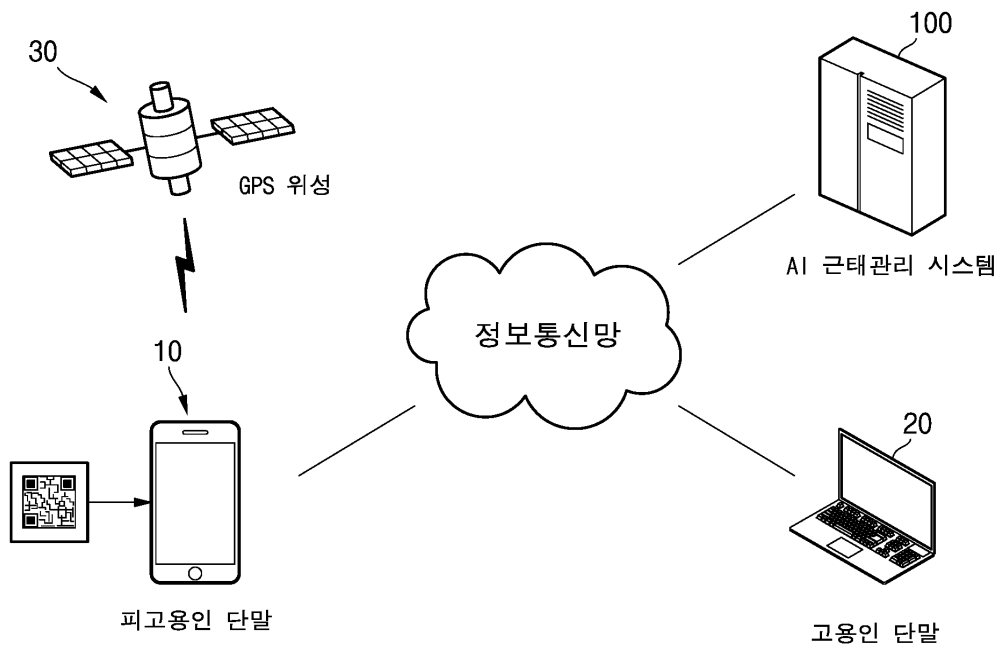
[0139] 이상에서는 본 발명에 따른 바람직한 실시예들에 대하여 도시하고 또한 설명하였다. 그러나 본 발명은 상술한 실시예에 한정되지 아니하며, 특허 청구의 범위에서 첨부하는 본 발명의 요지를 벗어남이 없이 당해 발명이 속하는 기술 분야에서 통상의 지식을 가진 자라면 누구든지 다양한 변형 실시가 가능할 것이다.

부호의 설명

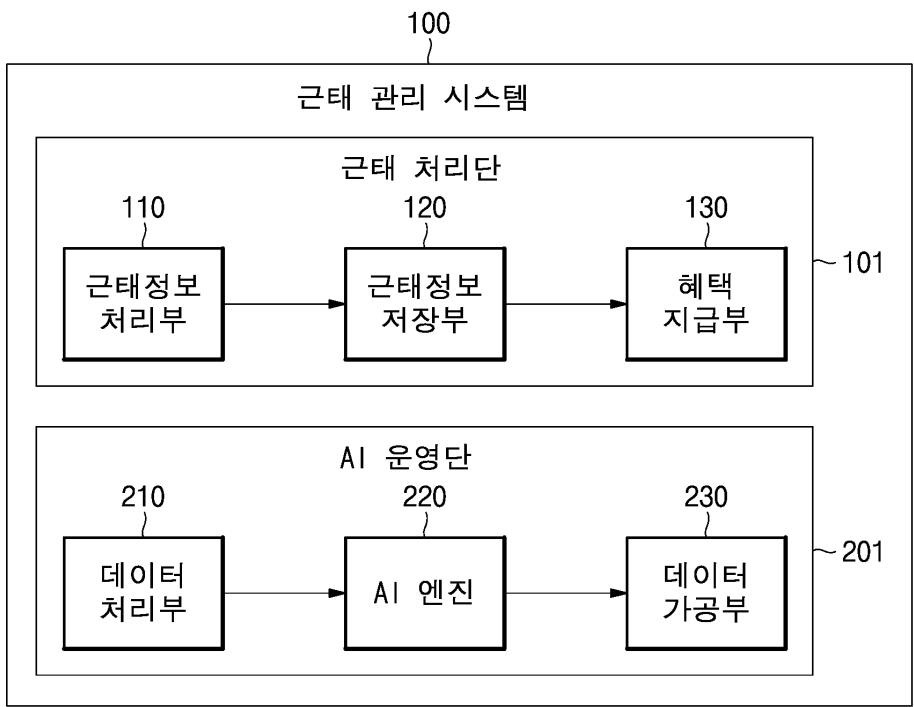
| | | |
|--------|----------------|-------------------|
| [0140] | 10 : 피고용인 단말 | 20 : 고용인 단말 |
| | 30 : GPS 위성 | 100 : AI 근태관리 시스템 |
| | 101 : 근태 처리단 | 110 : 근태정보 처리부 |
| | 120 : 근태정보 저장부 | 130 : 혜택 지급부 |
| | 201 : AI 운영단 | 210 : 데이터 처리부 |
| | 220 : AI 엔진 | 230 : 데이터 가공부 |

도면

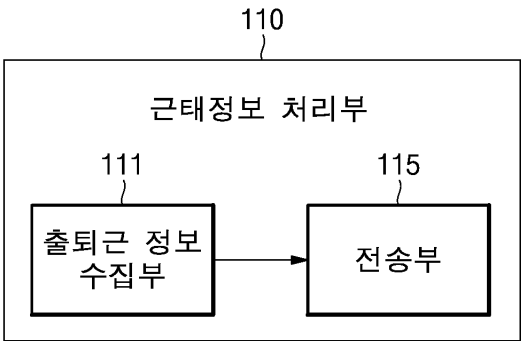
도면1



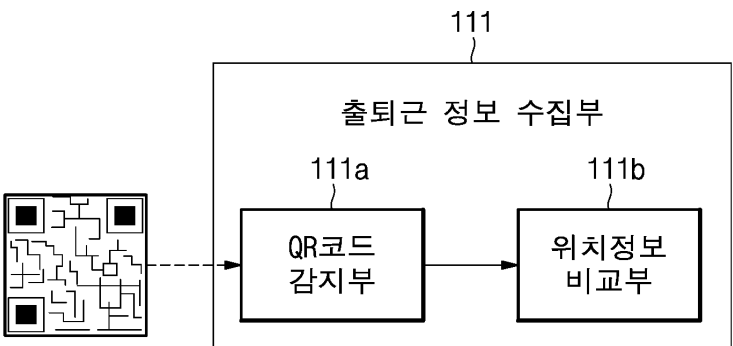
도면2



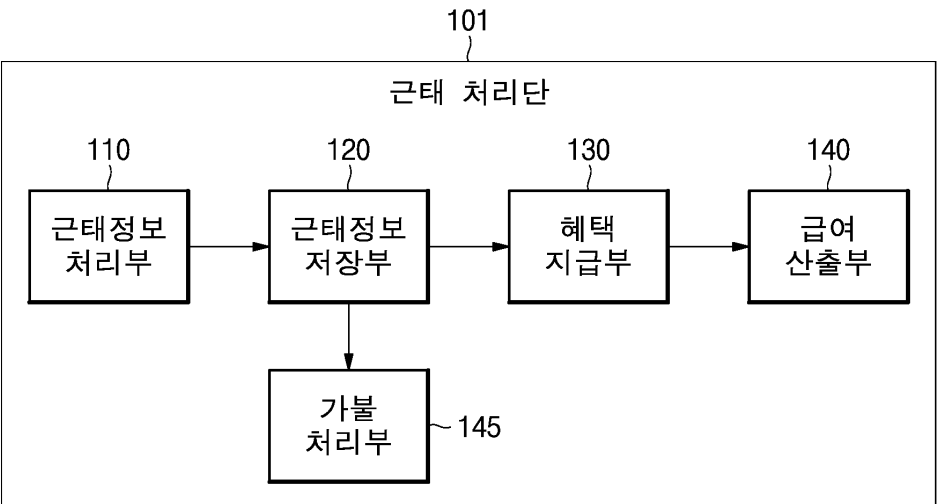
도면3



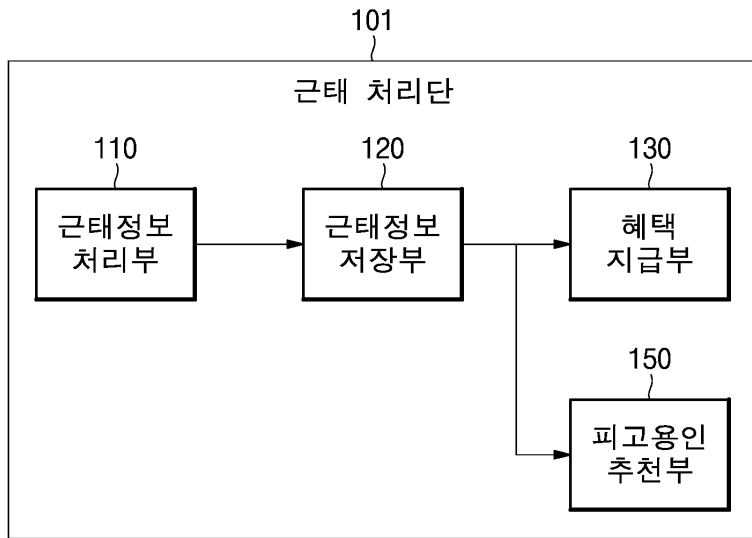
도면4



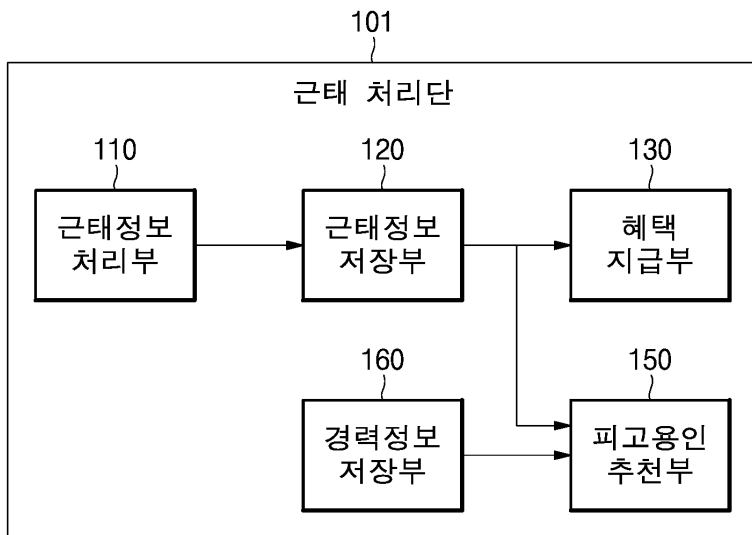
도면5



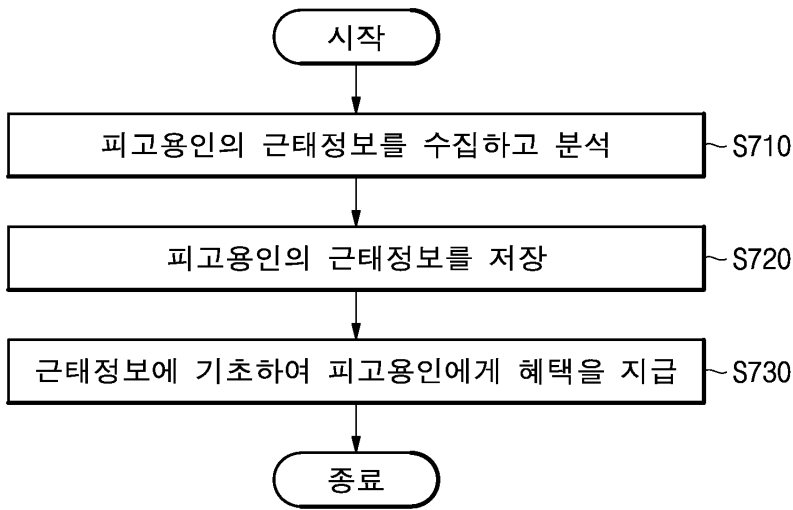
도면6



도면7



도면8



도면9

```

import pandas as pd
import numpy as np
from sklearn.metrics import roc_auc_score
from sklearn.ensemble import GradientBoostingClassifier

df_Time = pd.read_csv('F:/Python/Wesop/Data/출퇴근2.csv', engine='python')
df_Payment = pd.read_csv('F:/Python/Wesop/Data/매월급여정보2.csv', engine='python')
df_Insurance = pd.read_csv('F:/Python/Wesop/Data/매월보험정보2.csv', engine='python')
df_Store = pd.read_csv('F:/Python/Wesop/Data/사업장2.csv', engine='python')
df_User = pd.read_csv('F:/Python/Wesop/Data/유저2.csv', engine='python')
df_Emp = pd.read_csv('F:/Python/Wesop/Data/직원2.csv', engine='python')
RedDate=pd.read_csv('F:/Python/Wesop/Data/휴일데이터.csv', engine='python')
    
```

도면10

```

#날짜데이터를 날짜 형식으로 변경
df['Time'] = df['NEW_START'].astype('datetime64[ns]')
df['Time'] = df['NEW_END'].astype('datetime64[ns]')
df['Time'] = df['NEW_START_TIME'].astype('datetime64[ns]')
df['Time'] = df['NEW_END_TIME'].astype('datetime64[ns]')
df['Time'] = df['DATE'].astype('datetime64[ns]')
df['Time'] = df['DATE'].astype('datetime64[ns]')
df['Time'] = df['DATE'].astype('datetime64[ns]')

#데이터에서 필요한 컬럼만 추출
df_Emp = df_Emp[['EMP_SEQ', 'STORE_SEQ', 'CALENDAR', 'IS_RABAGIN']] # 직원id, 사업장id, 무동장여부, 객니지여부
df_Time2 = df_Time[['DATE', 'EMP_SEQ', 'NEW_START', 'NEW_START_TIME', 'LATE', 'START']] #날짜, 직원id, 출장출근시간, 실제출근시간, 공휴일, 출근시간
df_Store2 = df_Store[['STORE_SEQ', 'TYPE']] #사업장id, 신사업장 여부
df_Payment2 = df_Payment[['EMP_SEQ', 'PAY_TYPE', 'PAY']] # 직원id, 급여유형, 급여

#데이터를 결합
df = pd.merge(df_Emp2, df_Time2, left_on='EMP_SEQ', right_on='EMP_SEQ')
df = df.drop(['EMP_SEQ'], axis = 1)
df = pd.merge(df, df_Store2, on='STORE_SEQ')
df = pd.merge(df, df_Payment2, on='EMP_SEQ')

#무일정 데이터 삭제
df = df[df['CALENDAR']!=0]
df = df.drop(['CALENDAR'], axis = 1)

#직원 id별 누적 지각수 계산
by_id = df[df['NEW_START_TIME']-df['NEW_START']>'00:05:00']
by_id_count = pd.DataFrame(by_id['EMP_ID'].value_counts())
by_id_count['idx'] = by_id_count.index
by_id_count.columns = ['LateCount', 'EMP_ID']
df = pd.merge(df, by_id_count, on='EMP_ID')
dt = pd.merge(dt, by_id_count, on='EMP_ID', how='left')

#매장별 직원수 계산
store_emp = pd.DataFrame(df_Emp.loc[df_Emp['END'].isna()], 'STORE_SEQ').value_counts()
store_emp['store'] = store_emp.index
store_emp.columns = ['Store_num', 'STORE_SEQ']
df = pd.merge(df, store_emp, on='STORE_SEQ')
dt = pd.merge(dt, store_emp, on='STORE_SEQ')

#기본정보에서 중복제거
dt = dt.drop_duplicates(['EMP_ID'], keep='last')

#출근데이터 없는 데이터 삭제
df = df[df['NEW_START_TIME'].notnull()]

#실제출근시간-일정출근시간에서 5분이상이면 지각으로 판정
df.loc[df['NEW_START_TIME']-df['NEW_START']>'00:05:00', 'late'] = 1
df.loc[df['NEW_START_TIME']-df['NEW_START']<='00:05:00', 'late'] = 0

#출근시간의 '시'만 남김
df['START'] = df['START'].apply(lambda x: int(x.split(':')[0]))

#날짜 중 데이터 삭제
df = df.drop(['DATE', 'NEW_START_TIME', 'NEW_START', 'STORE_SEQ', 'EMP_ID'], axis=1)

##### 데이터셋 준비 완료 #####

#독립변수와 종속변수 분리
x = df.drop(['late'], axis = 1)
y = df['late']

#범주형변수 원형인코딩 후 연속형변수와 합침
x1 = x[['PAY', 'LateCount', 'Store_num']]

x2 = x.drop(['PAY', 'LateCount', 'Store_num'], axis = 1)
x2 = x2.astype('category')
x2 = pd.get_dummies(x2, drop_first=True)

x = pd.concat([x1, x2], axis=1)

#전체 데이터의 80%를 테스트셋으로 분리
tr_num = round(df.shape[0]*0.8)
x_train = x.loc[0:tr_num, :]
y_train = y.loc[0:tr_num]
x_test = x.loc[tr_num+1:df.shape[0], :]
y_test = y.loc[tr_num+1:df.shape[0]]

```

도면11

```

#랜덤포레스트
from sklearn.ensemble import RandomForestClassifier

model_rf = RandomForestClassifier(random_state=99)
model_rf.fit(x_train, y_train)
predict_rf=model_rf.predict(x_test)
print("acc:",model_rf.score(x_test, y_test))
print("auc:",roc_auc_score(y_test, predict_rf))
print("recall:",metrics.recall_score(y_test, predict_rf))

```

```

acc: 0.7426981919332406
auc: 0.4587732316590049
recall: 0.06467661691542288

```

```

#로지스틱 회귀
from sklearn.linear_model import LogisticRegression

model_lr=LogisticRegression(random_state=99)
model_lr.fit(x_train, y_train)
predict_lr=model_lr.predict(x_test)
print("acc:",model_lr.score(x_test, y_test))
print("auc:",roc_auc_score(y_test, predict_lr))
print("recall:",metrics.recall_score(y_test, predict_lr))

```

```

acc: 0.8602225312934632
auc: 0.5
recall: 0.0

```

```

#STOCHASTIC GRADIENT DESCENT
from sklearn.linear_model import SGDClassifier

model_sdgc=SGDClassifier(random_state=99)
model_sdgc.fit(x_train, y_train)
predict_sdgc=model_sdgc.predict(x_test)
print("acc:",model_sdgc.score(x_test, y_test))
print("auc:",roc_auc_score(y_test, predict_sdgc))
print("recall:",metrics.recall_score(y_test, predict_sdgc))

```

```

acc: 0.8602225312934632
auc: 0.5
recall: 0.0

```

```

#GRADIENT BOOSTING CLASSIFIER
model_gbc = GradientBoostingClassifier(random_state=99)
model_gbc.fit(x_train, y_train)
predict_gbc=model_gbc.predict(x_test)
print("acc:",model_gbc.score(x_test, y_test))
print("auc:",roc_auc_score(y_test, predict_gbc))
print("recall:",metrics.recall_score(y_test, predict_gbc))

```

```

acc: 0.7559109874826148
auc: 0.4831199700768591
recall: 0.1044776119402985

```

도면12

```

* #테스트 데이터 입력
df_Test_bk = pd.read_csv('F:/Python/Wesop/Data/test.csv', engine='python')
df_Test=df_Test_bk

#테스트데이터 전처리(기본정보와 합침)
df_Test['DATE']=df_Test['DATE'].astype('datetime64[ns]')
df_Test=pd.merge(df_Test,RedDate,on='DATE')
df_Test=df_Test.drop(['DATE'],axis=1)
df_Test['START']=df_Test['START'].apply(lambda x: int(x.split(':')[0]))
df_Test=pd.merge(df_Test,dt,on=['EMP_ID'],how='left')

df_Test1=df_Test[['PAY','LateCount','Store_num']]
df_Test2=df_Test.drop(['PAY','LateCount','Store_num','STORE_SEQ_x','STORE_SEQ_y','EMP_ID','CALENDAR'],axis=1)
df_Test2=df_Test2.astype('category')
df_Test2=pd.get_dummies(df_Test2,drop_first=True)
df_Test=pd.concat([df_Test1,df_Test2],axis=1)

* for i in x.columns:
*     if (i not in(df_Test.columns)):
*         df_Test[i]=0
df_Test['LateCount']=df_Test['LateCount'].fillna(0)

* #테스트데이터 예측 그래디언트 부스팅
t=pd.DataFrame()
t=pd.DataFrame(model_gbc.predict_proba(df_Test)[:,1]).apply(lambda x : round(x,2))
t.columns=['LateRate']
t=pd.concat([df_Test_bk,t],axis=1)
t=pd.merge(t,dt,on='EMP_ID',how='left')
t.loc[t['CALENDAR']==1,'LateRate']='CALENDAR'
result=pd.concat([df_Test_bk,t[['LateRate']]],axis=1)

```

도면13

| | STORE_SEQ | EMP_ID | DATE | START | LateRate |
|----|-----------|--------|------------|----------|----------|
| 0 | 800 | 779 | 2019-09-17 | 10:11:00 | CALENDAR |
| 1 | 1244 | 1323 | 2019-09-17 | 9:10:00 | 0.61 |
| 2 | 686 | 674 | 2019-09-17 | 10:25:00 | CALENDAR |
| 3 | 799 | 777 | 2019-09-17 | 9:30:00 | 0.15 |
| 4 | 799 | 808 | 2019-09-17 | 10:47:00 | CALENDAR |
| 5 | 1052 | 1276 | 2019-09-17 | 11:00:00 | 0.12 |
| 6 | 815 | 823 | 2019-09-17 | 11:00:00 | 0.12 |
| 7 | 1052 | 1277 | 2019-09-17 | 11:00:00 | 0.12 |
| 8 | 1052 | 1275 | 2019-09-17 | 11:00:00 | 0.12 |
| 9 | 1011 | 1103 | 2019-09-17 | 11:00:00 | 0.14 |
| 10 | 541 | 470 | 2019-09-17 | 12:00:00 | 0.67 |
| 11 | 886 | 868 | 2019-09-17 | 12:00:00 | 0.06 |
| 12 | 778 | 874 | 2019-09-17 | 12:08:00 | CALENDAR |
| 13 | 1008 | 1089 | 2019-09-17 | 14:00:00 | 0.06 |
| 14 | 886 | 854 | 2019-09-17 | 13:00:00 | 0.15 |
| 15 | 387 | 195 | 2019-09-17 | 13:00:00 | 0.05 |
| 16 | 387 | 672 | 2019-09-17 | 17:00:00 | 0.06 |
| 17 | 387 | 323 | 2019-09-17 | 17:00:00 | 0.07 |
| 18 | 1011 | 1079 | 2019-09-17 | 14:00:00 | 0.06 |
| 19 | 977 | 980 | 2019-09-17 | 13:00:00 | 0.69 |
| 20 | 1007 | 1039 | 2019-09-17 | 13:00:00 | 0.13 |
| 21 | 1019 | 1327 | 2019-09-17 | 13:01:00 | CALENDAR |
| 22 | 1019 | 1326 | 2019-09-17 | 9:00:00 | 0.61 |
| 23 | 1009 | 1084 | 2019-09-17 | 14:00:00 | 0.06 |
| 24 | 1011 | 1090 | 2019-09-17 | 14:00:00 | 0.06 |
| 25 | 1009 | 1098 | 2019-09-17 | 14:00:00 | 0.06 |
| 26 | 1011 | 1099 | 2019-09-17 | 14:00:00 | 0.06 |
| 27 | 1011 | 1100 | 2019-09-17 | 14:00:00 | 0.06 |

도면14

```

#라이브러리 적재
import pandas as pd
import numpy as np
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.metrics import roc_auc_score

#데이터 적재
df_Time = pd.read_csv('F:/Python/Wesop/Data/출퇴근2.csv', engine='python')
df_Payment = pd.read_csv('F:/Python/Wesop/Data/매월급여정보2.csv', engine='python')
df_Insurance = pd.read_csv('F:/Python/Wesop/Data/매월보험정보2.csv', engine='python')
df_Store = pd.read_csv('F:/Python/Wesop/Data/사업장2.csv', engine='python')
df_User = pd.read_csv('F:/Python/Wesop/Data/유저2.csv', engine='python')
df_Emp = pd.read_csv('F:/Python/Wesop/Data/직원2.csv', engine='python')
df_Schedule = pd.read_csv('F:/Python/Wesop/Data/직원근무일정2.csv', engine='python')

#날짜데이터를 날짜 형식으로 변경
df_Time['NEW_START']=df_Time['NEW_START'].astype('datetime64[ns]')
df_Time['NEW_END']=df_Time['NEW_END'].astype('datetime64[ns]')
df_Time['NEW_START_TIME']=df_Time['NEW_START_TIME'].astype('datetime64[ns]')
df_Time['NEW_END_TIME']=df_Time['NEW_END_TIME'].astype('datetime64[ns]')
df_Time['DATE']=df_Time['DATE'].astype('datetime64[ns]')
df_Emp2['START']=df_Time['START'].astype('datetime64[ns]')

#데이터에서 필요한 컬럼만 추출
df_Emp2 = df_Emp[['EMP_SEQ', 'STORE_SEQ', 'TYPE', 'CALENDAR', 'IS_MANAGER', 'START', 'END']]
# 직원id, 사업장id, 급여유형, 우월정보, 매니저여부, 근무시작일, 근무종료일
df_Time2 = df_Time[['EMP_ID', 'NEW_START', 'NEW_START_TIME']] #날짜, 직원id, 월정출근시간, 실제출근시간
df_Store2 = df_Store[['STORE_SEQ', 'TYPE']] #사업장id, 5인사업장 여부
df_User2 = df_User[['MEMBER_SEQ', 'GENDER']] #직원id, 성별
df_Payment2 = df_Payment[['EMP_ID', 'PAY_TYPE', 'PAY', 'MEALS', 'SELF_DRIVING', 'BONUS', 'INCENTIVE']]
#직원id, 급여유형, 급여, 식대, 자가운전, 보너스, 인센티브
df_Payment2 = df_Payment2.drop_duplicates(['EMP_ID'], keep='last') #마지막 급여만 남김
#직원id, 근무수당, 연봉, 이대보통금 여부
df_Insurance2 = df_Insurance.drop_duplicates(['EMP_ID'], keep='first')
df_Insurance3 = df_Insurance.drop_duplicates(['EMP_ID'], keep='last')
df_Insurance2 = pd.merge(df_Insurance2, df_Insurance3, on='EMP_ID')

#데이터셋 결합
df = pd.merge(df_Emp2, df_Store2, on='STORE_SEQ', how='left')
df = pd.merge(df, df_User2, left_on='EMP_SEQ', right_on='MEMBER_SEQ', how='left')
df = df.drop(['MEMBER_SEQ'], axis = 1)
df = pd.merge(df, df_Payment2, left_on='EMP_SEQ', right_on='EMP_ID', how='left')
df = df.drop(['EMP_SEQ'], axis = 1)
df = pd.merge(df, df_Insurance2, on='EMP_ID', how='left')

#직원id, 성별, 능력, 지각수 계산
by_id = df_Time2[df_Time2['NEW_START_TIME']-df_Time2['NEW_START']>'00:05:00']
by_id_count = pd.DataFrame(by_id['EMP_ID'].value_counts())
by_id_count['ids'] = by_id_count.index
by_id_count.columns = ['LateCount', 'EMP_ID']
df = pd.merge(df, by_id_count, on='EMP_ID', how='left')
display(df[df['END'].notnull()].shape)

#매장별 직원수 계산
store_emp = pd.DataFrame(df_Emp2.loc[df_Emp2['END'].isna(), 'STORE_SEQ'].value_counts())
store_emp['store'] = store_emp.index
store_emp.columns = ['Store_sum', 'STORE_SEQ']
df = pd.merge(df, store_emp, on='STORE_SEQ', how='left')
display(df[df['END'].notnull()].shape)

#근로종료일 있는 데이터만 추출
df = df[df['END'].notnull()]

#근로종료일-근무시작일 90일이상이면 지각으로 판정
df.loc[df['END']-df['START']>'00:05:00', 'longtime'] = 1
df.loc[df['END']-df['START']<='00:05:00', 'day'] = 0

#지각 데이터 삭제
df = df.drop(['STORE_SEQ', 'EMP_ID'], axis=1)

##### 데이터셋 준비 완료 #####

#특징변수와 종속변수 분리
x = df.drop(['day'], axis = 1)
y = df['day']

#방주형변수 원핫인코딩 후 연속형변수와 합침
x1 = x[['PAY', 'LateCount', 'Store_sum']]
x2 = x.drop(['PAY', 'LateCount', 'Store_sum'], axis = 1)
x2 = x2.astype('category')
x2 = pd.get_dummies(x2, drop_first=True)
x = pd.concat([x1, x2], axis=1)

#전체 데이터의 80%를 테스트셋으로 분리
tr_sum = y.sum()
x_train = x.loc[0:tr_sum*0.8]
y_train = y.loc[0:tr_sum*0.8]
x_test = x.loc[tr_sum*0.8:tr_sum*1.0]
y_test = y.loc[tr_sum*0.8:tr_sum*1.0]

```

도면15

```

#GRADIENT BOOSTING CLASSIFIER
from sklearn.ensemble import GradientBoostingClassifier

model_gbc = GradientBoostingClassifier(random_state=99)
model_gbc.fit(x_train, y_train)
predict_gbc = model_gbc.predict(x_test)
print("acc:", model_gbc.score(x_test, y_test))
print("auc:", roc_auc_score(y_test, predict_gbc))
print("recall:", metrics.recall_score(y_test, predict_gbc))

```

도면16

| STORE_SEQ | EMP_ID | START | Rate |
|-----------|--------|------------|------|
| 800 | 779 | 2018-07-03 | 0.56 |
| 1244 | 1323 | 2018-07-03 | 0.78 |
| 686 | 674 | 2018-07-20 | 0.12 |
| 799 | 777 | 2018-07-20 | 0.98 |
| 799 | 808 | 2018-08-07 | 0.75 |
| 1052 | 1276 | 2018-08-07 | 0.4 |
| 815 | 823 | 2018-08-11 | 0.48 |
| 1052 | 1277 | 2018-08-08 | 0.89 |
| 1052 | 1275 | 2018-08-11 | 0.23 |
| 1011 | 1103 | 2018-09-28 | 0.49 |
| 541 | 470 | 2018-10-01 | 0.93 |
| 886 | 868 | 2018-10-19 | 0.35 |
| 778 | 874 | 2018-10-19 | 0.16 |
| 1008 | 1089 | 2018-10-20 | 0.36 |
| 886 | 854 | 2018-11-01 | 0.52 |
| 387 | 195 | 2018-11-01 | 0.89 |
| 387 | 672 | 2018-11-01 | 0.53 |
| 387 | 323 | 2018-11-01 | 0.79 |
| 1011 | 1079 | 2018-11-01 | 0.83 |

도면17

```

import pandas as pd
import numpy as np
from pandas import DataFrame
from sklearn.metrics import roc_auc_score
from sklearn import metrics
%matplotlib inline
import matplotlib.pyplot as plt
from scipy.stats import chi2_contingency

df_Late_Rate = pd.read_csv('F:/Python/Wesop/Data/지각확률.csv', engine='python')
df_Rate = pd.read_csv('F:/Python/Wesop/Data/장기근속확률.csv', engine='python')
df_manager_score = pd.read_csv('F:/Python/Wesop/Data/점주평가.csv', engine='python')

score=(1-df_Late_Rate)+df_Rate+df_manager_score

```

도면18

| EMP_ID | LateRate | | Rate | manager_score | score |
|--------|----------|------|------|---------------|-------|
| 779 | CALENDAR | 0.5 | 0.56 | 3 | 4.06 |
| 1323 | | 0.61 | 0.61 | 3 | 4.17 |
| 674 | CALENDAR | 0.5 | 0.12 | 3 | 3.62 |
| 777 | | 0.15 | 0.15 | 3 | 4.83 |
| 808 | CALENDAR | 0.5 | 0.75 | 3 | 4.25 |
| 1276 | | 0.12 | 0.12 | 3 | 4.28 |
| 823 | | 0.12 | 0.12 | 3 | 4.36 |
| 1277 | | 0.12 | 0.12 | 3 | 4.77 |
| 1275 | | 0.12 | 0.12 | 3 | 4.11 |
| 1103 | | 0.14 | 0.14 | 3 | 4.35 |
| 470 | | 0.67 | 0.67 | 3 | 4.26 |
| 868 | | 0.06 | 0.06 | 3 | 4.29 |
| 874 | CALENDAR | 0.5 | 0.16 | 3 | 3.66 |
| 1089 | | 0.06 | 0.06 | 3 | 4.3 |
| 854 | | 0.15 | 0.15 | 3 | 4.37 |
| 195 | | 0.05 | 0.05 | 3 | 4.84 |
| 672 | | 0.06 | 0.06 | 3 | 4.47 |
| 323 | | 0.07 | 0.07 | 3 | 4.72 |
| 1079 | | 0.06 | 0.06 | 3 | 4.77 |

도면19

