

## ■ 채용공고 자동 필터링 시스템 - 필터링 조건 상세 문서

작성자: 이재륜

(filter\_core\_v2.py 기준 / Core Logic 설명서)

본 문서는 filter\_core\_v2.py에서 수행되는 모든 검증 항목과 판정 기준을 정리한 개발자용 설명 문서이다.

각 행에 대해 총 9개 검증 항목을 검사하며, 개별 결과를 종합하여 최종 판정(OK / 要確認 / NG)을 결정한다.

### 1. 전체 판정 로직 개요

각 행마다 아래 9개 체크 항목을 개별 판정한다

- (1) 필수항목(check\_required)
- (2) 이메일(check\_email)
- (3) 고용형태(check\_employment)
- (4) 직종(check\_job\_title)
- (5) 취업처 회사명 특수기호(check\_company\_special)
- (6) 소개원 회사명 특수기호(check\_intro\_company\_special)
- (7) 비공개 규칙(check\_private\_intro)
- (8) 시구읍면 문자깨짐(check\_city\_garbled)
- (9) 최저임금(judge\_min\_wage)

판정 우선순위는 다음과 같다

- (1) 하나라도 NG일 경우 -> 최종 NG
- (2) NG 없고 要確認가 존재할 경우 -> 최종 要確認
- (3) 전부 OK일 경우 -> 최종 OK

### 2. 필수 항목 존재 여부 검사

검사 대상 컬럼은 다음과 같다

- (1) 就業先会社名 (취업처 회사명)
- (2) 雇用形態 (고용형태)
- (3) 職種 (직종)
- (4) 応募先メールアドレス (응모지 메일 주소)
- (5) 市区町村 (시구읍면)
- (6) 給与形態 (급여형태)
- (7) 給与下限 (급여하한)

판정 기준은 다음과 같다

- (1) 모든 필수 컬럼 값이 존재할 경우 -> OK
- (2) 하나라도 공란일 경우 -> NG

### **3. 응모 이메일 형식 검사**

검사 내용은 다음과 같다

- (1) 이메일 컬럼 공란 여부
- (2) 복수 이메일 분리 가능 (, ; 공백 줄바꿈 모두 허용)
- (3) 각 이메일이 정규식에 맞는지 확인

판정 기준은 다음과 같다

- (1) 모든 이메일 형식이 정상일 경우 -> OK
- (2) 이메일이 공란일 경우 -> NG
- (3) 하나라도 형식 오류가 있을 경우 -> NG

### **4. 고용형태 허용 여부 검사**

"正社員", "契約社員", "派遣社員", "パート", "アルバイト", "アルバイト・パート", "アルバイト/パート", "アルバイト / パート", "アルバイト、パート", "パート・アルバイト", "パート/アルバイト", "パート / アルバイト", "パート、アルバイト", "業務委託"

- (1) 허용 리스트에 있는 값과 정확히 일치할 경우 -> OK
- (2) 공란일 경우 -> NG
- (3) 허용되지 않은 표기일 경우 -> NG

"(株)", "(株)", "(株)", "(有)", "(有)", "(有)"

- (1) 특수기호를 포함하지 않을 경우 -> OK
- (2) 공란일 경우 -> NG
- (3) 특수기호를 포함할 경우 -> NG

- (1) 공란일 경우 -> OK
- (2) 특수기호 포함하지 않을 경우 -> OK
- (3) 특수기호를 포함할 경우 -> NG

## 7. 취업처회사명이 비공개일 경우 규칙 검사

기본 규칙은 다음과 같다

취업처회사명(就業先会社名) = 非公開인 경우 소개원회사명(紹介元会社名)가 필수이다

판정 기준은 다음과 같다

- (1) 취업처회사명이 非公開가 아니면 → OK
- (2) 취업처회사명이 非公開이고, 소개원회사명이 공란인 경우 -> NG
- (3) 취업처회사명이 非公開이고, 고용형태가 派遣社員이고, 소개원회사명이 非公開인 경우 -> NG
- (4) 취업처회사명이 非公開이고, 고용형태가 派遣社員이고, 소개원회사명이 공란이 아니고 非公開도 아닌 경우 -> OK
- (5) 취업처회사명이 非公開이고, 고용형태가 派遣社員가 아니고, 소개원회사명이 공란이 아닌 경우 -> OK (소개원회사명이 非公開여도 OK하는 이유: ?)

## 8. 시,구,읍,면 문자 깨짐 검사

검사 대상은 다음과 같다

市区町村

문자 깨짐의 판단 기준은 다음과 같다

- (1) 문자열에 공백을 포함할 경우 -> NG
- (2) 제어 문자(Wx00~ Wx1F범위)를 포함할 경우 -> NG
- (3) 깨짐으로 판단되는 특정 문자("闖|驥|伴")를 포함할 경우 -> NG

## 9. 직종 적합성 검사

검사 내용은 다음과 같다

- (1) 공란 여부
- (2) 도도부현명 포함 여부

- (3) 지명 형식 포함 여부
- (4) 모집 조건, 근무 조건 혼입 여부
- (5) 숫자 포함 여부

따라서 판정 기준은 다음과 같다

- (1) 공란일 경우 -> NG
- (2) 지명 또는 도도부현을 포함할 경우 -> NG
- (3) 모집/근무/고용 조건 키워드를 포함할 경우 -> 要確認
- (4) 숫자를 포함할 경우 -> 要確認
- (5) 그 이외 -> OK

정의된 도도부현 명칭은 다음과 같다

“北海道, 青森, 岩手, 宮城, 秋田, 山形, 福島, 茨城, 栃木, 群馬, 埼玉, 千葉, 東京, 神奈川, 新潟, 富山, 石川, 福井, 山梨, 長野, 岐阜, 静岡, 愛知, 三重, 滋賀, 京都, 大阪, 兵庫, 奈良, 和歌山, 鳥取, 島根, 岡山, 広島, 山口, 徳島, 香川, 愛媛, 高知, 福岡, 佐賀, 長崎, 熊本, 大分, 宮崎, 鹿児島, 沖縄”

지명 형식은 다음과 같다

区/市/町/村/駅

직종 문자열이 너무 짧은 경우(길이 3 미만), 区/市/町/村/駅와 같은 단일 한자 포함으로 인한 오탐지를 방지하기 위해 지명 형태 검사를 수행하지 않는다. 이는 의미 없는 단일 문자 또는 깨진 데이터가 지명 포함 직종으로 잘못 판정되는 것을 막기 위한 안전 장치이다.

모집/근무/고용 조건 키워드는 다음과 같다

"募集", "急募", "大募集", "積極採用", "オープニング", "新規", "正社員", "契約社員", "派遣社員", "アルバイト", "パート", "業務委託", "夜勤", "日勤", "深夜", "早朝", "交替", "シフト", "残業", "未経験", "経験不問", "学歴不問", "資格不問", "〇〇不問", "歓迎", "優遇", "高収入", "日払い", "週払い", "即日", "短期", "長期", "寮", "社宅", "在宅", "リモート", "テレワーク", "マネージャー", "リーダー", "部長", "課長", "係長", "主任", "候補"

## 10. 최저임금 판정

최저임금 판정을 위해 사용하는 컬럼은 다음과 같다

- (1) 都道府県 ( addressRegion ) → col\_pref
- (2) 勤務地住所 → col\_address
- (3) 市区町村 ( addressLocality ) → col\_city
- (4) 職種 → col\_job
- (5) 就業先会社名 → col\_work\_company
- (6) 給与形態 ( unitText ) → col\_wage\_unit
- (7) 給与下限 ( minValue ) → col\_wage\_lower

### 10-(1) 도도부현 결정 로직

우선순위1: 도도부현 컬럼이 MIN\_WAGE에 정확히 존재하면 그대로 사용한다 (여기서 MIN\_WAGE란 미리 정의된 도도부현과 그에 따른 최저임금 리스트를 말한다)

우선순위2: 아래 텍스트들에서 PREF\_RE로 도도부현을 추출한다

- 주소(addr)
- 시구읍면(city)
- 직종(job)
- 회사명(comp)

### 10-(2) 급여 단위(unitText)로 시급을 환산한다

현재 unitText 매핑은 다음과 같다

UNIT\_MAP = {1:"HOUR", 2:"DAY", 3:"MONTH", 4:"YEAR", 5:"WEEK"}

환산 상수는 고정값으로 다음과 같다

- DAY 환산: 1일 = 8시간
- MONTH 환산: 1개월 = 160시간

- YEAR 환산: 1년 = 1920시간

환산 규칙은 다음과 같다

- HOUR: 그대로
- DAY: amount / 8
- MONTH: amount / 160
- YEAR: amount / 1920
- WEEK: 환산하지 않음 → hourly=None
- 그 외(UNKNOWN): hourly=None

#### 10-(3) 최저임금 판정 분기

- step1: 도도부현 확보 실패시 -> NG
- step2: unitText 결측/파싱 실패시 -> NG
- step3: minValue 결측/파싱 실패시 -> NG
- step4: 주급인 경우 -> 要確認
- step5: 그 외 환산 불가인 경우 -> NG
- step6: 최저임금과 비교하여 hourly >= minw 인 경우 -> OK
- step7: 최저임금과 비교하여 hourly < minw 인 경우 -> NG