

# 교통사고 심각도(Lethal) 예측을 위한 데이터 분석 및 모델링 보고서

Data Science Portfolio

작성자: 송은희

작성일: 2026-01-27

# 1. 프로젝트 개요

## 1.1 문제 정의

교통사고 데이터에서 사고의 심각도를 예측하는 문제는 단순한 분류 성능을 넘어 실제 위험 상황을 얼마나 놓치지 않는지가 중요한 과제이다. 교통사고 심각도 데이터에서 다루는 **Lethal** 사고는 발생 빈도는 낮지만 사회적, 인적 피해가 매우 크며 예측 관점에서 이를 놓칠 경우 실질적인 안전 대응에 한계가 발생한다.

본 프로젝트는 교통사고 발생 시 사고가 **치명적(Lethal)**으로 이어질 가능성을 사전에 예측하는 것을 **목표**로 하며 단순한 정확도 향상보다는 위험 신호를 최대한 포착하는 방향의 분류 전략에 초점을 두었다.

## 1.2 데이터 출처

Kaggle에서 제공된 **RoadSense** 교통사고 데이터셋을 사용하였다.

해당 데이터는 실제 교통사고 사례를 기반으로 구성되어 있으나, 대회용 데이터 특성상 학습용 데이터에는 다양한 노이즈가 포함되어 있다. 또한 사고 발생 시점의 환경 정보, 사고 관련 인원 정보, 차량 및 도로 특성 등 사고의 심각도에 영향을 줄 수 있는 다양한 변수를 포함하고 있다..

본 프로젝트에서는 대회에서 제공된 학습용 데이터셋을 기반으로 전처리, 피처 엔지니어링, 모델링을 수행하였다.

## 1.3 프로젝트 목표

Kaggle 대회에서 제시된 평가지표인 **F1-score**를 기본 지표로 사용하 단순히 점수를 높이는 것에만 집중하지 않았다.

실제 교통사고 예측 문제의 특성을 고려할 때 치명적인 사고를 정상 사고로 잘못 분류하는 경우(**False Negative**)는 상대적으로 더 큰 위험을 초래할 수 있다.

이에 따라 본 프로젝트의 목표는 다음과 같다.

- **F1-score**를 기준으로 한 전반적인 분류 성능 유지
- **Lethal** 사고에 대한 재현율(**Recall**) 개선

- False Negative를 최소화하는 방향의 피처 선택 및 모델 설계

본 프로젝트는 성능이 좋은 모델이 아니라 위험 신호를 놓치지 않는 모델을 설계하는 데 중점을 두었다.

## 2. 데이터 이해

### 2.1 데이터 구성

본 프로젝트에서 사용한 RoadSense 데이터셋은 교통사고 단위로 구성된 정형 데이터로 사고의 발생 환경, 도로 및 차량 특성, 사고 관련 인원 정보 등을 포함하고 있다.

각 사고는 하나의 AccidentId로 식별되며 타깃 변수인 Gravity는 사고의 심각도를 나타낸다.

본 분석에서는 이 중 치명적 사고(Lethal) 여부를 이진 분류 문제로 정의하였다.

데이터는 크게 다음과 같은 범주로 구성되어 있다.

- 사고 발생 환경 : Accidents Dataset
- 도로 및 인프라 정보 : Places Dataset
- 사고 관련 인적 정보 : Users Dataset
- 사고 차량 및 유형 정보 : Vehicles Dataset

### 2.2 타깃 변수 분포

타깃 변수인 Gravity는 사고의 심각도를 범주형으로 나타내며 이를 Lethal / Non-Lethal의 이진 형태로 재구성하였다. 전체 사고 중 Lethal 사고의 비율은 약 0.055%로 명확한 클래스 불균형(class imbalance)을 가지는 데이터이다.

이와 같은 분포 특성으로 인해 단순 정확도(Accuracy) 기준의 모델 평가는 실제 위험 사고를 제대로 반영하지 못할 가능성이 크다. 예를 들어 대부분의 사고를 Non-Lethal로 예측하더라도 겉보기 성능은 높게 나타날 수 있으나 정작 중요한 Lethal 사고를 놓치는 문제가 발생할 수 있다.

### 2.3 클래스 불균형과 평가 지표의 한계

본 데이터셋의 클래스 불균형 특성은 모델 학습 및 평가 과정에서 중요한 제약 조건으로 작용한다.

특히 **Lethal** 사고를 **Non-Lethal**로 잘못 분류하는 경우(**False Negative**)는 현실적인 관점에서 더 큰 위험을 내포한다.

이에 따라 본 프로젝트에서는 **Accuracy** 대신 **F1-score**를 주요 성능 지표로 사용하였으며, 동시에 다음과 같은 관점에서 모델 성능을 해석하였다.

- **Lethal** 클래스에 대한 재현율(**Recall**)
- **False Negative** 발생 여부
- 정밀도와 재현율 간의 균형

이를 통해 단순한 수치 경쟁이 아닌 **실제 적용 가능성을 고려한 모델 평가 기준을 설정**하였다.

이러한 데이터 특성은 이후 피처 엔지니어링 및 모델링 전략을 설계하는 데 중요한 기준으로 활용되었다.

## 3. 분석 전략 및 접근 방식

### 3.1 초기 접근 방식과 한계

초기 분석 단계에서는 사고 심각도에 영향을 줄 수 있다고 판단한 다양한 피처 엔지니어링을 선행한 후, 이를 한 번에 모델에 적용하는 방식으로 분석을 진행하였다.

이러한 접근은 전반적인 모델 성능을 빠르게 확인하는 데에는 유용했으나, 개별 피처가 모델 성능에 기여하는 정도나 어떤 피처가 실제로 의미 있는 신호로 작용하는지를 해석하기에는 한계가 있었다.

특히 이 방식에서는 특정 피처가 성능 개선에 기여했는지 혹은 오히려 노이즈로 작용했는지를 명확히 구분하기 어려웠다.

### 3.2 분석 전략의 전환

이와 같은 한계를 인식한 이후, 본 프로젝트에서는 분석 전략을 **기초 모델(Base Model) 중심의 점진적 검증 방식**으로 전환하였다.

먼저 최소한의 핵심 변수만을 사용하여 기초 모델을 구성한 후 피처를 하나씩 추가하면서 모델 성능의 변화를 관찰하였다. 이를 통해 단순히 성능이 높아지는지 여부가 아니라 각 피처가 **Lethal 사고 예측에 실제로 기여하는 신호인지**를 성능 변화 기반으로 평가하고자 하였다.

### 3.3 피처 단위 검증 전략

피처 엔지니어링 과정에서는 각 피처를 독립적인 실험 단위로 간주하였다.

새로운 피처를 추가할 때마다 **F1-score**, 재현율, **False Negative** 변화 여부를 함께 확인함으로써 모델 성능의 개선 여부뿐 아니라 위험 신호 포착 관점에서의 기여도를 중심으로 피처를 평가하였다.

이를 통해 성능 개선 효과가 불분명하거나 해석이 어려운 피처는 과감히 제외하고 의미 있는 신호를 제공하는 피처만을 최종 모델에 반영하였다. - (현재의 처리전략으로 수정 될 수 있음.)

### 3.4 도메인 기반 전처리 기준

전처리 단계에서는 데이터가 생성된 맥락을 고려한 도메인 기반 해석을 적용하였다.

예를 들어 성별 정보는 사고 기록 상 반드시 존재해야 하는 정보로 판단하였으며, **Unknown** 또는 결측값은 실제로 값이 존재하지 않는 범주가 아니라 **기록 과정에서 누락된 정보**로 해석하였다.

이에 따라 데이터의 의미를 기준으로, 결측을 유지할지 혹은 다른 라벨로 재분류할지에 대한 판단 기준으로 활용하였다.

### 3.5 전략의 범위와 한계

본 프로젝트에서는 피처 엔지니어링과 모델 성능 해석에 중점을 두었으며, 결측치 처리 방식 자체를 다양하게 비교·실험하는 단계까지는 진행하지 않았다.

향후 분석에서는 결측치 처리 방식에 따른 성능 변화 역시 추가적인 실험 대상으로 고려할 수 있을 것으로 판단된다.

## 4. 데이터 전처리 처리 전략 (Data Preprocessing Strategy)

### 4.1 전처리의 목표와 접근 관점

본 프로젝트에서 전처리는 단순히 모델 학습을 위한 사전 단계가 아니라 **데이터의 의미적 정합성을 복원하는 핵심 분석 과정**으로 정의하였다.

RoadSense 데이터는 프랑스 교통사고 기록을 기반으로 한 대규모 정형 데이터로 동일한 개념을 표현하는 값들이 다양한 언어 표현, 대소문자, 단위 차이, 오타 등의 형태로 파편화되어 존재하였다.

이러한 상태에서의 모델 학습은 의미적으로 동일한 정보를 서로 다른 범주로 인식하게 만들어 모델 성능 저하 및 해석 불가능성을 유발할 수 있다고 판단하였다. 이에 따라 데이터의 ‘결’을 이해하여 의미 단위로 정제한다는 관점에서 전처리 전략을 수립하였다.

이 과정은 단순한 결측치 제거가 아니라 데이터가 생성된 맥락을 고려한 **도메인 기반 정합성 확보**에 초점을 두었다.

### 4.2 범주형 데이터 정합성 확보 (Category Cleaning)

#### 4.2.1 문제 인식

범주형 변수(Category 계열)에서는 동일한 의미의 값이 서로 다른 문자열로 다수 존재하는 문제가 반복적으로 확인되었다.

이륜차(Scooter) 관련 변수

- Scooter 50cm
- Scooter<50cm
- scooter 50
- Scooter 50 cc
- SCOOTER-50

위와 같이 표기 방식, 단위, 대소문자 차이로 인해 사실상 동일한 개념이 여러 개의 카테고리로 분리되어 있었다. 이는 모델 입장에서 동일한 신호를 여러 개의 희석된 피처로 인식하게 만들며 결과적으로 학습 효율과 일반화 성능을 저하시킬 수 있는 구조였다.

#### 4.2.2 처리 전략

본 프로젝트에서는 이러한 문제를 해결하기 위해 **정규표현식(Regex)** 기반의 **표준화 전략**을 적용하였다.

- 대소문자 통일
- 불필요한 공백 및 특수문자 제거
- 단위 표현(cm, cc 등) 통합
- 의미 단위 기준의 카테고리 재정의

이를 통해 다수의 파편화된 문자열 표현을 의미적으로 일관된 **단일 표준 카테고리**로 통합하였다.

#### 4.2.3 Before vs After 비교 - Date

사고 발생 일자를 나타내는 **Date** 변수는 동일한 의미의 날짜가 서로 다른 문자열 포맷으로 혼재되어 있었다.

- 날짜 포맷 혼재
- 월(Month)이 문자열 약어 Jan, Feb, Mar 등 표현
- 시간 정보가 불필요하게 포함된 경우
- 구분자가 일관되지 않음

##### [Before → After 정리 표]

Before	After
24/01/2018	2018-01-24
Feb 12 2018	2018-02-12
04-Mar-18	2018-04-03

2018-11-30 00:00:00	2018-11-30
12-Jul-18	2018-07-12

#### Data 전처리 결과 요약

- 다양한 날짜 문자열 포맷을 **단일 기준(YYYY-MM-DD)**으로 통합
- 날짜 파생 변수(month, weekday 등) 생성 가능 상태 확보
- 문자열 기반 노이즈 제거로 **시간 관련 피처 엔지니어링 안정성 확보**

본 처리는 단순 형식 변환이 아니라 데이터가 생성된 연도와 의미적 맥락을 기준으로 날짜 정보를 복원하고 정합성을 확보하는 과정이었다.

#### 4.2.4 Before vs After 비교 - Hour

Hour 변수는 사고 발생 시간을 나타내지만 학습 데이터에는 서로 다른 시간 포맷이 혼재되어 있었다.

- 초 이상의 포맷 존재 (HH:MM:SS:xx)
- AM / PM 포맷 혼재
- train과 test 간 시간 표현 방식의 불일치

#### [Before → After 정리 표]

Before	After
11:35:00:20	11:35:00
06:30:00:58	06:30:00
15:30:00:44	15:30:00

5:35 PM	17:35:00
3:57 PM	15:57:00

#### Hour 전처리 결과 요약

- 서로 다른 시간 포맷을 단일 포맷(HH:MM:SS)으로 통합
- 시간 관련 피처 생성(예: hour\_group) 가능 상태 확보
- train/test 입력 정합성 확보로 모델 학습 안정성 강화

Hour의 전처리는 시간 정보의 단순 변환을 넘어 모델 입력 정합성을 확보하고 시간 기반 분석의 신뢰도를 높이기 위한 과정이었다.

#### 4.2.5 Before vs After 비교 - Light (category 데이터 대표)

Light 변수는 사고 당시 조도 상태를 나타내는 변수이나 대소문자 표기 불일치로 인해 동일한 의미의 값이 여러 범주로 분리되어 있었다.

- 대소문자 혼재
- 동일한 개념 다른 범주로 인식할 가능성 존재
- 카테고리 희석 및 성능 저하 우려

#### [Before → After 정리 표]

Before	After
daylight, Datlight, DAYLIGHT	Daylight
twilightordawn, TWILIGHTORDAWN, TwilightOrDawn	TwilightOrDawn

nightstreelightson, NIGHTSTREELIGHTSON, NightStreelightsOn	NightStreelightsOn
nightnostreetlight, NIGHTNOSTREETLIGHT, NightNoStreetLight	NightNoStreetLight
nightstreelightsoff, NIGHTSTREELIGHTSOFF, NightStreelightsOff	NightStreelightsOff

#### Light 전처리 결과 요약

- 표기 방식이 다른 값들을 **5개의 의미 단위 카테고리**로 통합
- 동일 개념의 카테고리 파편화를 제거하여 모델 학습 효율성 개선
- 조도 기반 피쳐 엔지니어링(예: 야간/가로등 조건 파생 변수) 기반 마련

Light 전처리는 표기 정규화 수준을 넘어 카테고리 의미 단위를 복원하여 모델 입력의 정합성과 해석 가능성을 확보하는 과정이었다.

이외에도 피쳐 엔지니어링 과정 전반에서 **Test** 데이터와의 정합성을 유지하는 방향으로 전체 데이터셋 전처리를 수행하였다.

### 4.3 결측치 및 Unknown 값 처리 기준

결측치 처리 또한 기계적인 대체 방식이 아닌 **변수의 의미와 데이터 생성 맥락을 기준으로 판단**하였다.

예를 들어 **성별(Gender)** 변수의 경우 사고 기록 상 반드시 존재해야 하는 정보로 판단하였으며, **Unknown** 또는 결측값은 실제 값이 존재하지 않는 범주라기보다는 **기록 과정에서 누락된 정보**로 해석하였다.

#### 결측치 처리 전략

- 결측값을 무조건 제거하지 않는다.
- 의미적으로 **Unknown**을 유지해야 하는 경우는 별도의 범주로 보존한다.
- 모델이 결측 자체를 하나의 신호로 인식할 수 있도록 처리 기준을 명확히 정의한다.

이러한 접근은 데이터 손실을 최소화하면서 현실 데이터의 불완전성을 모델링 단계에 반영하기 위한 전략적 선택이었다.

## 4.4 전처리 전략의 결론

본 프로젝트의 전처리 과정은 단순한 데이터 정제가 아니라 분석 전반의 기준을 설정하는 핵심 과정으로 정의하였다.

모델이 동일한 의미의 신호를 일관되게 해석할 수 있도록 정합성 확보에 초점을 맞추었으며 의미가 불분명한 값은 제거의 대상이 아니라 해석의 대상으로 판단하였다.

전처리는 모델 성능에 직접적인 영향을 미치는 단계이기도 하지만 동시에 모델 결과에 대한 신뢰도를 확보하기 위한 필수적인 과정으로 설계하였다. 모델을 잘 학습시키기 위한 준비 단계가 아니라 데이터가 가진 의미를 복원하고 책임 있게 전달하는 과정으로 인식하였다.

이러한 전처리 기준은 이후 피쳐 엔지니어링 및 모델링 전략을 설계하는 데 있어 중요한 판단 기준으로 활용되었으며 본 프로젝트의 핵심 강점 중 하나로 작용하였다.

## 5. 피쳐 엔지니어링 및 실험 결과

(실험 결과 내용 한줄 정리 입력)

### 5.1 베이스라인 기초 모델 설정

본 프로젝트는 피쳐의 기여도를 해석 가능하게 만들기 위해 먼저 최소 변수로 구성된 베이스라인을 설정한 뒤 점진적으로 변수를 추가하는 방식으로 실험을 진행하였다.

**기초 입력 변수 : Hour, Weather, SurfaceCondition, Vehicle\_count\_user, Safety\_used\_yes\_count, Safety\_used\_no\_count, Persons**

이 베이스라인은 복잡한 피쳐 엔지니어링을 하기 전에 최소 신호로 어디까지 가능한지를 확인하기 위한 기준점으로 사용되었다.

## 5.2 Date 파생 변수 실험

사고 발생 시점의 신호를 반영하기 위해 Date에서 파생 변수를 생성하여 실험하였다.

해당 실험에서는 month, weekday, is\_weekend 포함 여부를 비교하며 성능을 확인하였다.

특히 is\_weekend는 직관적으로 의미가 있을 수 있는 변수지만, 실험 과정에서 제거 후 성능 변화를 별도로 검증하였다. 해당 실험 결과(ROC-AUC, F1 포함)는 다음과 같이 기록되었다.

### [Date 관련 변수 모델 결과표]

실험ID	변경피처	TN	FP	FN	TP	F1-score	PR-AUC	ROC-AUC
exp_02	Date, Hour 변수 추가	8747	285	504	28	0.06627	0.06995	0.55456
exp_03	month, weekday, is_weekend 생성	8365	667	472	60	0.09531	0.07176	0.55359
exp_04	is_weekend 변수 제거	8302	730	469	63	0.09509	0.07201	0.55426

실험 결과 is\_weekend 변수를 추가하였을 때 F1-score 및 ROC-AUC의 유의미한 개선은 확인되지 않았으며, FN 관점에서는 오히려 제거 후 성능이 소폭 개선되는 경향을 보였다.

month, weekday 변수만으로도 사고 발생 시점의 시간적 특성이 충분히 반영되었음을 의미하며, is\_weekend는 중복 신호로 판단하여 최종 모델 변수에서 제외하였다.

## 5.3 Hour 파생 변수 실험

-정리예정-

## 5.4 Light 파생 변수 실험

-정리예정-

## 5.5 Maneuver 파생 변수 실험

-정리예정-

## 5.6 정리 -

## 6. 모델링 및 평가 전략

### 6.1 LightGBM 선택 이유

사고 심각도 예측을 위한 분류 모델로 LightGBM (Light Gradient Boosting Machine)을 사용하였다. LightGBM을 선택한 이유는 다음과 같다.

첫째, 본 데이터는 범주형 변수와 수치형 변수가 혼재된 정형(Tabular) 데이터로 구성되어 있으며, LightGBM은 이러한 데이터 구조에서 안정적인 성능을 보이는 트리 기반 앙상블 모델이다.

둘째, 사고 심각도 예측 문제는 변수 간의 비선형 관계와 상호작용 효과가 중요한 영역으로 선형 모델보다 트리 기반 모델이 이러한 패턴을 포착하는 데 유리하다고 판단하였다.

셋째, LightGBM은 대규모 데이터에 대한 학습 효율이 높고, 피처 중요도 및 분기 구조를 통해 해석 가능성을 확보할 수 있으며 불균형 데이터 환경에서도 비교적 안정적인 학습이 가능하다는 장점이 있다.

결론적으로 모델 성능뿐 아니라 분석 결과 해석과 실험 반복 가능성을 고려하여 LightGBM을 주요 모델로 선택하였다.

### 6.2 실험 설계 방식

모델 학습 및 평가는 앞서 정의한 분석 전략에 따라 **점진적 실험 설계 방식**으로 진행하였다.

먼저 최소한의 핵심 변수로 구성된 베이스라인 모델을 설정한 후 피처를 하나씩 추가하거나 제거하면서 모델 성능의 변화를 단계적으로 비교하였다.

각 실험에서는 다음 지표를 함께 확인하였다.

- F1-score (대회 평가 지표)
- Lethal 클래스에 대한 재현율(Recall)
- False Negative(FN) 변화
- ROC-AUC 및 혼동행렬

이를 통해 단순히 점수가 상승했는지 여부가 아니라 **치명 사고를 얼마나 놓치지 않는지**를 중심으로 모델 성능을 해석하고자 하였다. 특히 성능 변화가 미미한 경우에는 추가된 피처가 실제로 의미 있는 신호를 제공하는지 혹은 기존 피처와 중복된 정보를 제공하는지를 함께 고려하여 피처 채택 여부를 결정하였다.

## 6.3 Threshold 처리 전략

전처리 및 피처 엔지니어링, 모델 구조에 대한 실험을 충분히 수행한 이후 모델이 최종 실험 완료 후 최종 단계에서 **threshold** 조정을 적용하는 전략을 채택하였다.

이는 **threshold** 조정이 모델이 학습한 신호 자체를 개선하는 방법이 아니라 이미 학습된 예측 확률에 대해 판단 기준을 조정하는 후처리 단계라는 점을 고려한 결정이었다. 따라서 초기 실험 단계에서 **threshold**를 임의로 조정하기보다는 기본 **threshold(0.5)**를 기준으로 피처 구성 및 모델 구조에 따른 성능 차이를 충분히 검증하였다.

이후 최종 모델이 확정된 시점에서 예측 확률 **predict\_proba**을 기반으로 **threshold**를 조정하여 **False Negative**를 최소화하는 방향으로 성능을 재조정하였다. **Threshold** 조정 과정에서는 **F1-score**뿐 아니라 **Lethal** 클래스에 대한 재현율(**Recall**)과 **False Negative** 감소 여부를 함께 고려하여 모델의 실질적인 위험 탐지 성능을 평가하였다.

이를 통해 단순히 점수를 높이는 것이 아니라 **치명 사고를 놓치지 않는** 방향으로 모델의 의사결정 기준을 최종 보정하였다.

## 7. 결과

### 7.1 성능 비교 표

### 7.2 FN / FP 해석

### 7.3 중요 피처 인사이트

## 8. 결론 및 한계

### 8.1 얻은 인사이트

## 8.2 실제 적용 범위 ( 의미 ) - 실제 업무 바탕

## 8.3 개선할 수 있는 방향