한국정보전자통신기술학회논문지(jkiiect)'18-10, Vol.11 No.5
논문 18-11-05-543 http://dx.doi.org/10.17661/jkiiect.2018.11.5.543

예측모델 구축을 위한 분석 단계별 레이아웃 표준화 연구

김효관* 황원용**

Analysis Standardization Layout for Efficient Prediction Model

Hyo-Kwan Kim* Won-Yong Hwang**

요 약 예측의 중요성은 경제상황의 급변 등으로 점차 중요해지고 있다. 예측 모델을 구현하기 위해서는 다수의 데이터 엔지니어와 사이언티스트들이 프로젝트에 참여하게 된다. 이때 모델을 고도화 시키기 위하여 다양한 예측모델 아이디어가 제시된다. 제시된 모든 아이디어의 유효성을 판단하기 위하여 예측 모델에 대한 정확도를 테스트하고 유효하지 않은 경우 다시 모델을 재설계하고 개발하는데 오랜 시간이 소요된다. 본 연구에서는 다양한 아이디어를 하나의 모델로 통합하는 가장 효율적인 방법론을 찾기 위하여 레고 형태의 개발 방법론을 제시한다. 레고 형태의 개발방법론은 각 아이디어에 대한 개발 코드에 대하여 동일한 데이터 레이아웃을 설정함에 따라 가능하다. 따라서 아이디어 별로 유효성 검증이 가능하고 레고 형태로 개발함에 따라 아이디어의 추가 및 삭제가 용이하여 전체 개발공정 시간을 단축할 수 있다. 마지막으로 제시한 방법이 실제 아이디어의 추가/삭제가 용이한지 개발 및 테스트를 수행하였다.

Abstract The importance of prediction is becoming more emphasized, due to the uncertain business environment. In order to implement the predictive model, a number of data engineers and scientists are involved in the project and various prediction ideas are suggested to enhance the model. it takes a long time to validate the model's accuracy. Also It's hard to redesign and develop the code. In this study, development method such as Lego is suggested to find the most efficient idea to integrate various prediction methodologies into one model. This development methodology is possible by setting the same data layout for the development code for each idea. Therefore, it can be validated by each idea and it is easy to add and delete ideas as it is developed in Lego form, which can shorten the entire development process time. Finally, result of test is shown to confirm whether the proposed method is easy to add and delete ideas.

Key Words : prediction, data engineer, data scientist, data layout, development method

1. 서론

경제 상황의 급변, 소비 경향의 변화, 대체기술의 부상 등 불확실한 경영환경으로 다양한 산업 분야에서 예측, 고객 분석 등의 데이터 분석모델 구축을 시도하여 안정적인 경영환경을 유지하려고 노력하고 있다.

이때 하나의 모델을 구축하기 위하여 다수의 데이터 엔지니어 및 사이언티스트가 프로젝트에 투입되어 개발함에 있어 일치되지 않는 다양한 예측모델 아이디어가 제시된다[1-3]. 제시된 모든 아이디어를 모델에 반영하기 위해서는 모델의 정확도 확인 및 유효성 판단 후 개발 모델 수정을 위하여 많은 시간이 소요되어 전체 개발공정 시간이 길어지게 된다.

그림 1은 예측을 위한 표준 흐름(수집->정제->표준예측) 이후 다양한 산업특화 로직으로 인한 이슈를 나타낸 그림이다.

* Corresponding Author: Department of Fintech Korea Polytechnics (haiteam@kopo.ac.kr)
** Department of Fintech Korea Polytechnics
Received September 07, 2018 Revised September 10, 2018 Accepted October 01, 2018

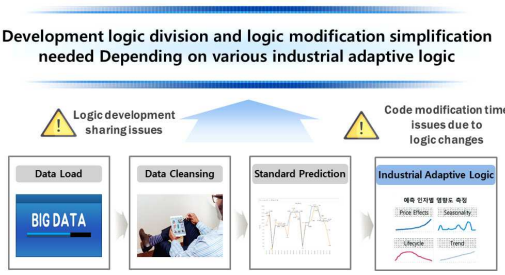

그림 1. 산업 특화로직 표준화 필요성
Fig. 1. Necessity of industrial standardization logic

예측 모델을 개발하기 위하여 표준화된 알고리즘은 다양한 연구 분야에서 제시되어왔다[4-6]. 하지만 예측모델은 고도화를 위하여 계속 새로운 아이디어가 추가적으로 제시되어야 하나 아직 효율적인 개발 방법론에 대해서는 정의되지 않고 있다.

본 연구에서는 예측분야의 분석모델을 구현하기 위하여, 먼저 기본이 되는 표준 예측모델을 정의한 후 모델 고도화를 위한 다양한 아이디어를 하나의 분석 모델에 구현하기 위한 가장 효율적인 방법을 연구하였다. 결론적으로 각 아이디어에 대한 데이터 레이아웃 표준화를 통하여 독립적으로 개발 및 정확도 테스트가 가능하고, 레고 형태의 구조를 갖추어 상황에 따라 자유롭게 아이디어 로직을 추가하거나 제거할 수 있는 방법론을 제시하였다.

2. 본론

2.1 연구배경 및 방법

예측 모델 프로젝트는 시스템 설계의 경우 프로젝트에 투입된 모든 인원이 동시에 참여하여 설계가 가능하다, 하지만 분석 모델 구현의 경우 데이터 사이언티스트 1명이 주도적으로 모델을 개발하지 않는 한 다수의 엔지니어가 참여하여 아이디어를 제시하고 검증하기 위해 많은 개발/테스트 시간이 소요된다.

예측모델 설계 시, 실제 정의한 표준 예측모델 외에 다양한 산업에 특화된 로직이 필요하다. 이때 제시된 다수의 아이디어에 대해서 개발하고 검증하면서 동시에 다른 아이디어를 검증하기에는 많은 개발공정 시간이 소요된다.

따라서 본 논문에서는 프로그램 설계 시점에 다양한 산업특화 로직을 개발자들이 자유롭게 아이디어를 설계하고 독립적으로 개발 및 검증이 가능한 구조를 설계하여 개발된 아이디어별로 정확도 검증이 가능하고 로직의 추가 및 삭제가 자유로운 방법론을 그림 2과 같이 제시하였다.

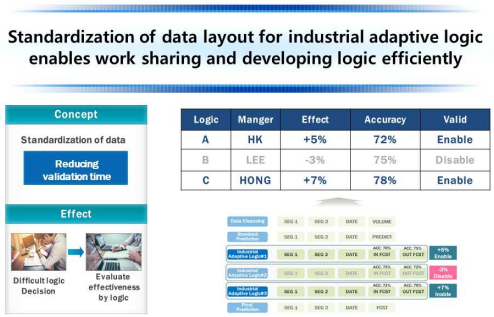

그림 2. 데이터 레이아웃 표준화
Fig. 2. Data layout Standardization

연구 방법은 그림 3과 같이 빅데이터 분석 프레임워크인 스파크를 활용하여 간단히 테스트를 하였다.

그림 3. 연구 방법
Fig. 3. Method of Research

2.2 이론적 고찰

2.2.1 예측 기법

예측은 대상 예측기간에 따라 장기예측, 중기예측, 단기예측으로 분류될 수 있다. 일반적으로 장기예측은 예측대상 기간이 2년 이상인 경우로 고려할 수 있으며 제품기획, 능력계획, 입지결정 등 주로 전략적 의사결정과 관련된 경우에 활용 된다.

장기 예측은 예측 기간이 길기 때문에 환경예측에 근거한 주관적 판단이 많이 이용되며 정확도가 상대적으로 낮다. 중기예측은 보통 6개월에서 2년을 대상 기간으로 하며 계량적 접근이 가능하고 전문가의 의견도 많은 도움이 된다. 단기예측은 보통 6개월 이내의 분기별, 월별, 주별, 일별예측을 말하며 상대적으로 정확한 예측이 가능하다.

또한 수치를 이용한 계산방법이 중심이 되는가 안 되는가에 따라 크게 정성적 예측기법(qualitative method)과 정량적 예측기법(quantitative method)으로 분류된다. 정량적 예측기법은 다시 인과형 예측기법(causal forecasting method)과 시계열 예측기법(time series analysis)으로 분류된다.

정성적 예측기법에는 델파이(delphi)법, 시장조사법(market research), 판넬동의법(panel consensus) 등이 있으며 전문가나 외부기관으로부터의 정보가 중요하게 활용될 수 있다.

타임시리즈 예측기법은 과거의 수요를 분석하여 시간에 따른 수요의 패턴을 파악하고 이의 연장선상에서 미래의 수요를 예측하는 정량적 단기 예측방법이다. 즉 과거 수요의 흐름으로 부터 미래의 수요를 투영하는 방법으로서 과거의 수요패턴이 미래에도 지속된다는 시장의 안정성이 기본적인 가정으로 필요하다. 그러나 과거의 수요패턴이 항상 계속적으로 유지된다고 할 수 없으므로 시계열 예측기법은 주로 중, 단기 예측에 이용되며 적은 자료로도 비교적 정확한 예측이 가능하다. 이동평균법, 지수평활법 등 실제 예측분야에서 패턴을 찾기 위해 사용된다.

그림 4는 수요 거래량에 대한 패턴을 원, 계절요인 추세순환, 불규칙요인으로 패턴을 그래프로 시각화한 내용이다[6].

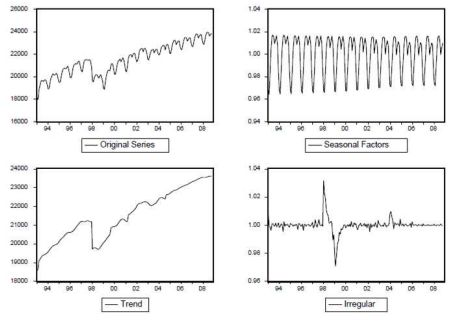

그림 4. 시계열 그래프 파형
Fig. 4. Time Series Graph

추세요인은 보통 5년 이상의 동일한 상승 또는 하강하는 추세를 보이는 장기적인 패턴을 말한다. 추세변동은 짧은 기간에 순간적으로 나타나는 패턴이 아니라 장기적인 추세경향이 나타나는 현상으로 직선이나 부드러운 곡선의 연장선으로 표시한다.

순환요인은 특정기간동안 반복하는 주기적인 변동을 말한다. 보통 3년 정도 일정한 기간을 주기로 순환 패턴이 보이며 시간의 흐름에 따라 반복되는 패턴이 추세 선을 따라 변화하는 것이 순환변동이다.

계절요인은 일정한 기간 동안 반복되는 기후, 수요, 생활 패턴 등에 따라 나타나는 변동이다. 보통 계절에 따라 순환하며 반복적으로 패턴이 생성되는 특성을 지니고, 순환변동과 다른 점은 순환 주기가 짧아 예측분야에 많이 활용된다.

불규칙 요인은 패턴으로 감지할 수 없는 이벤트, 전쟁, 파업, 급격한 사회변화 등에 의한 변동 뿐 만 아니라, 명절과 요일의 차이 등에 의한 변동으로 어떠한 규칙성이 없이 예측이 불가능한 순간적으로 발생하는 변동을 말한다.1-6).

2.2.2 분산처리 프레임워크 (Spark)

스파크는 UC Berkeley의 AMP(Algorithms Machines People) 랩에서 개발 하였으며, 하둡의 맵리듀스 작업 중 디스크 기반으로 처리하여 디스크의 병목현상이 발생되던 부분을 인 메모리 기반으로 효율화하고 데이터 분석 작업에 용이한 인 메모리 환경의 분산처리 플랫폼이다. 스파크는 2013년 아파치의 인

큐베이터 프로젝트로 채택되고 2014년 정식으로 1.0 버전을 출시하였다. 스파크는 대용량 데이터를 빠르게 연산하기 위한 오픈소스 분산처리 플랫폼으로써 RDD(Resilient Dataset) 개념을 사용하며, 데이터 중 일부를 분산 노드들의 메인 메모리에 적재하여 반복 연산 시 분산된 노드들에 일을 분배 한 후 작업이 완료되는 경우 다시 합치는 작업을 수행함으로써 데이터 작업 수행 속도를 높이도록 하였다. 그림 5는 스파크의 기본 구조 및 기계학습의 한 애플리케이션인 로지스틱 회귀분석(Logistic Regression) 알고리즘을 100GB의 데이터를 가지고 수행 시 하둡과 비교했을 때 10배 이상 빠르다는 점을 그래프로 보여준다[7].

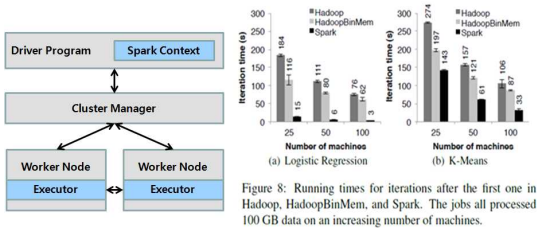

그림 5. 스파크 클러스터 구조 및 속도비교
Fig. 5. Spark Cluster & Performance

그림 6은 스파크 코어에서 분산처리가 가능하다는 부분을 그림으로 표현하였다.

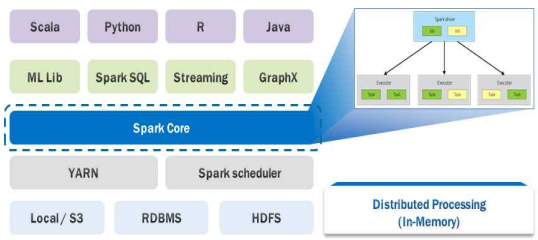

그림 6. 스파크 아키텍처
Fig. 6. Architecture of Spark

2.3 예측모델 개발공정

본 연구에서는 데이터분석을 위한 개발공정을 다음과 같이 정의하였다.

계획 수립 단계에서는 개발의 전체 공정 및 계획을 수립하고 프로젝트에 가장 적합한 개발환경 및 실행개념을 정의한다.

기본설계 단계에서는 바로 수집 가능한 데이터와 외부에서(웹 크롤링 등) 활용할 수 있는 데이터 인터페이스를 정의한 후 실제 분석모델의 큰 그림인 분석모델 개략도를 설계한다.

상세설계 단계에서는 분석모델을 구체화하여 각 분석모델 단계별 프로세스다이어그램 및 모델을 코딩 가능한 수준으로 구체화한다.

개발/테스트 단계에서는 개발환경 설정 및 실제 설계한 문서를 바탕으로 개발 및 각 단위 별 단위 테스트 후 단위 테스트가 완료되면 모델 내 통합 후 통합테스트를 진행한다.

이 외에 실제 정확도 확인 및 모델 보완 및 배포를 통한 운영이관 등의 단계가 있지만 본 연구에서는 개발/테스트 까지만 정의하였다.

표 1. 소프트웨어 개발 공정
Table 1. Software Development Process

Process	Activities
Planning	Establish development process and progress management plan
Preliminary Design	Data search and Design interface and execution structure
Critical Design	Model Design for each step
Implement/Test	Implementation and Unit/Integration Test

2.4 분석모델 표준레이아웃 설계

분석 모델측면에서만 보면 데이터 불러오기, 데이터 전처리, 예측 모델 구현, 결과 저장 순으로 진행된다.

여기서 예측 모델 구현은 표준 로직 및 다양한 산업특화 로직으로 구현된다.

이때 산업특화로직을 단계 별로 그림 7과 같이 설계 및 구현하면 블록 형태로 각 로직이 독립적으로 설계 및 테스트가 가능하여 팀원 간 업무 역할을 정확하게 분담하여 병렬적으로 수행 가능하다.

그림 7. 업무의 병렬화
Fig. 7. Parallelization of tasks

또한 블록 형태의 설계로 인하여 그림 8과 같이 구현된 로직을 입력 시점의 정확도와 출력 시점의 정확도를 비교하여 유효성을 판단하여 자유롭게 추가 및 삭제가 가능하다.

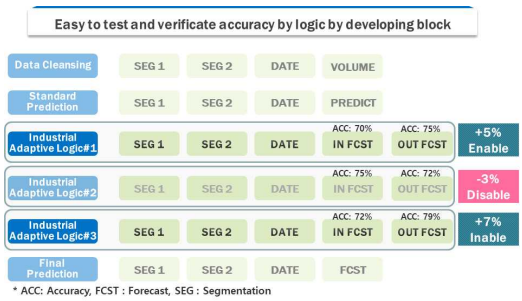

그림 8. 독립적인 로직 구현 및 검증
Fig. 8. Independent logic implementation/validation

단 한 개의 로직만 남게 되어도 프로그램이 구동되는데 전혀 에러가 발생하지 않는다. 또한 각 단계별 예측 값 및 실제 값을 과거 시점 기준으로 예측모델을 시뮬레이션 하여 개별적으로 정확도를 검증 가능하다.

2.5 분석모델 설계/테스트

테스트를 위해 제품 예측모델을 테스트로 구동하였다. 모델은 표준 로직인 타임시리즈 예측모델을 기본으로 입력 값으로 특화로직 2개를 구현하였다.

첫 번째 로직은 프로모션 효과적용 로직으로 예측 주차에 프로모션이 존재하는 경우 프로모션 비율을 적용하여 수요예측 값을 조정하는 로직으로 구현하였다.

두 번째 로직은 과거 동 주차 제품 별 예측 값을 보정하는 로직으로 과거 동 주차에 100만 건이 판매되었는데 예측 값이 너무 낮거나 높은 경우에 예측 값을 보정하는 로직으로 구현하였다.

그림 9와 같이 두 개의 로직을 동일한 데이터 레이아웃으로 구성하고 프로그램을 구동한 후 하나의 로직을 제거하였을 때 에러나지 않음을 확인하였다.

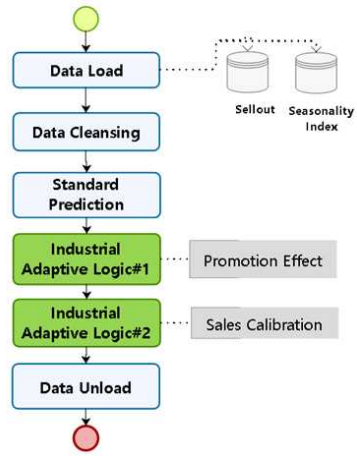

그림 9. 분석모델 프로세스 다이어그램
Fig. 9. Process Diagram of Model

로직은 변경 가능하지만 팀원 간 자유롭게 로직을 구현하고, 독립적으로 검증한 후 자유롭게 로직을 추가 및 삭제 가능하도록 설계 하였다.

테스트로 그림10과 같이 스파크 환경에서 로직의 추가 및 삭제 시에도 전혀 에러가 발생하지 않고 결과가 산출됨을 확인할 수 있다.

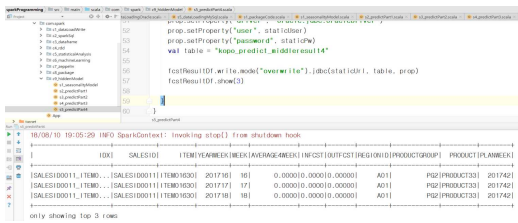

그림 10. 테스트 수행결과 화면
Fig. 10. Result of Prediction Model

산업 특화로직 1번만 적용 했을 때의 결과는 그림 11과 같다.

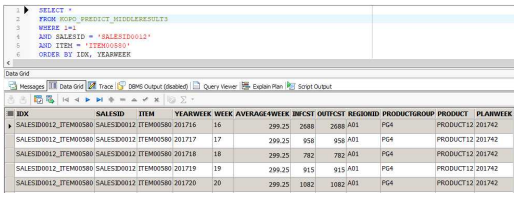

그림 11. 산업 특화로직 1번 적용한 결과화면
Fig. 11. Result of Prediction (Applied #1 logic)

추가로 산업특화로직 2번만 적용 했을 때도 에러 없이 결과 정상 산출됨을 그림 12를 통해 확인할 수 있다.

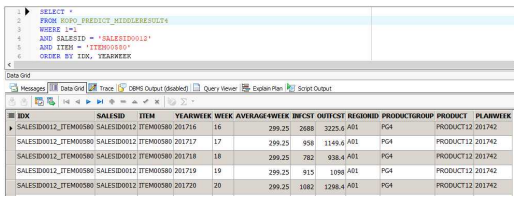

그림 12. 산업 특화로직 2번 적용한 결과화면
Fig. 12. Result of Prediction (Appled #2 logic)

3. 결론

본 논문은 예측모델을 구현함에 있어 다양한 수요예측 모델을 효율적으로 프로그램 내에서 관리할 수 있도록 각 분석 단계별 인/아웃 데이터 레이아웃을 동일하게 설계하였다.

장점으로는 제안한 특화로직을 독립적으로 정확도 테스트가 가능하며, 예측모델 정확도 향상에 불필요하다면 즉시 삭제 하 수 있도록 데이터 레이아웃 측면에서 표준화 할 수 있도록 설계 하여 기존 개발 방식과 다르게 생산성 및 코드관리가 표 2와 같이 용이함을 확인할 수 있다.

표 2. 개발방법론 비교
Table 2. Development methodology comparison

Target	Original Method (Spiral)	Proposed Method
Code revision	Code modification and other impact analysis required	Each code block individually manageable
Accuracy calculation	Hard	Each code block can be calculated individually

앞으로의 연구과제는 다양한 산업 특화로직을 상황에 따라 추가 및 삭제가 용이하여 개발자 모두 자신의 로직을 병렬적으로 개발할 수 있고, 각 로직에 대한 정확도를 개별적으로 테스트/검증 할 수 있는 장점 외에 고객측면에서 분석모델을 직접 시뮬레이션 할 수 있는 환경을 설계하여 개발한 로직을 쉽게 조합하여 실행 가능한 테스트 벤치의 표준화 연구가 필요하다.

REFERENCES

[1] Taylor, James W, "An evaluation of methods for very short-term load forecasting using minute-by-minute British data," international Journal of Forecasting, vol.24, no.4, pp.645-658, 2008.
[2] Taylor, James W, "Triple seasonal methods for short-term electricity demand forecasting," European Journal of Operational Research, vol.204, no.1, pp.139-152, 2010.
[3] SongGyungBin, "An Algorithm of Short-Term Load Forecasting, vol.10, no.53A, pp.529-535, 2004.
[4] HyoKwanKim, "A study on the advanced demand forecasting system using big data technique", 2017
[5] SeoMyungYul, "A Study on the Seasonal Adjustment of Time Series and Demand Forecasting for Electronic Product Sales," koras, vol.3, no.1, pp13-39, 2003.
[6] NamGyunHeo, "A Study on Air Demand Forecasting Using Multivariate Time Series Models" Communications for Statistical Applications and Methods, vol.22, no.5, pp1007-1017, 2009.
[7] Berkeley, http://www.eecs.berkeley.edu/Pubs/TechRpts/2011/

저자약력

김 효 관(Hyo-Kwan Kim) **[정회원]**

- 2001년 ~ 2007년: 성균관대학교 정보통신공학부 학사
- 2012년 ~ 2014년: 한국교통대학교 컴퓨터공학과 석사
- 2015년 ~ 2017년: 한국교통대학교 컴퓨터공학과 박사
- 2011년 ~ 2017년: 삼성 SDS 빅데이터 솔루션(Brightics)개발
- 2018년 현재 : 한국폴리텍대학 서울강서캠퍼스 스마트금융과 교수

※ 관심분야 : 금융데이터 분석, 핀테크

황 원 용(Won-Yong Hwang) **[정회원]**

- 2004년 ~ 2007년: LG전자 MC연구소 휴대폰개발
- 2007년 ~ 2008년: LIG넥스원 PGM연구센터 지대공유도무기 개발
- 2009년 ~ 2011년: 한국과학기술원 공학석사
- 2011년 ~ 2017년: 삼성SDS EFSS개발
- 2018년 현재 : 한국폴리텍대학 서울강서캠퍼스 스마트금융과 교수

※ 관심분야 : 블록체인, 핀테크