**“SW프로젝트”**

**요약서, 결과보고서 및 성과물 증빙 제출 안내**

**1. 양식의 푸른색 글씨는 작성요령으로 제출 시에는 삭제 후 제출**

**2. 본 문서에 포함되어있는 2개의 양식들을 모두 작성하여야 함**

**2-1**. “SW프로젝트 요약서”

**2-2.** “SW프로젝트 결과보고서”

**3. 성과물 증빙 (있는 경우에 한함)**

**3-1.** 논문 발표 실적이 있다면, 아래 3가지의 사본을 모두 결과보고서 뒤에 첨부

**가.** 학회 표지

**나.** 본인 이름이 있는 목차

**다.** 사사표기(감사의 글), 논문제목, 저자가 있는 논문 표지 사본

**3-2.** 특허출원 및 소프트웨어등록 실적이 있다면, 아래 문서들의 사본을 결과보고서 뒤에 첨부

**가.** 등록번호가 있는 프로그램 등록증 (소프트웨어등록의 경우)

**나.** 출원인 및 (가)출원번호가 있는 특허명세서 (특허출원의 경우)

**다.** 특허출원 및 소프트웨어등록의 경우 반드시 행정팀 안내를 받은 후 진행

**문의**: 졸프조교 (seniorproj2023hy@gmail.com)

**SW프로젝트 요약서**

|  |  |
| --- | --- |
| **프로젝트 기간** | **2023.03.02. - 2023.11.30. (총09개월)** |
| **프로젝트 팀원** | **고관우(컴퓨터소프트웨어학부, 4학년)**  **오주영(컴퓨터소프트웨어학부 4학년)** |
| **지도교수** | **채동규 교수님** |
| **프로젝트 명** | **Clustering을 활용한 주가예측** |
| **프로젝트 내용** | 이름, 목적, 수행내용, 결과  **프로젝트 이름**  Clustering을 활용한 주가예측  **프로젝트 목적**  기존의 주가 데이터들을 활용해 다음날 주식의 가격 예측  **프로젝트의 수행내용**  기존의 논문들은 하나의 주식의 데이터 혹은 해당 주식의 정보를 기반으로 RNN으로 주가를 예측했습니다. 저희는 이런 방식에서 벗어나 다양한 주식들의 데이터를 결합해 주가를 예측해보려 했습니다. 특히 주식의 그래프가 변동성을 반영한다고 생각해 모든 주식들의 그래프 기반으로 주가를 예측했습니다. |
| **기대효과 및 개선방향** | 이런식으로 clustering을 활용한 주가 예측은 기존에 논문에서 찾아볼 수 없었습니다. 단일 주식의 데이터가 적어 해당 주식으로 주가를 예측할 수 없을 때 clustering을 통해 관련주를 찾고 해당 주식으로 기존의 주식을 예측할 수 있다고 생각합니다.  다만 저희가 사용한 방식은 여전히 주가 자체의 데이터가 부족하다고 생각합니다. 특히 기본적으로 주가는 하루 단위로 움직이는 양이 적어 변하지 않는다고 가정하는 것이 오히려 정확한 경우가 많았습니다. 때문에 비율관계를 활용해서 주가에 곱을 하는 저희의 계산 방식이 주가 자체가 큰 경우에 대해 부정확한 결과를 낳았습니다. 하지만 주가 자체가 작은 경우 상당히 유사한 결과를 도출했습니다. |

|  |  |
| --- | --- |
| **SW프로젝트 결과보고서** | |
| **프로젝트명** | **Clustering을 활용한 주가예측** | |
| **프로젝트**  **요약** | **프로젝트 목적**  기존의 주가 데이터들을 활용해 다음날 주식의 가격 예측  **프로젝트의 수행내용**  기존의 논문들은 하나의 주식의 데이터 혹은 해당 주식의 정보를 기반으로 RNN으로 주가를 예측했습니다. 저희는 이런 방식에서 벗어나 다양한 주식들의 데이터를 결합해 주가를 예측해보려 했습니다. 특히 주식의 그래프가 변동성을 반영한다고 생각해 모든 주식들의 그래프 기반으로 주가를 예측했습니다.  **결과**  CSV 파일에 모든 주식들에 대한 예측가격, 실제 가격 그리고 전체 주식들에 대한 RMSE를 기록했습니다. | |
| **프로젝트**  **기간** | **2023.03.02. - 2023.11.30. (총09개월)** | |
| **산출물** | 졸업 작품 ( O ), 졸업 논문 ( )  (해당 내용은 동그라미 표시) | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **학과** | **학번** | **학년** | **이름** | **연락처** |
| 컴퓨터소프트웨어 | 2020060100 | 4 | 고관우 | [kwko97@hanyang.ac.kr](mailto:kwko97@hanyang.ac.kr)  010-7345-1831 |
| 컴퓨터소프트웨어 | 2020059152 | 4 | 오주영 | [jason7599@hanyang.ac.kr](mailto:jason7599@hanyang.ac.kr)  010-7599-1214 |
|  |  |  |  |  |

**목 차**

1. 프로젝트 개요

1.1 프로젝트 목적 및 배경

1.2 프로젝트 최종 목표

2. 프로젝트 내용

3. 프로젝트의 기술적 내용

4. 프로젝트의 역할 분담

4.1 개별 임무 분담

4.2 개발 일정

5. 결론 및 기대효과

1. 프로젝트 개요

1.1 프로젝트 목적 및 동기

기존 논문들은 RNN, SVM 등등의 모델을 활용해 주가의 변동을 예측합니다. 이런 방식에서 벗어나, 저희는 여러 주식간의 상관관계와 유사도에 집중하고자 했습니다. 즉 주식 하나하나의 변동을 예측하기보다 관련주와 적대주 등의 관계를 파악해 이를 기반으로 전체 주식장의 움직임을 예측해보자는 도전을 했습니다. 뉴스나 온라인 커뮤니티의 의견을 스크레이핑한 뒤 자연어처리 기술로 예측을 하는 것이 아닌, 순수하게 관련 주식들의 변동을 통해 예측을 하는 방식입니다. 기본적으로 주식 시장은 사람의 투자를 기반으로 움직이기 때문에 그래프만으로도 어느정도 정확한 결과를 도출해낼 수 있을 것이라고 생각했습니다.

1.2 프로젝트 목표

저희는 하나의 주식에 대해 다음날 주가를 예측하는 것이 아닌 전체 주식시장의 데이터를 가져와 모든 주식들의 다음날 주가를 예측할 수 있도록 했습니다. 이를 위해서 모든 주식 데이터를 가져오는게 우선이었습니다. 그래서 api를 제공하는 라이브러리를 찾아보던 중 FinanceDataReader를 알게되어 이를 활용해 모든 주식 데이터를 가져왔습니다. 주식 데이터를 받아온 뒤 이를 벡터로 처리하기 위해 변화비율(Change)을 numpy에 넣어 처리하고 하나의 벡터로 만들었습니다. 그 다음으로 이런 벡터들끼리의 관계를 파악하기 위해 distance를 정의했습니다. 이 때 논문에서 찾은 correlation을 활용한 공식을 적용했습니다. Distance 공식까지 파악한 뒤에는 KMeans Clustering을 구현했습니다. 각 cluster 마다 주식들이 들어있고 해당 주식들의 Change값을 활용해 주가를 예측하도록 했습니다. 이를 바탕으로 실제 주식의 값과 예측한 주식의 값을 비교했습니다.

2. 프로젝트 내용

처음 프로젝트를 시작할 때 딥러닝, 인공지능에 대한 지식이 아예 없었습니다. 이런 지식들을 쌓기 위해 1학기 기간 동안 딥러닝및응용 수업과 데이터사이언스 수업을 듣고 주제를 결정해야겠다고 생각했습니다. 이 때 딥러닝및응용 수업에서 이론적인 부분을 학습하고 pytorch 프레임워크에 대한 학습이 필요하겠다고 생각해 네이버에서 제공하는 “모두를 위한 딥러닝”을 통해 pytorch 프레임워크를 학습했습니다. 하지만 기존의 논문들이 이미 DNN, RNN 등의 방식으로 주가를 예측하는 알고리즘이 있었고 다른 방식을 활용해 이를 구현하고 싶었습니다. 마침 데이터사이언스 수업 때 배운 Clustering 알고리즘을 활용해 관련주, 적대주의 관계를 통해 주가를 예측할 수 있지 않을까 생각했고 실제로 이를 활용한 논문들이 없어서 해당 주제로 결정했습니다.

기본적으로 주가 예측 알고리즘에 대한 지식이 전무해 다양한 논문들을 통해 어떤 알고리즘이 활용되는지 공부했습니다. 논문을 읽어보면서 기존의 알고리즘들은 주가 예측을 어떤 방식으로 하는지, distance는 어떻게 사용했는지 등을 파악하는데 집중했습니다. 특히 저희는 Clustering 알고리즘을 사용하는 만큼 distance 공식과 어떤 Clustering 알고리즘을 사용할 건지 중요했습니다. 마침 논문에서 distance 공식을 봤고 단순히 Euclidian 방식을 적용하는 것보다 correlation을 활용한 공식이 성능이 더 좋을 것이라고 생각했습니다. 또한 여러 Clustering 알고리즘에 대해 조사해봤지만 KMeans Clustering의 성능이 좋다는 글을 봐서 KMeans Clustering 알고리즘을 채택했습니다. 하지만 적절한 K값을 찾기 힘들다는 한계가 있어서 이를 명확하게 증명하는 것이 어려울 것이라고 생각했습니다.

저희가 프로젝트 진행 과정 중 가장 어려웠던 점은 Label이 없는 것 이었습니다. 특히 Clustering 알고리즘은 비지도학습이어서 Label 없이 Clustering은 됐지만 주가를 어떻게 예측해야 될지 몰랐습니다. 실제로 비지도학습에서 Label이 없는 문제는 여전히 학계에서도 다뤄지고 있는 문제이고 이를 해결한 논문이 없었습니다. 그래서 저희가 생각해낸 방식은 동일한 Cluster 내의 주식들의 Change 값을 활용해 저희만의 수식을 세워 다음날 결과를 예측했습니다.

3. 프로젝트의 기술적 내용

주가 데이터를 불러오는 방법으로, 처음에는 Kaggle 등의 웹사이트의 데이터를 미리 csv 형태로 저장해둔 뒤, 프로그램이 이를 불러와 활용하는 방식을 고려했습니다. 하지만, 프로그램이 이런 파일들에 대한 의존성이 생길 뿐 아니라 이런 데이터는 시간이 흐름에 따라 최근 정보를 포함시키지 않는다는 문제점이 있었습니다. 따라서 저희는 FinanceDataReader 라이브러리를 통해 주가 데이터를 불러오는 방식을 채택했습니다.

불러온 주가 데이터를 저희 프로그램에 적합하게 수정해주는 Data Cleaning 작업이 필요했습니다. 학습을 진행할 지정한 날짜 범위를 포함하지 않는 주식들, 이 범위 내에 빠진 값이 있는 주식들은 입력에서 제외해주었습니다. 뿐 아니라 기간 내에 상장이 폐지된 주식들이 포함되는 문제가 있었습니다. 이런 주식들은 전날 대비 변동률이 0으로 채워져 있어 distance와 similarity를 계산할 수 없다는 문제를 일으켰습니다. 따라서 저희는 각 주식의 Volume(거래량) 값이 0을 포함할 경우 비교군에서 제외했습니다.

클러스터링 알고리즘으로는 SOM(Self Organizing Map), DBSCAN 등의 알고리즘을 고려해 보았으나, 결국 단순하면서도 성능이 증명된 K-Means 알고리즘을 채택하였습니다. 초기 시드 포인트는 단순히 무작위 값이 아닌 실제 주가의 데이터를 샘플링하도록 구현하였습니다. Distance 메트릭으로는 PCC(Pearson Correlation Coeffecient)를 활용하였습니다. 시간 간격에 여유를 주는 DTW(Dynamic Time Warping)을 적용해보기도 하였으나, 오히려 더 안 좋은 클러스터링 결과를 가져오는 것을 볼 수 있었습니다. 처음에는 아무런 라이브러리 없이 순수 파이썬 코드만으로 구현을 해보았으나, 학습할 기간과 주식의 수가 커지자 수행 시간이 매우 느려지는 문제에 직면하였습니다. 이를 완화하기 위해 scipy 라이브러리의 cdist 메서드를 활용해 코드상의 explicit for문을 다수 없앴으며, 이를 통해 수행 시간을 크게 단축할 수 있었습니다. 또한 iteration마다 이전의 것에서 label 변화가 없는 경우 종료하도록 코드를 작성한 것도 수행 시간의 큰 단축을 가져다 주었습니다.

클러스터링 결과를 기반으로 실제 예측을 하기 위해서 사용한 알고리즘은 다음과 같습니다. 예측하는 주가는 같은 클러스터 내의 다른 한 주식의 최종적으로 학습된 날의 변동률과, 이 주식과의 similarity를 곱해 누적하는, aggregate 방식입니다. 이 합산된 값이 곧 예측하고자 하는 주가의 예측 변동률이 됩니다. 이때 클러스터 크기가 예측값에 영향을 주지 않도록 비율관계를 적용했습니다. 이렇게 이웃 주식들의 최종적으로 학습된 날의 변동률을 사용함은, 다음과 같은 가정을 전제했습니다. 한 주식의 어떤 특정 날짜에 관측된 변동률은, 그 전 날짜의 변동률과 비슷할 것이라는 가정입니다. 예를 들어 전날 폭등하던 주식은 오늘도 상승세를 보일, 또 반대로 전날 폭락하던 주식은 오늘도 하락세를 보일 가능성이 높다고 판단하는 것입니다.

정확도를 측정하는 방식으로는, 처음에는 변동률에 대한 예측값과 실제값의 부호의 일치 여부를 사용하는 것을 고려하였습니다. 즉 상승, 하락 두 경우만을 따지는 방식이었습니다. 하지만 이런 방식은 예측 정확성을 판단하기에는 부족하다는 결론을 내렸습니다. 실제 변동률이, 예를 들어, +100%인 반면 예측 변동률이 +0.1%인 경우, 정확하다고 볼 수 없겠으나 이런 메트릭에 따르면 잘 예측한 것이 될 것입니다. 혹은 +0.1% 변동률을 보인 주식에 대해 -0.1%를 예측한 경우 나름 유사하게 예측을 했다고 볼 수 있겠지만 이런 메트릭에 의하면 틀린 예측이 됩니다. 따라서 이런 단순한 상승 하락을 따지는 것이 아닌, 실제 값들과 예측 값들을 각각의 벡터로 두어 이 사이 RMSE를 측정하는 방식을 활용하였습니다. 이런 하나의 RMSE 값만으로는 정확성을 판단할 수 없기에, 다른 “Baseline” 모델을 두어 이 모델의 예측값의 RMSE와 비교하는 방식을 채택하였습니다. 이때 Baseline 모델이라 함은 저희 모델과 비교를 하기 위한 단순한 예측 모델로, 주식의 마지막으로 학습된 주가 N일의 평균으로 예측을 하는 등의 기본적인 방식으로 작동합니다. 저희 모델이 이런 Baseline 모델들보다 더 좋은 성능, 즉 더 작은 RMSE를 내는 것을 목표로 구현을 하였습니다.

4. 프로젝트의 역할 분담

4.1 개별 임무 분담

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **번호** | **학과** | **학번** | **학년** | **이름** | **담당업무** |
| 1 | 컴퓨터소프트웨어 | 2020060100 | 4 | 고관우 | 코드 작성 및 공부 |
| 2 | 컴퓨터소프트웨어 | 2020059152 | 4 | 오주영 | 코드 작성 및 공부 |
| 3 |  |  |  |  |  |

4.2 개발 일정

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **구분** | **추진내용** | **수행기간** | | | | | | | | | | | |
| **1월** | **2월** | **3월** | **4월** | **5월** | **6월** | **7월** | **8월** | **9월** | **10월** | **11월** | **12월** |
| 계획 | 딥러닝, 데이터사이언스 공부 |  |  |  |  |  |  |  |  |  |  |  |  |
| 분석 | 논문을 통해 주가 예측 알고리즘 공부 |  |  |  |  |  |  |  |  |  |  |  |  |
| 설계 | KMeans Clustering 알고리즘 |  |  |  |  |  |  |  |  |  |  |  |  |
| 개발 | 순수 python 코드로 전부 구현 -> 라이브러리를 활용해 코드 개선 |  |  |  |  |  |  |  |  |  |  |  |  |
| 테스트 | 실제 주가와 예측한 주가의 RMSE를 통해 파악 |  |  |  |  |  |  |  |  |  |  |  |  |
| 종료 | 보고서 작성 |  |  |  |  |  |  |  |  |  |  |  |  |

5. 결론 및 기대효과

AI에 대해서 거의 몰랐지만 이번 졸업 프로젝트를 통해 AI 기술에 대해 공부하는 시간을 가졌습니다. 특히 다양한 논문을 읽어보면서 주가 예측의 성능이 기대와 달리 성능을 좋게 가지는 것이 많이 어려운 점을 배웠습니다.

아무것도 모른 채로 주제가 재밌어 보여 시작했는데 프레임워크를 활용해 기술적으로 구현하기가 어려웠습니다. 그래서 최대한 저희도 할 수 있는 내용이면서 남들이 하지 않던 아이디어를 찾고자 했습니다.

그래서 여러 논문들을 읽어보면서 timeseries clustering을 활용한 논문들이 없어서 이걸 해보면 좋지 않을까 싶어서 해당 아이디어를 채택했습니다.

주가 예측 프로그램을 구현하면서 timeseries Clustering에 대한 정보가 빈약에 저희가 수식을 생각해 구현한 내용도 있었는데 이런 부분에 대해 좀 더 정보가 많았으면 성능이 더 좋아지지 않을까 생각했습니다.

저희가 활용한 방식은 거의 모든 주식들에 대해 적용할 수 있었고 논문에 작성된 단일 주식의 데이터가 적어 해당 주식으로 주가를 예측할 수 없을 때 clustering을 통해 관련주를 찾고 해당 주식으로 기존의 주식을 예측할 수 있다고 생각합니다.

하지만 저희가 사용한 방식은 여전히 주식 자체의 데이터가 부족하다고 생각합니다. 특히 국내 주식들이 모두 합해야 3000개 이내이고 이를 Clustering에 활용하기에는 데이터의 양이 부족하다고 생각합니다. 그래서 주식 개수가 실제로 증가한다면 더 좋은 결과를 낳을 수 있지 않을까 생각했습니다. 실제로 주식 개수를 1000개만 쓰는 것보다 2000개를 썼을 때 더 정확한 성능을 도출해 냈고 해당 부분이 아쉬웠습니다.

또한 기본적으로 주가는 하루 단위로 움직이는 양이 적어 변하지 않는다고 가정하는 것이 오히려 정확한 경우가 많았습니다. 때문에 비율관계를 활용해서 주가에 곱을 하는 저희의 계산 방식이 주가 자체가 큰 경우에 대해 부정확한 결과를 낳았습니다. 하지만 주가 자체가 작은 경우 상당히 유사한 결과를 도출했습니다.