**분류**- 분류는 주어진 데이터의 특징을 활용하여 레이블, 카테고리, 클래스 등을 예측하는 작업

**26.1 활용 사례**- 신용리스크 예측  
- 뉴스기사 분류  
- 사용자 행위 분류  
- 현재 사내 감성 분석, 스팸 필터링 등도 모두 분류에 해당

**26.2 분류 유형**- 이진 분류  
 - 분류할 레이블 종류가 2개인 경우로, 사기거래여부, 스팸여부 등에 해당  
- 다중 클래스 분류  
 - 분류할 레이블 종류가 3개 이상인 경우로, 다범주 분류라고도 함. 3개 이상의 레이블 중 하나에 속하는 경우에 해당  
- 다중 레이블 분류  
 - 한 객체가 여러 범주에 해당할 수 있는 경우로, 이미지 객체 인식 등에 해당

**26.3 MLlib의 분류 모델**- Spark는 기본적으로 다중 레이블 분류를 지원하지 않으며, 다중 레이블 모델을 학습시키려면 레이블당 하나의 모델을 학습시킨 후 수동으로 조합해야 함  
- Spark에서 제공하는 대표적인 분류 모델은 아래와 같음  
 - 로지스틱 회귀  
 - 의사결정트리  
 - 랜덤 포레스트  
 - 그래디언트 부스티드 트리  
 - 다층퍼셉트론  
 - 선형 서포트 벡터 머신  
  
- 모델 확장성  
 - 일반적으로 스파크는 대용량의 머신러닝 모델을 학습시키는데 많은 지원을 함

|  |  |  |  |
| --- | --- | --- | --- |
| 모델 | 최대 특징 수 | 최대 학습 데이터 수 | 최대 타깃 범주 수 |
| 로지스틱 회귀 | 1 ~ 1천만개 | 제한 없음 | 특징 수 x 클래스 수 < 1천만개 |
| 의사결정트리 | 1,000개 | 제한 없음 | 특징 수 x 클래스 수 < 10,000개 |
| 랜덤 포레스트 | 10,000개 | 제한 없음 | 특징 수 x 클래스 수 < 100,000개 |
| 그래디언트 부스티드 트리 | 1,000개 | 제한 없음 | 특징 수 x 클래스 수 < 10,000개 |

**26.4 로지스틱 회귀**- 로지스틱 회귀는 가장 널리 사용되는 분류 기법 중 하나로, 특징과 가중치를 결합하여 특정 레이블에 속할 확률을 얻는 선형 방법론(학습으로 얻어진 가중치를 통해 특징별 중요도 판단 또한 가능)  
- 모델 하이퍼파라미터(모델 기본 구조 결정)  
 - family : 이항/다항 분류 지정  
 - elasticNetParam : L1 과 L2 조합 지정(0~1사이)  
 - fitIntercept : 절편 적합 여부(True or False)  
 - standardization : 입력 데이터 표준화 여부(True or False)  
- 학습 파라미터(학습 수행 시 관여)  
 - maxIter : 총 학습 반복 횟수(default = 100)  
 - weightCol : 특정 row에 가중치를 부여하는 데 사용되는 가중치 컬럼  
- 예측 파라미터(예측 수행 시 관여)  
 - threshold : 주어진 클래스를 예측하는 확률 임계값  
 - thresholds : 다중 분류시, 각 클래스에 대한 임계값 리스트  
- 실습 예제

**26.5 의사결정트리**- 트리 구조를 통해 분류하는 모델  
- 모델 하이퍼파라미터(모델 기본 구조 결정)  
 - maxDepth : 트리 최대 깊이 지정(default=5)  
 - maxBins : 연속형 변수를 구간화할 경우, 구간의 수(default=32)  
 - impurity : 트리 분할 기준으로, entropy와 gini 중 설정  
 - minInfoGain : 트리 분할시 사용할 최소 information gain 기준(default=0)   
- 학습 파라미터(학습 수행 시 관여)  
 - checkpointInterval : 학습 과정 동안 모델 작업 내역을 저장할 반복 수  
- 예측 파라미터(예측 수행 시 관여)  
 - threshold : 주어진 클래스를 예측하는 확률 임계값  
- 실습 예제

**26.6 랜덤 포레스트**- bagging 기법으로 여러 의사결정트리를 독립으로 학습한 다음 투표 혹은 평균으로 최종 예측  
- 모델 하이퍼파라미터(모델 기본 구조 결정, 대부분 의사결정트리와 동일)  
 - numTrees : 학습 시킬 트리 수  
 - featureSubsetStartegy : 각 트리 분할시, 고려할 피쳐 수(auto, all, sqrt 등)  
- 학습 파라미터(학습 수행 시 관여, 의사결정트리와 동일)  
- 예측 파라미터(예측 수행 시 관여, 의사결정트리와 동일)  
- 실습 예제

**26.6 그래디언트 부스티드 트리**- boosting 기법으로 여러 약분류기(트리)에 가중치를 부여하여 강분류기로 학습  
- 모델 하이퍼파라미터(모델 기본 구조 결정, 대부분 의사결정트리와 동일)  
 - lossType : loss함수 지정(현재 logistic loss만 지원)  
 - maxIter : 총 학습 반복 수(default=100)  
 - stepSize : 학습률  
- 학습 파라미터(학습 수행 시 관여, 의사결정트리와 동일)  
- 예측 파라미터(예측 수행 시 관여, 의사결정트리와 동일)  
- 실습 예제

**26.7 나이브 베이즈**- 베이즈 정리에 기반한 분류 모델로, 각 피쳐가 독립이라고 가정  
- 모델 하이퍼파라미터(모델 기본 구조 결정)  
 - modelType : bernoulli와 multinomial 두가지 타입 모델 설정 가능  
 - weightCol : 특정 row에 가중치를 부여하는 데 사용되는 가중치 컬럼  
- 학습 파라미터(학습 수행 시 관여)  
 - smoothing : 확률이 0으로 나오는 것을 방지하기 위해, 계산시 더하는 값(default=1)  
- 예측 파라미터(예측 수행 시 관여)  
 - threshold : 주어진 클래스를 예측하는 확률 임계값  
- 실습 예제

**회귀**- 회귀는 주어진 데이터의 특징을 활용하여 실수(연속형 값)을 예측하는 방법

**27.1 활용 사례**- 영화 관객 수 예측  
- 회사 수익 예측  
- 농작물 수확량 예측

**27.2 MLlib의 회귀 모델**  
- Spark에서 제공하는 대표적인 회귀 모델은 아래와 같음  
 - 선형 회귀  
 - 일반화 선형 회귀  
 - 등위 회귀  
 - 의사결정트리  
 - 랜덤 포레스트  
 - 그래디언트 부스티드 트리  
 - 생존 회귀  
  
- 모델 확장성  
 - 일반적으로 스파크는 대용량의 머신러닝 모델을 학습시키는데 많은 지원을 함

|  |  |  |
| --- | --- | --- |
| 모델 | 최대 특징 수 | 최대 학습 데이터 수 |
| 선형 회귀 | 1 ~ 1천만개 | 제한 없음 |
| 일반화 선형 회귀 | 4,096개 | 제한 없음 |
| 등위 회귀 | N/A | 수백만 건 |
| 의사결정트리 | 1,000개 | 제한 없음 |
| 랜덤 포레스트 | 10,000개 | 제한 없음 |
| 그래디언트 부스티드 트리 | 1,000개 | 제한 없음 |
| 생존 회귀 | 1 ~ 천만개 | 제한 없음 |

**27.3 선형 회귀**- 선형 회귀는 입력 피쳐들의 선형 조합(가중치가 곱해진)으로 학습된 모델로, 로지스틱 회귀처럼 ElasticNet 정규화를 통해 L1, L2 정규화를 혼합할 수 있음  
- 모델 하이퍼파라미터(모델 기본 구조 결정, 로지스틱 회귀와 동일)  
- 학습 파라미터(학습 수행 시 관여, 로지스틱 회귀와 동일)   
- 실습 예제

**27.4 일반화 선형 회귀**- 일반화 선형 회귀는 가우스(선형 회귀), 이항(로지스틱 회귀), 포아송(포아송 회귀), 감마(감마 회귀) 등 다양한 분포 집합으로부터 예상되는 오차 분포를 선택할 수 있음  
- 또한 선형 함수와 분포 함수 평균 사이 관계를 정하는 링크 함수 설정 가능  
- 모델 하이퍼파라미터(모델 기본 구조 결정)  
 - family : 모델에서 사용할 오차 분포(gaussian, binomial 등)  
 - link : 선형 함수와 분포 함수 평균 사이 관계 지정(probit, logit, identity 등)  
 - solver : 최적화 알고리즘(iteratively reweighted least squares만 제공)  
- 학습 파라미터(학습 수행 시 관여, 로지스틱 회귀와 동일)   
- 예측 파라미터(예측 수행 시 관여)  
 - linkPredictionCol : 예측에 적용할 각 링크 함수를 정의하는 컬럼   
- 실습 예제

**27.5 의사결정트리**- 분류에 적용된 의사결정트리와 거의 유사하나, 수치를 예측함  
- 모델 하이퍼파라미터(모델 기본 구조 결정, 의사결정트리와 거의 동일)  
 - impurity : gini, entropy 대신 variance 사용  
- 학습 파라미터(학습 수행 시 관여, 의사결정트리와 동일)  
- 실습 예제

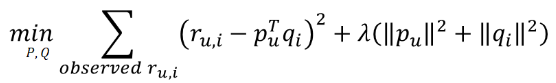
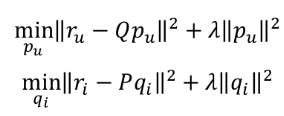
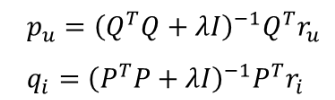
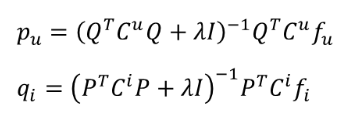
**27.6 랜덤 포레스트**- 분류에 적용된 랜덤 포레스트와 거의 유사하나, 수치를 예측함  
- 모델 하이퍼파라미터(모델 기본 구조 결정, 랜덤 포레스트와 거의 동일)  
 - impurity : gini, entropy 대신 variance 사용  
- 학습 파라미터(학습 수행 시 관여, 랜덤 포레스트와 동일)  
- 실습 예제

**27.6 그래디언트 부스티드 트리**- 분류에 적용된 그래디언트 부스티드 트리와 거의 유사하나, 수치를 예측함  
- 모델 하이퍼파라미터(모델 기본 구조 결정, 그래디언트 부스티드와 거의 동일)  
 - impurity : gini, entropy 대신 variance 사용  
- 학습 파라미터(학습 수행 시 관여, 그래디언트 부스티드와 동일)  
- 실습 예제

**27.8 평가기와 모델 튜닝 자동화**- 회귀를 위한 평가기는 RegressionEvaluator 이며, 이를 사용하기 위해 파이프라인을 구축하고, 테스트하고자 하는 파라미터를 지정하면 가장 적합한 모델이 자동 선택됨  
- 실습 예제

**추천**- 추천은 매우 직관적인 작업으로, 사람들의 평점을 통한 명시적 선호 또는 관찰된 행동을 통한 암시적 선호도를 연구함으로써 특정 사용자와 다른 사용자 사이 유사성이나 제품간 유사성을 도출하여 사용자에게 추천할 수 있음  
- 명시적(explicit) 선호도 : 사용자가 제품에 대해 평가한 평점  
- 암시적(implicit) 선호도 : 평점과 같이 직접적으로 주어지지 않고 특정 행위로 은연중 드러나는 정보

**28.1 활용 사례**- 영화 추천  
- 과목 추천  
- 관심 종목 추천(사내 진행)

**28.2 교차최소제곱 알고리즘을 사용한 협업 필터링 구현**- 스파크에서는 교차최소제곱(Alternating Least Square)을 주력 추천 알고리즘으로 제공  
- 목적함수  
  
 - r은 u유저의 i아이템에 대한 평점  
 - p는 유저 행렬  
 - q는 아이템 행렬  
- 최적화 방법  
 - p를 변수라고 생각하면, Q는 현재 우리가 알고 있는 데이터 X처럼 고정, 따라서 기존의 목적함수도 p와 q의 목적함수 두 개로 분리  
   
 - 이후, OLS 최적화 방법으로 해를 구하면 아래와 같은 수식이 나오며, P와 Q를 각각 고정한 다음 번갈아 업데이트  
   
- Implicity 반영  
 - C는  사용자가 아이템을 얼마나 선호하는지를 나타낸 Confidence 값(r을 이용해 계산)이며, f는 r이 양수인 경우1, 아닌 경우 0으로 반환하는 함수  
   
- 사용자와 아이템간 평점으로 구성된, 3개의 컬럼(사용자 아이디, 아이템 목록, 평점)을 가진 데이터셋만 있으면 됨  
- 데이터가 주어지면 ALS는 사용자가 아직 평가하지 않은 아이템의 평점을 예측하는데 사용할 수 있는 특징 벡터를 생성  
- 데이터가 많을수록 정확하며 신규사용자/신규 제품과 같은 콜드스타트에 약함  
- 모델 하이퍼파라미터(모델 기본 구조 결정)  
 - rank : 학습될 사용자와 아이템의 특징 벡터 차원 수(default=10)   
 - regParam : 과적합 방지를 위한 정규화 파라미터(default=0.1)   
 - implicitiPrefs : 데이터가 명시적인지 암시적인지 지정(default=”explicit”)  
 - nonnegative : 알고리즘이 연산하는 최소제곱 문제에 비음수 제약 조건을 설정하여 음수가 아닌 특징 벡터만 반환하도록 구성(default=False)  
- 학습 파라미터(학습 수행 시 관여)  
 - ALS에 대한 학습 파라미터는 클러스터에 데이터가 분산되는 방식을 보다 저수준까지 제어함  
 - 클러스터 상에 분산된 데이터 그룹을 **블록**이라고 하며 각 블록에 배치할 데이터양을 결정하는 것은 학습 시간에 큰 영향을 미칠 수 있음(성능과는 무관)  
 - numUserBlocks : 사용자를 분할할 블록 수(default=10)  
 - numItemBlocks : 아이템을 분할할 블록 수(default=10)  
 - maxIter : 총 학습 반복 횟수(default = 10)  
 - checkpointInterval : 학습 과정 동안 모델 작업 내역을 저장할 반복 수  
- 예측 파라미터(예측 수행 시 관여)  
 - ALS는 콜드 스타트 전략 예측 파라미터 하나만 제공함  
 - 즉, 학습데이터셋에는 없었던 사용자 또는 아이템에 대해 무엇을 예측해야 하는지 결정  
 - 기본적으로 스파크는 최종 운영되는 모델에 없는 사용자나 아이템이 발생했을 때 NaN을 예측값으로 반환함  
 - coldStartStrategy : “drop”으로 설정시, NaN 값이 포함된 예측 DataFrame의 row를 제거하도록 설정할 수 있음(default=”NaN”)  
- 실습 예제