ICT707

DATA SCIENCE

PRACTICE TASK 3

Semester 1, 2020

Table of Contents

Advancement of Data Science

What Is Data Science?

Data science provides meaningful information based on large amounts of complex data or big Data. Data science, or data-driven science, combines different fields of work in statistics and computation to interpret data for decision-making purposes.

Data science uses techniques such as machine learning and artificial intelligence to extract meaningful information and to predict future patterns and behaviours. The field of data science is growing as technology advances and big data collection and analysis techniques become more sophisticated

Understanding Data Science

Data is drawn from different sectors, channels, and platforms including cell phones, social media, e-commerce sites, healthcare surveys, and Internet searches. The increase in the amount of data available opened the door to a new field of study based on big data—the massive data sets that contribute to the creation of better operational tools in all sectors.

The continually increasing access to data is possible due to advancements in technology and collection techniques. Individuals buying patterns and behaviour can be monitored and predictions made based on the information gathered.

However, the ever-increasing data is unstructured and requires parsing for effective decision making. This process is complex and time-consuming for companies—hence, the emergence of data science.

Data science, or data-driven science, uses big data and machine learning to interpret data for decision-making purposes.

A Brief History of Data Science

The term data science has existed for the better part of the last 30 years and was originally used as a substitute for "computer science" in 1960. Approximately 15 years later, the term was used to define the survey of data processing methods used in different applications. In 2001, data science was introduced as an independent discipline. The Harvard Business Review published an article in 2012 describing the role of the data scientist as the “sexiest job of the 21st century.”

KEY TAKEAWAYS

- Advances in technology, the Internet, social media, and the use of technology have all increased access to big data.
- Data science uses techniques such as machine learning and artificial intelligence to extract meaningful information and to predict future patterns and behaviours.
- The field of data science is growing as technology advances and big data collection and analysis techniques become more sophisticated.

Data Science Today

Companies are applying big data and data science to everyday activities to bring value to consumers. Banking institutions are capitalizing on big data to enhance their fraud detection successes. Asset management firms are using big data to predict the likelihood of a security's price moving up or down at a stated time.

Companies such as Netflix mine big data to determine what products to deliver to its users. Netflix also uses algorithms to create personalized recommendations for users based on their viewing history. Data science is evolving at a rapid rate, and its applications will continue to change lives into the future.

Spark in Data Science

"Apache Spark is a fast and general-purpose Cluster computing system. It is an *open-source cluster computing framework* for *real-time processing.* It has a thriving open-source community and is the most active Apache project at the moment. Spark provides an interface for programming entire clusters with implicit data parallelism and fault-tolerance. It provides high-level APIs in Java, Scala, Python and R, and an optimized engine that supports general execution graphs. It also supports a rich set of higher-level tools including Spark SQL for SQL and structured data processing, MLlib for machine learning, GraphX for graph processing, and Spark Streaming.

Spark refers to the Apache Spark distributed computing framework, originally accessible using the Scala programming language. PySpark is the interface that gives access to Spark using the Python programming language. Another alternative to PySpark would be Spark R, which understands the R language.

Apache Spark is best for huge data, AWS Athena or Google BigQuery can be good competitors for Spark, but Spark has more enriched features. In such case, Spark steals over other competitors. For Data Visualization and creating Dashboards that provide monitoring and insights based on data streams. Here Spark does not come up to that level for this use case.BI tools like Tableau and SiSense provide much better support than Spark for streaming data within a certain range of the data set which is being used.

Machine Learning Implementation

Dataset

Data source:

1. 'rating.csv': https://www.kaggle.com/rounakbanik/the-movies-dataset?select=ratings_small.csv

2. 'movies.csv': https://www.kaggle.com/rounakbanik/the-movies-dataset?select=movies_metadata.csv

We have taken the datasets from Kaggle movie data set. The data set primarily consists of two datasets, where rating.csv contains 4 columns and 45466 rows and movies.csv contains 8 columns and 45466. These data sets are collected from different periods of time and is dependent on the size of the set. The link to the repository is given above. We are here using the Kaggle datasets which is recommended for practice purpose. We are going to utilizing the 12M datasets which provides different as aspects of movie to recommend a user.

Collaborative filtering

With Collaborative filtering we build the movie recommendation model using ALS on the training data, we make predictions (filtering) about the interests of a user by collecting preferences or taste information from many users (collaborating).

Spark MLlib library for Machine Learning provides a Collaborative Filtering implementation by using Alternating Least Squares. The implementation in MLlib has these parameters: (*numBlocks, rank, iterations, lambda, implicitPrefs, alpha*)

```
from pyspark.ml.evaluation import RegressionEvaluator
from pyspark.ml.recommendation import ALS

als = ALS(maxIter=5,regParam=0.09,rank=25,userCol="userId",itemCol="movieId",ratingCol="rating",coldStartStrategy="drop",nonnegative=True)
model = als.fit(training) # fit the ALS model to the training set
```

After successful execution of spark jobs, it's time to evaluate the build model using inbuilt transform function. This function is more or less similar to predict() function in the traditional machine learning algorithm(Sklearn). However, transform () function transform the input test data or unseen data in order to generate predictions.

```
evaluator=RegressionEvaluator(metricName="rmse",labelCol="rating",predictionCol="prediction")
predictions=model.transform(test)
rmse=evaluator.evaluate(predictions)
print("RMSE="+str(rmse)
```

```
RMSE=0.8780123904093605
```

Evaluating a model is a core part of building an effective machine learning model. In PySpark we will be using RMSE (Root mean squared Error) as our evaluation metric.

The RMSE described our error in terms of the rating column.

Logistic regression

Logistic regression is a popular method to predict a categorical response. It is a special case of Generalized Linear models that predicts the probability of the outcomes. In spark.ml logistic regression can be used to predict a binary outcome by using binomial logistic regression, or it can be used to predict a multiclass outcome by using multinomial logistic regression. Use the family parameter to select between these (featuresCol, labelCol, maxIter) or leave it unset and Spark will infer the correct variant.

```
from pyspark.ml.classification import LogisticRegression
lr = LogisticRegression(featuresCol = 'features', labelCol = 'userId',
maxIter=10)
lrModel = lr.fit(training)
```

When the data was fitted perfectly. We will see the model's performance with the test data. We have to practice caution when the models show extraordinary performance with the training data as this can be due to overfitting problem which makes the model not to generalize to unseen data.

```
predict_train=lrModel.transform(training)
predict_test=lrModel.transform(test)
```

We can get accuracy, precision and recall using BinaryClassificationEvaluator which can be used for binary classification as well. Predict using the test data and evaluate the predictions, where the predictions dataframe contains the original data and the predictions.

```
from pyspark.ml.evaluation import BinaryClassificationEvaluator
evaluator=BinaryClassificationEvaluator(rawPredictionCol='rawPrediction
',labelCol='userId')
predict_test.select("userId","rawPrediction","prediction","probability"
).show(5)
print("The area under ROC for train set is {}".format(evaluator.evaluat
e(predict_train)))
print("The area under ROC for test set is {}".format(evaluator.evaluate
(predict_test)))
```

```
The area under ROC for train set is 1.0
The area under ROC for test set is 1.0
```

Evaluating a model is a core part of building an effective machine learning model. In PySpark we will be using RMSE (Root mean squared Error) as our evaluation metric.

References

- https://spark.apache.org/docs/latest/ml-collaborative-filtering.html
- https://spark.apache.org/docs/latest/ml-guide.html
- https://tryolabs.com/blog/introduction-to-recommender-systems/
- https://towardsdatascience.com/learning-how-recommendation-system-recommends-45ad8a941a5a
- https://spark.apache.org/docs/latest/api/python/pyspark.ml.html
- https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-recommendation-engine-python/