Analysis of CicloPi usage

Big Data Analytics

Data Science and Business informatics

Università di Pisa

Alessandro Macagno - 545974

Abstract

Bike sharing has become in the last years a major service available in almost every big city in Italy. The number of people using this service is constantly increasing, as noted by the entrance in the market of private players, like oBike and ReadyBike.
The biggest problem in Italy nowadays for bike sharing services is vandalism: people tend to use bikes and not to respect them. This causes a huge number of unavailable bikes and a lack of service, and so customer dissatisfaction.
Data Mining can't help to teach users to behave as if bikes were their own, but it can help to identify peaks of need and usage as well as moments suitable for maintenance.
My goal is to develop a ML model that provides useful information to identify peaks of usage, using Pyspark 2.3.1. The followed methodology is CRISP-DM.

Business understanding

I was given the Pisa_Operazioni_2015.csv dataset containing all the 2015 records of usage of CicloPi bike rental service.
My goal is to build a model that provides a forecast of how many bikes should at least be up and running in the circuit per day. This information is very useful for the company managing the service: in a low volume day they can set up maintenance and have less available phone operators, while in a peak day there should be many more available bikes and operators.
This model can be joint with the minimal bike movement model, explained during BDA course. CicloPi company could use the joined models to save lot of money.

The first phase in business understanding is to try to figure out what are the variables that can affect bike usage: weather, holiday, season, etc. I have looked for datasets about Pisa in 2015:

- from wordweatheronline.com I found all the information about weather
- feiertagskalender.ch provides holidays informations

The results are amazing: 86,3% of accuracy for a decision tree model with a 3-class classifier (low, medium, high usage). This was achieved through many attempts of different classifiers, from bayes to regression models and random forest classifiers.

The proposed model could be profitably adopted by CicloPi company. The proposed model can be also used by other companies providing their own dataset of usage, but manually downloading and elaborating weather datasets because of wordweather API plan limitations.

Improvements and next developments include a deeper analysis of weather (like temperatures, humidity, etc.). By interviewing cicloPi users more variables could come out, answering the question "When do you use CicloPi? When do you not? Why?". This could lead to some service improvements.

Data Understanding

The initial dataset Pisa_Operazioni_2015.csv has many columns which do not help my model: in fact all I need is how many bikes were used during a certain day. The process could be as simple as grouping by day the dataset counting how many bikes were used, with a resulting 365-rows dataset.
An easy improvement is to segment the day in 3 8-hour-long ranges, to perfectly answer the described business question in case of daily work time (8 hour in Italy and most of Europe). Unfortunately, worldweatheronline provides data divided day by day or 3, 6 and 12 hour, so it's not possible to simulate the 8-hour daily work routine.
My decision has been to divide the day in 4 ranges, 6 hour each: night (from 00 to 6 am, morning from 6 am to 12 am, afternoon from midday to 6 pm and evening from 6pm to midnight).
I could not find a downloadable dataset for holidays in Pisa during 2015, so I put them manually using Excel. Not a very scalable solution, but for the big majority of days the answer is False.

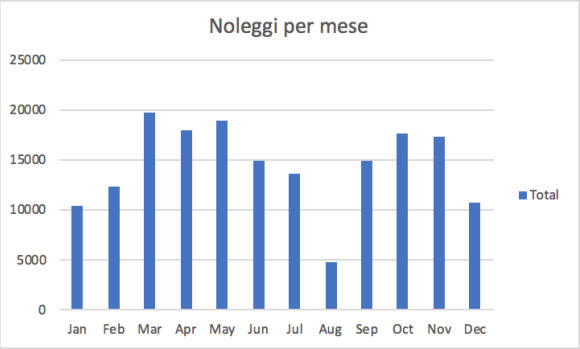
Data Preparation

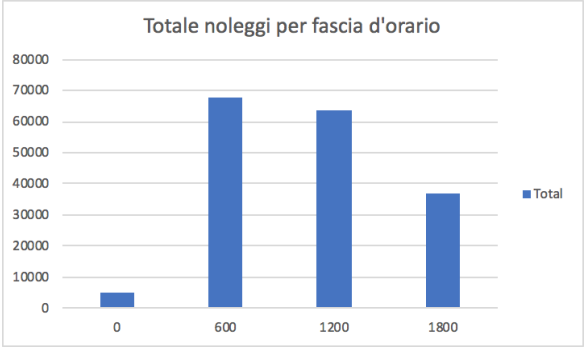
Pisa_Operazioni_2015.csv has 9 columns: Comune (always Pisa), StazPrelievo (where the bike was taken), DataOraPrelievo (when the bike was taken), ColonninaPrelievo (which column the bike was taken from), StazDeposito (where the bike was left), ColonninaDeposito (which column the bike was left at), DataOraDeposito (when the bike was left), IDBadge (unique ID of user's badge), NumeroBadge (another ID of the badge).
None of the provided column were useful apart from DataOraPrelievo. I added a column day_range (which meets the format of worldweatheronline) where I put 00, 600, 1200 or 1800 according to the hour the bike was taken. I chose to use DataOraPrelievo and not DataOraDeposito because sounds the most appropriate. This was achieved via a UDF function.
I also added a separate column for day and month. Then I grouped by day and day range, with a count of records. The count column indicates how many bikes were taken during a day and a day range. I renamed the column total_count because of some reserved keyword errors in Pyspark.
Then, I downloaded the JSON answer from wordweatheronline, which is huge, and translated it into a CSV format, grabbing only the category of weather observed in a certain day. In a future release of the model we could add the temperature, the humidity, wind speed and lots of other informations.
The resulting csv has 6 columns: day, day_range, weather (string column of 19 different values), day_of_week (obtained via DAY_OF_WEEK excel function and then with day_of_week pyspark function), celebration (True / False. NB: Sundays are not holidays) and season (using Excel function found online).
The two datasets were joined on day and day_range. I found two day ranges where there was no rent, I put the count at zero. The resulting dataset has 1458 rows.

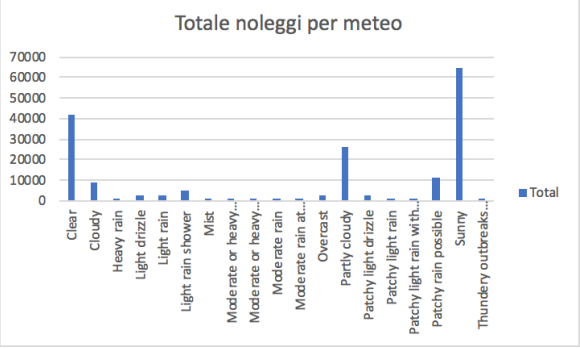
The target variable total_count was discretized in three buckets. The first bucket is from 1 to 34 total counts, the second from 35 to 142 total counts, and the third from 143 to the max

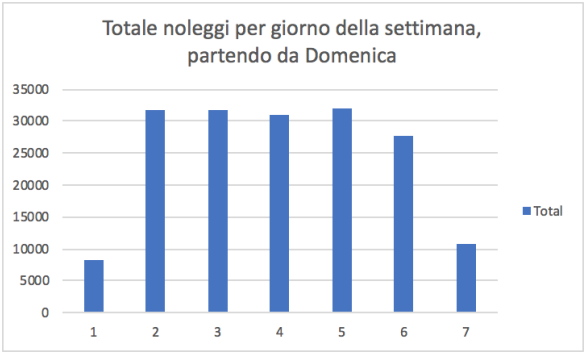
Useful statistics

173 332 total rents, max 445 in a April's Thursday with cloudy weather, min 0 during two nights with rain. Some charts showing the distribution of rentals, obtained via Pivot Table in Excel.









Modeling

For the modeling phase I followed two classic approaches:

- forecast the exact number of rentals. A regression problem
- forecast the binned number of rentals, using as target the column count_discretized. A multi-classification problem

The first approach did not quite work. The linear regression model assumes that the distribution is linear, and it is not true in our problem. The generalized linear regression assumes a Gaussian distribution, and this is again not our case.

The second approach has given better results. I have compared two classification models: Decision Tree and Naive Bayes Classifier.

The decision tree is a greedy algorithm that performs a recursive binary partitioning of the feature space. The tree predicts the same label for each bottommost (leaf) partition. Each partition is chosen greedily by selecting the *best split* from a set of possible splits, in order to maximize the information gain at a tree node

Naive Bayes. This is one of the most known and used classification algorithm. It relies on Bayes probability theorem and often gives very good results. The target variable was the count_discretized.

To train the models I splitted the dataset in 70% of training and 30% of testing. I skipped the validation phase because I did not have enough data.

Spark ML library requires a features column and a label column to train the model and then test it. This is built via VectorAssembler class, which grabs several columns and creates an array in a new column features.
In order to use VectorAssembler class every column has to be a integer, so every categorical features is transformed to int via StringIndexer class and encoded using OneHotEncoderEstimator.

As suggested in many papers found online I created a Pipeline with all the features above-created, and then started training and testing.

Decision Tree

```
dt = DecisionTreeClassifier(featuresCol = 'features', labelCol = 'label')
dtModel = dt.fit(train)
```

Bayes Classifier.

```
nb = NaiveBayes(smoothing=1.0, modelType="multinomial")
model = nb.fit(train)
```

Results

To evaluate classification I used Pyspark.ml.MulticlassClassificationEvaluator. The metric names available are accuracy, f1, weighted Precision and weighted Recall. Quoting Pyspark documentation:
“For multiclass metrics, the notion of positives and negatives is slightly different. Predictions and labels can still be positive or negative, but they must be considered under the context of a particular class. Each label and prediction take on the value of one of the multiple classes and so they are said to be positive for their particular class and negative for all other classes. So, a true positive occurs whenever the prediction and the label match, while a true negative occurs when neither the prediction nor the label take on the value of a given class. By this convention, there can be multiple true negatives for a given data sample. The extension of false negatives and false positives from the former definitions of positive and negative labels is straightforward.”

Algorithm	Accuracy	F1	Precision	Recall
Dec Tree	86,4%	86,18%	87,09%	86,4%
Bayes	69,45%	68,78%	72,28%	69,45%

Conclusions

The Decision Tree Classifier has proven to work better than any other more complicated classifier. Our business goal has been achieved, CicloPi company could adopt the model to forecast the usage of bikes in Pisa in a certain day.

In order to improve the model we could add more features, try larger discretizations of total count, or try different models such as neural networks.

The answer of the business question is yes, we can build a model to forecast bike rentals for a certain day.

Resources

Github Repo: https://github.com/Alagaesia93/CicloPiUsage/tree/master
Datasets:

- https://github.com/Alagaesia93/CicloPiUsage/tree/master/data
- https://www.worldweatheronline.com/
- https://www.feiertagskalender.ch/ (holidays)