ML ENGINEER CASE STUDY

CONSTRUCTION OF A REPRODUCIBLE PIPELINE AND USE FOR TURNOVER PREDICTION

PRELIMINARY QUESTIONS & EDA

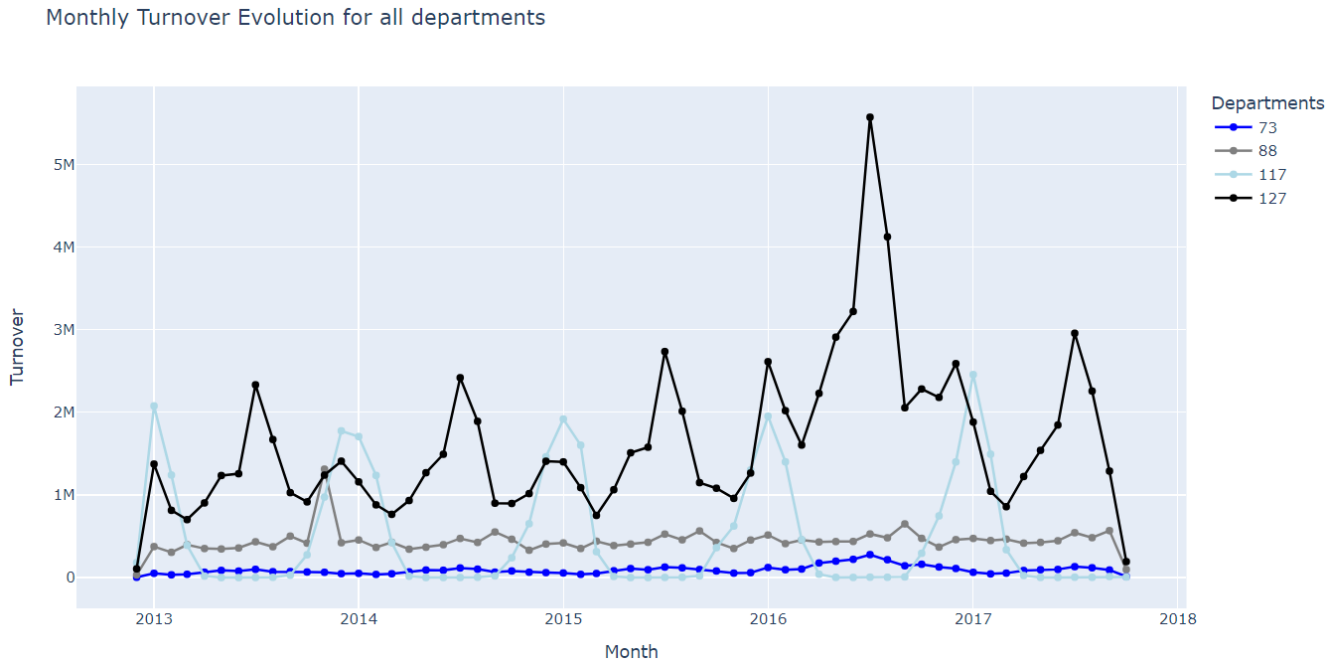
a. Which department made the highest turnover in 2016 ?

Department	Turnover in 2016
127	3.340203e+07
117	6.322402e+06
88	5.654691e+06
73	1.960281e+06

Department 127 made the highest turnover in 2016

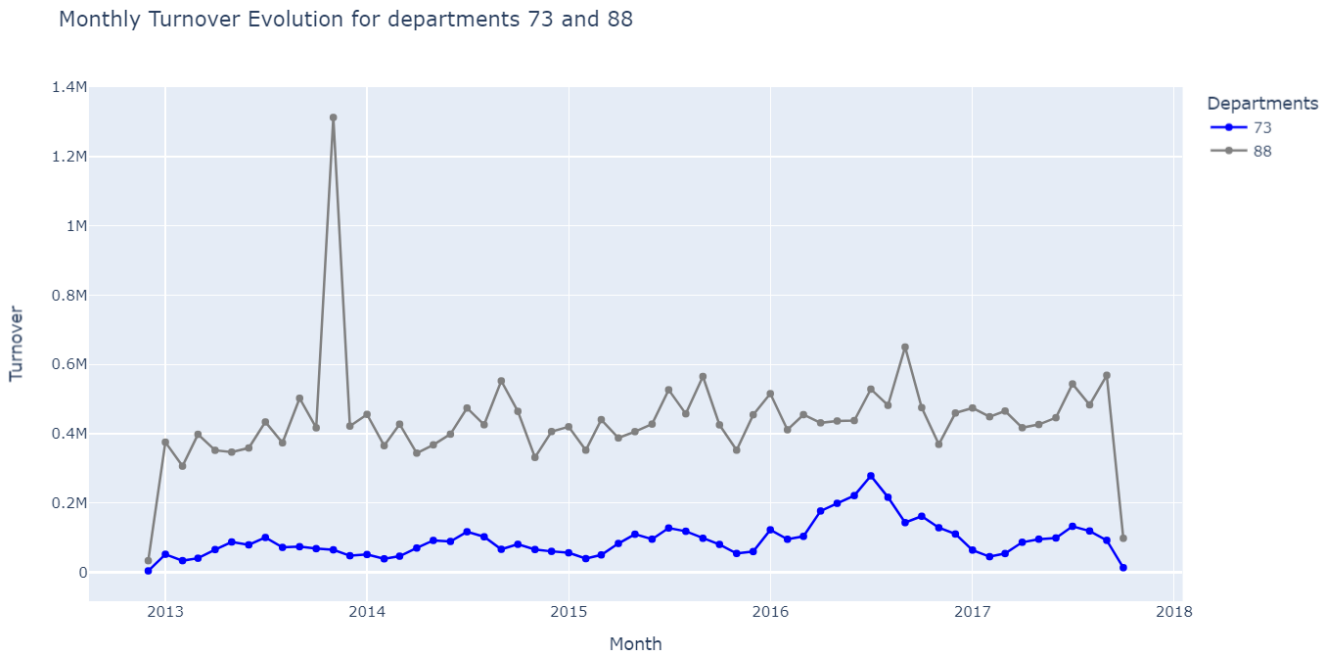
PRELIMINARY QUESTIONS & EDA

b. Based on sales can you guess what kind of sport represents department 73 ?



According to my knowledge of decathlon and some research, the **sports with the highest turnover are water sports, skiing/winter sports, hiking and fitness**.
Department **117** has most of its turnover between October and March, so we can assume that this department represents **winter sports**.
Department **127** has the highest turnover and two sales peaks: July-August and December-January. This may represent **water sports**, with the December/January peak corresponding to people going on winter vacations in the sun.

b. Based on sales can you guess what kind of sport represents department 73 ?



The last two departments, 73 and 88, have lower turnover (88 higher than 73) and are present in stores all year (unlike department 117). Sales in department 88 are very stable all year (except for a peak in November 2013), unlike department 73, which has a seasonal phenomenon in early spring and summer. With this information, we can match department **88** with **fitness** (similar behavior all year) and department **73** with **hiking**, which is more generally practiced in spring and summer, even if it remains present all year.

My guess is that department 73 represent hiking.

DECATHLON

ML PIPELINE

This project is a modular machine learning pipeline designed to handle data processing, model training, and prediction serving via an API. Each task is encapsulated in its own Docker container, allowing for easy management, deployment, and retraining of the model as needed.
The focus was made on code quality, reproducibility and documentation.

The project is organized into three main services:

1. **Data Processor Service:** Extracts and processes raw data from a CSV file.
2. **Model Trainer Service:** Trains a machine learning model using the processed data.
3. **Prediction API Service:** Provides a Flask-based API to serve predictions using the trained model.

Common issues involved in the deployment of machine learning models

1. **Data and Concept Drift:** The input data or the relationship between data and outcomes may change over time, leading to reduced model performance.
2. **Scalability:** Models need to handle larger data volumes and higher traffic in production, which can introduce latency and performance issues.
3. **Model Interpretability:** Complex models can be difficult to understand, making it challenging to explain predictions, especially in regulated industries.
4. **Integration Challenges:** Ensuring smooth integration with existing IT infrastructure can be difficult, particularly with legacy systems.
5. **Monitoring and Maintenance:** Continuous monitoring is essential to detect performance degradation and manage necessary retraining.
6. **Security and Privacy:** Models must be protected against attacks, and data privacy regulations must be adhered to.
7. **Latency:** Real-time applications require models to make quick predictions without sacrificing accuracy.

ML PIPELINE

Solution to monitor the model performance in production

To monitor the model performance in production we can implement the following features :

1. **Automated Metric Tracking:** Continuously monitor key metrics like accuracy, precision, and drift to detect performance issues. Use tools like Evidently AI for drift detection.
2. **Real-Time Dashboard:** Set up a dashboard with tools like Grafana or Kibana to visualize metrics and receive alerts for anomalies.
3. **Logging and Auditing:** Maintain detailed logs of inputs and predictions, and track model versions to ensure traceability.
4. **Periodic Retraining:** Regularly retrain the model with new data and use A/B testing to compare model versions before deployment.
5. **Resource Monitoring:** Use tools like Prometheus to monitor resource usage and latency, ensuring efficient operation under load.

ML PIPELINE

Next steps

To monitor the model performance in production we can implement the following features :

1. **Orchestration with Apache Airflow:** Automate and schedule the pipeline steps (data processing, model training, etc.) using Airflow's DAGs (the development of this feature was planned but did not succed due to deployment problems)
2. **Database Integration with PostgreSQL:** Store processed data, predictions, and model metadata in PostgreSQL for efficient management and retrieval.
3. **Monitoring and Logging:** Implement Prometheus and Grafana for monitoring performance metrics and use logging tools like ELK stack for tracking and debugging.
4. **CI/CD Integration:** Use tools like Jenkins or GitHub Actions for automated testing and deployment to ensure seamless updates to the pipeline.

ML PIPELINE

Pipeline creation using cloud services

This pipeline could have been developed using cloud services to simplify the development (using integrated services), have a better scalability, availability and cost efficiency. To create this pipeline on AWS the following steps could have been taken :

1. **Data Storage:** Use Amazon S3 to store raw and processed data.
2. **Data Processing:** Use AWS Glue or AWS Lambda to process data and store the results back in S3.
3. **Model Training:** Use Amazon SageMaker to train the model. SageMaker can read data from S3 and store the trained model back in S3.
4. **Model Deployment:** Deploy the model using SageMaker Endpoint for real-time predictions.
5. **API Service:** Use Amazon API Gateway with AWS Lambda to create a REST API that invokes the SageMaker endpoint for predictions.
6. **Monitoring:** Use Amazon CloudWatch to monitor model performance and set up alarms for anomalies.