LESSON 7: Responsible AI

Section 1: Lesson Overview ¶

This lesson will introduce you to the essential and difficult discussion about the **potential implications and challenges** posed by Machine Learning. In this lesson, we will explore:

- The modern-day **challenges** posed by **AI** in general and Machine Learning in particular.
- The **core principles** of **responsible AI** (as a broader perspective of Machine Learning).
- How **Microsoft** applies these principles.
- Two essential aspects of Machine Learning models that impact responsible AI: **transparency**, **explainability** and **fairness**.

Section 2: Modern AI - Challenges and Principles

Why worrying about responsible AI?

- **Increasing inequality**: ML models built on datasets with issues can potentially induce or increase the inequalities between the various demographics.
- **Weaponisation**: AI is being weabonised to improve vectors of attack in cybersecurity (e.g. profiling of vulnerable groups to improve fishing attacks delivered by email)
- **Unintentional bias**: e.g., a major online advertising system showing ads for high income jobs to men much more often than to women.
- **Adversarial attacks**: e.g. fooling self-driving cars by applying a specially designed sticker to the sign.
- **Killer drones**: do we really want to defer the decision to kill a human being to a ML model??
- **Deep fakes**: e.g. videos distorting reality, such as world leaders saying something they have never said; this enables the **weaponisation** of **misinformation** and poses a serious threat to the reliability of news and media.
- **Intentional data poisoning and bias**: models are often built with some amount of public data. High damage could be caused if someone intentionally manipulates these public datasets used for ML training.
- **Hype**: there is a lot of hype around ML which drives **unrealistic expectations**.

Approaches to **responsible AI**:

- **Model explainability/interpretability**: ability to interpret and explain the **behaviour** of a **trained model**. Two **aspects**:
    - **Global explanation**: understand the global behaviour of the model (how the model operates in general)
    - **Local explanations**: understand specific predictions made by the model.
- **Model fairness**: investigating whom the model might harm (e.g. who is **neglected?**, who is **mis-represented**?)

Section 3: Microsoft AI Principles

Six Principles Guiding Microsoft Responsible AI Development and Use

Fairness

Reliability and safety

Privacy and security

Inclusiveness

Transparency

Accountability

- **Fairness**: all systems should treat **all people fairly** and not affect similarly situated groups in different ways. Each AI system should be built from a **diverse pool** of **AI talent**, using **representative data** and **analytical techniques** that **detect** and **eliminate bias.** This wil require the **involvement of domain experts** in the **design process** and the **systematic evaluation** of the **data** and **models.**

- **Reliability and safety**: customers need to **trust** that the **AI solution** will **perform reliably and safely** withiin a **clear set of parameters** and **respond safely** to **unanticipated situations**. This requires **extensive testing** of **training data** and **models**, a **robust feedback mechanism** and **processes.** for **documenting/auditing performance**, as well as determining **how/when** the **AI system** should **seek the input of humans.**

- **Privacy and security**: AI systems should be secure and **respect existing privacy laws.** Without such protections, users will not be able to share the data needed to train the AI. AI systems should be **transparent** about **data collection**, use **good controls** and **good de-identification techniques** and have **policies** that **facilitate access to the data** that the AI needs to operate effectively.

- **Inclusiveness**: to **benefit everyone**, AI systems should **engage** and **empower people** and use **inclusive design practices** to **eliminate unintentional barriers.** AI technologies must **understand the context, needs and expectations** of people that use them and **address potential barriers** that could inentionally exclude people. AI can be a powerful tool to **enhance** the **opportunities** for those with **disabilities**.

- **Transparency**: when AI systems help making decisions that impact on people's lives, it is important that people understand **how those decisions were made.** People should know how the **AI system works** and how **interact with the data** to make those decisions. This makes easier to **identify** and **raise awareness** on issues like **potential bias**, **errors**, **unintented outcomes.**

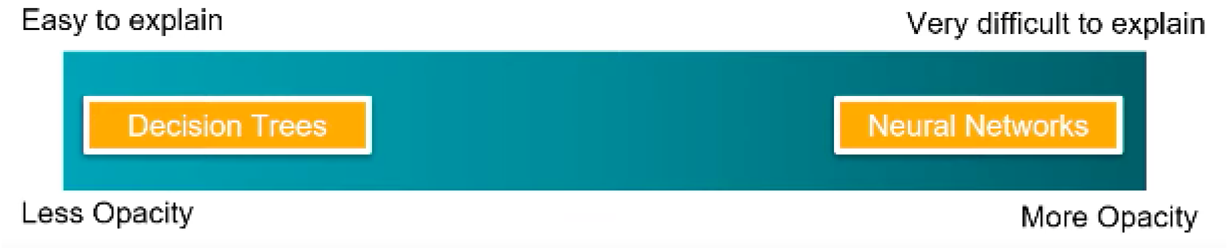
- **Accountability**: those who **desing and deploy** the **AI systems** must be **accountable** for **how their systems operates** and should **periodically check** whether their **accountability norms are being adhered to** and if they are **working effectively**.

Transparency and **accountability** are **foundational principles** to ensure to the effectiveness of all the others.

The remember all the principles use the acronym *PARFIT*.

Section 5: Model Transparency and Explainability

In ML we use a set of features vars to predict the likely value of a target label. A trained model is essentially a function that takes the features' values and computes a result (the predicted value). One of the challenges of this approach is the **opacity** of this **function**, i.e. the degree to which the inner workings of the function can be seen, understood and explained. This opacity depends quite a lot on the **class of algorithms** useed to train the model. The **spectrum** of **model explainability** can be represented like this:



- **Decision tree algorithms** profuce the **clearest trained models**, as they are essentially **self-explanatory** because you can inspect the entire chain of decisions that lead up to any prediction.
- **Deep neural networks** tend to produce the **most opaque models**, as they are essentially a **set of numeric weights** quite difficult to understand and explain even for the experts.

One of the most important aspects of model explainability is **fetaure importance**, i.e. how important is each given input feature in the model and its contribution to the resulting predictions.

In Azure ML we can **explain models** using the **Explainers**. There are two kinds of them, both available using Azure ML SDK:

- **Direct Explainers**: they are integrated in the SDK and **expose** a **common output format** and **API**. You will typically choose a **specific direct explainer**, based on your **model type** and then use it directly into your code to explain the model. Azure includes a half dozen direct explainers. Examples:
  - **SHAP Tree Explainer**: (**model specific**) used to explain trees and ensembles of trees
  - **SHAP Deep Explainer**: (**model specific**) used to explain deep neural nets
  - **Mimic Explainer**: (**model agnostic**) it creates its own model that it is trained to approximate the predictions of the original black-box model. The created approximation is readiliy explainable
  - **SHAP Kernel Explainer**: (**model agnostic**)
- **Meta-Explainers**: they are used for the **automatic selection** of **direct explainers**, based on a given model and a dataset, a meta-explainer would select the best direct explainer and use it to generate the model explanation. There are different types based on the kind of data that the model is making predictions against:
  - **Tabular Explainer**
  - **Text Explainer**
  - **Image Explainer**

The approach is to use Azure ML SDK to create global/local predictions in your notebook code and then use visualisations provided by Azure ML to explore the explanations graphically, either directly in the notebook or within Azure ML Studio

Section 6: Lab (Model Explainability)

Model interpretability with Azure Machine Learning service

Machine learning interpretability is important in two phases of machine learning development cycle:

- **During training**: Model designers and evaluators require interpretability tools to explain the output of a model to stakeholders to build trust. They also need insights into the model so that they can debug the model and make decisions on whether the behavior matches their objectives. Finally, they need to ensure that the model is not biased.

- **During inferencing**: Predictions need to be explainable to the people who use your model. For example, why did the model deny a mortgage loan, or predict that an investment portfolio carries a higher risk? The Azure Machine Learning Interpretability Python SDK incorporates technologies developed by Microsoft and proven third-party libraries (for example, SHAP and LIME). The SDK creates a common API across the integrated libraries and integrates Azure Machine Learning services. Using this SDK, you can explain machine learning models globally on all data, or locally on a specific data point using the state-of-art technologies in an easy-to-use and scalable fashion.

In this lab, we will be using a subset of NYC Taxi & Limousine Commission - green taxi trip records available from Azure Open Datasets. The data is enriched with holiday and weather data. We will use data transformations and the GradientBoostingRegressor algorithm from the scikit-learn library to train a regression model to predict taxi fares in New York City based on input features such as, number of passengers, trip distance, datetime, holiday information and weather information.

The primary goal of this quickstart is to explain the predictions made by our trained model with the various Azure Model Interpretability packages of the Azure Machine Learning Python SDK. https://docs.microsoft.com/en-us/azure/machine-learning/how-to-machine-learning-interpretability (https://docs.microsoft.com/en-us/azure/machine-learning/how-to-machine-learning-interpretability)

Refer to this Jupyter notebook: https://github.com/solliancenet/udacity-intro-to-ml-labs/blob/master/aml-visual-interface/lab-23/notebook/interpretability-with-AML.ipynb (https://github.com/solliancenet/udacity-intro-to-ml-labs/blob/master/aml-visual-interface/lab-23/notebook/interpretability-with-AML.ipynb)

Get the Global Feature Importance Values

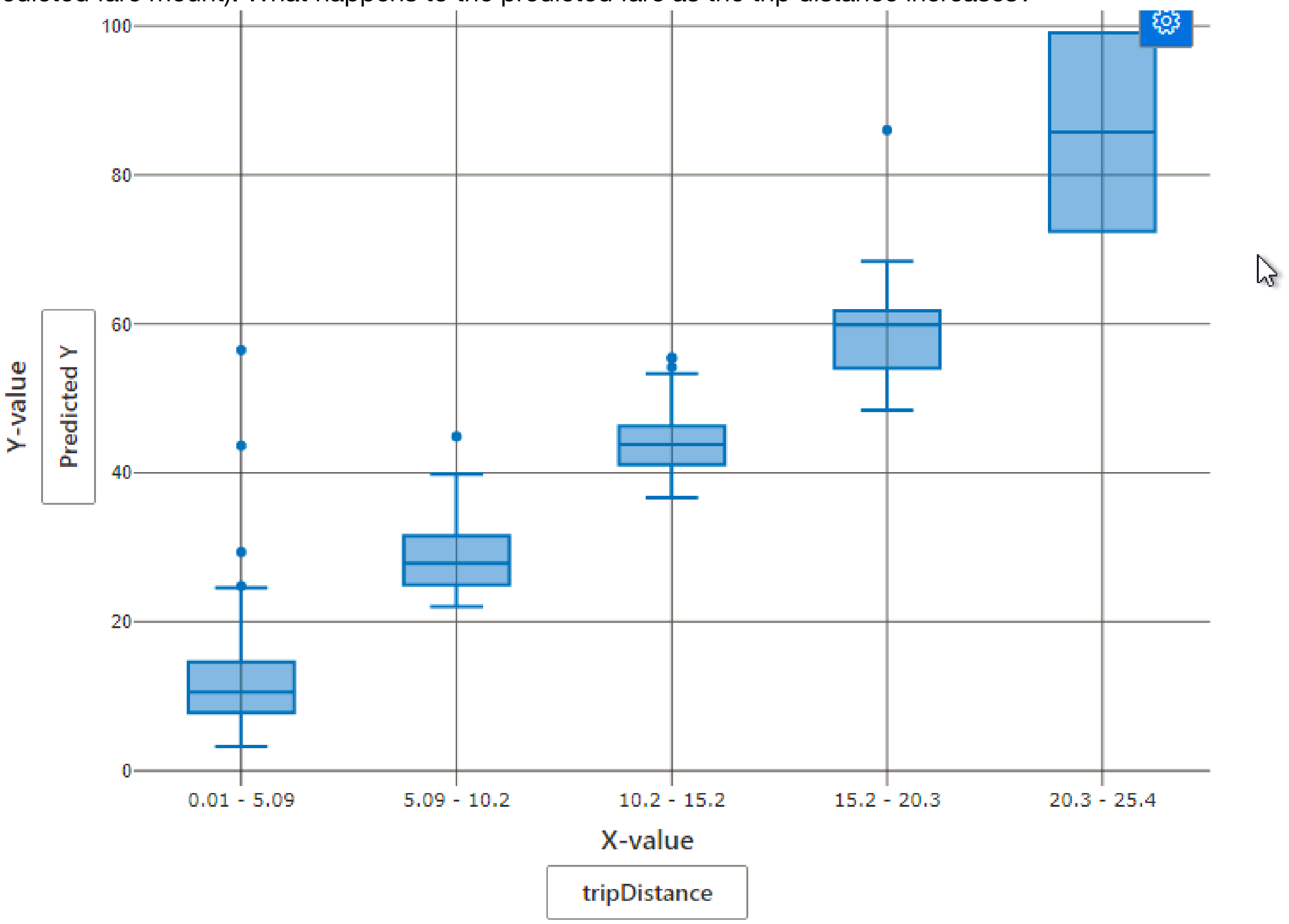
```
In [6]: # You can use the training data or the test data here
        global_explanation = tabular_explainer.explain_global(X_test)

        # Sorted feature importance values and feature names
        sorted_global_importance_values = global_explanation.get_ranked_global_values()
        sorted_global_importance_names = global_explanation.get_ranked_global_names()
        dict(zip(sorted_global_importance_names, sorted_global_importance_values))
```

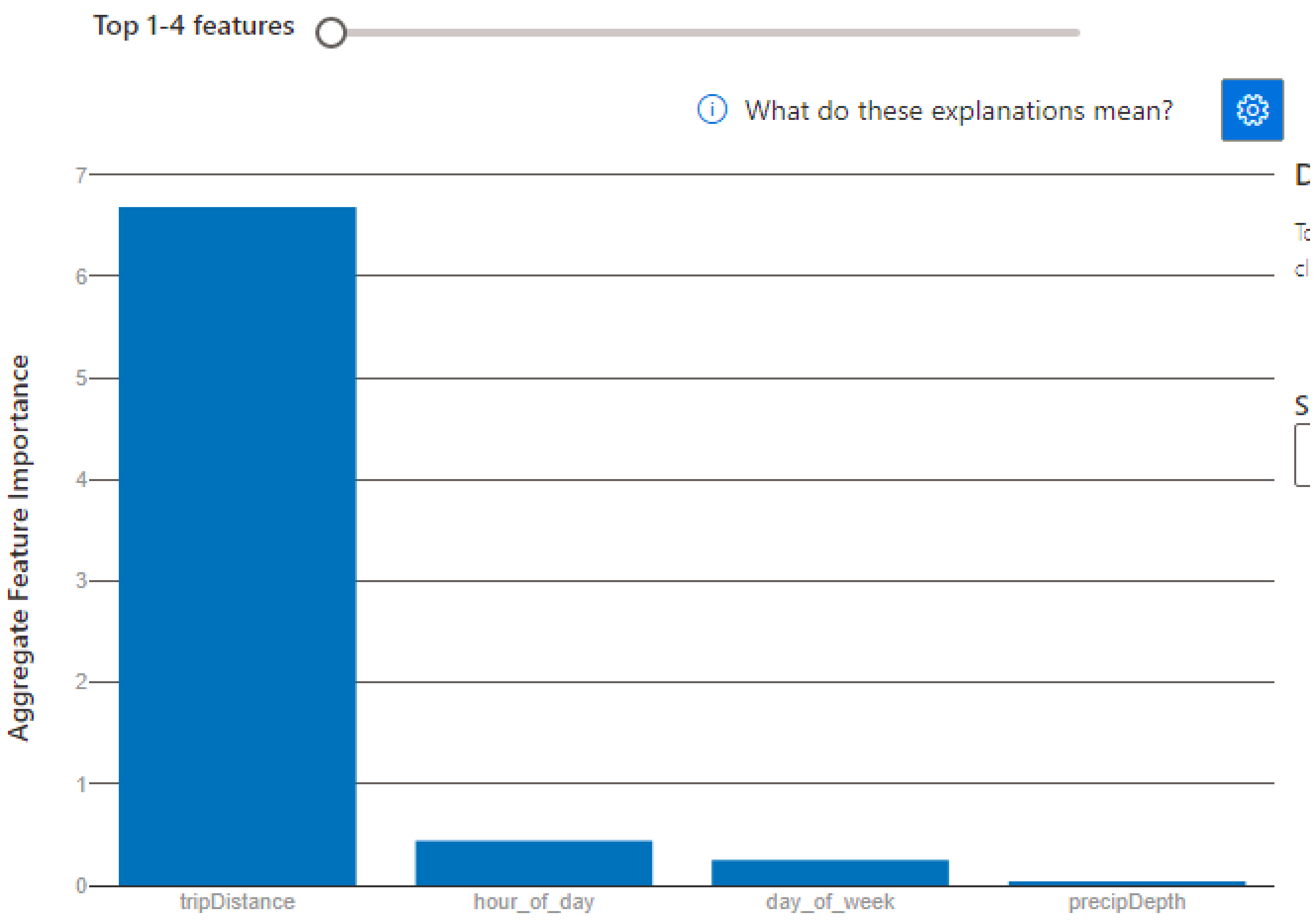
```
Out[6]: {'tripDistance': 6.68236779966518,
         'hour_of_day': 0.43776554746885865,
         'day_of_week': 0.2484834142012842,
         'precipDepth': 0.03667128026414406,
         'passengerCount': 0.03621362066630697,
         'temperature': 0.028673039018950554,
         'day_of_month': 0.02481752134301868,
         'snowDepth': 0.011707454642664863,
         'normalizeHolidayName': 0.008386694244178776,
         'isPaidTimeOff': 0.008167777534731321,
         'month_num': 0.0035810626741262738,
         'vendorID': 0.002070085886874342,
         'precipTime': 0.0014110619134127254}
```

In the Dashboard that is displayed, try answering the following questions:

1) Select the Data Exploration tab, the set the X value to tripDistance and the Y value to PredictedY (this is predicted fare mount). What happens to the predicted fare as the trip distance increases?



2) Select the Global Importance tab. Drag the slider under Top K Features so its value is set to 3. What are the top 3 most important features? Which feature has the highest feature importance (and is therefore the most important feature)?



Local Explanations

Local Explanation

You can use the TabularExplainer for a single prediction. You can focus on a single instance and examine model prediction for this input, and explain why.

We will create two sample inputs to explain the individual predictions.

- **Data 1**
  - 4 Passengers at 3:00PM, Friday July 5th, temperature 80F, travelling 10 miles
- **Data 2**
  - 1 Passenger at 6:00AM, Monday January 20th, rainy, temperature 35F, travelling 5 miles

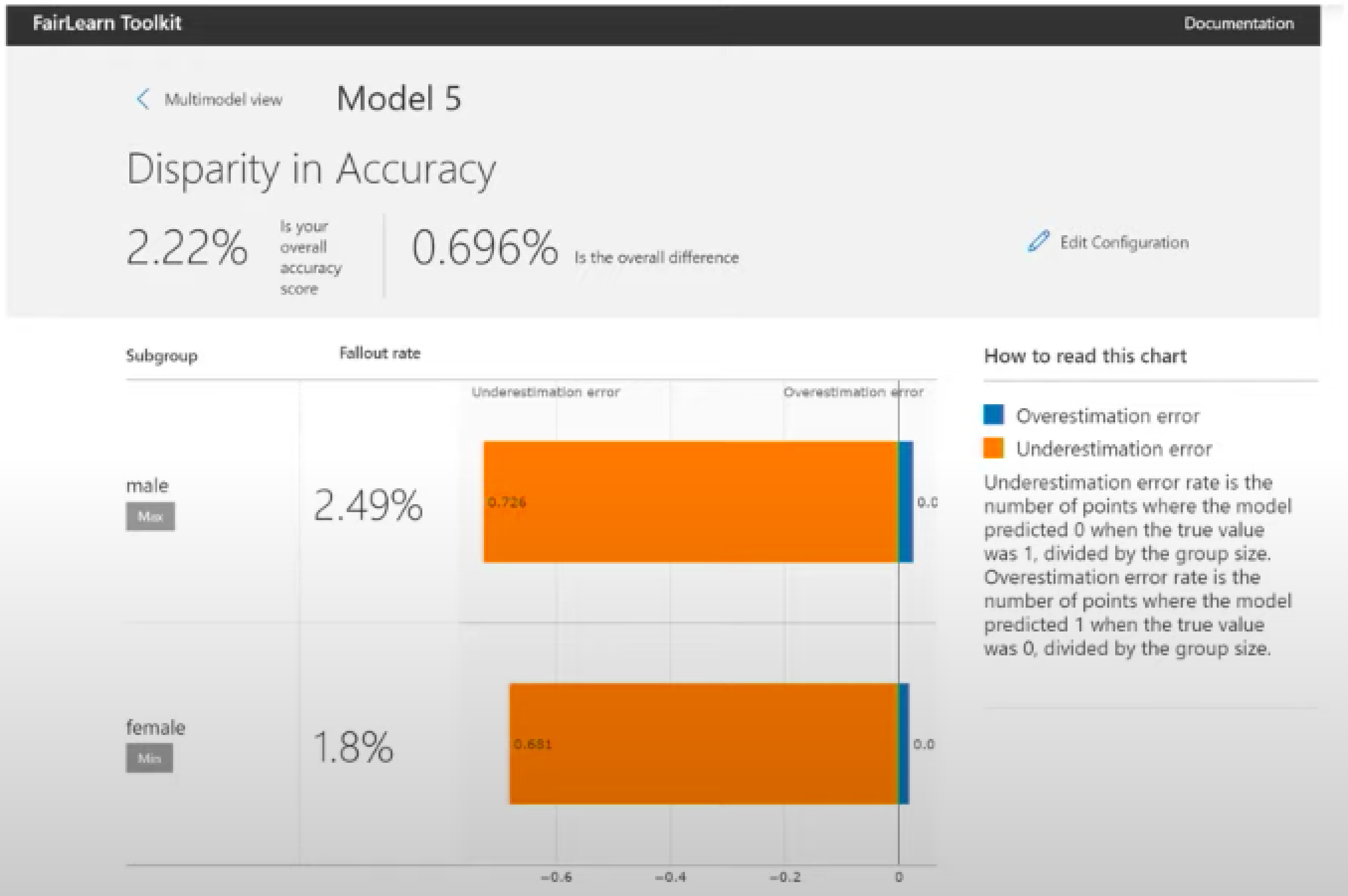
Out[9]:

	1st	2nd	3rd	4th	5th
Data 1	tripDistance	hour_of_day	passengerCount	day_of_week	temperature
	23.7423	0.812614	0.404525	0.130242	0.124966
Data 2	tripDistance	temperature	day_of_week	month_num	precipTime
	7.73348	0.0885307	0.0760823	0.00977716	0.000464409

Section 8: Model Fairness

FairLearn is a **toolkit** to **identify** and **mitigate unfairness** in ML models (binary classification and regression). It enables anyone involved in their development to assess their fairness and mitigate the observed unfairness.

It follows the principle of **group fairness**: which groups of individuals are at risk for experiencing harms? FairLearn uses various techniques to probe the model with your data to identify these groups. The analysis result are made available visually in an easy-to-understand dashboard.
Example screenshot:



Underestimation error rate is the number of points where the model predicted 0 when the true value was 1, divided by the group size. Overestimation error rate is the number of points where the model predicted 1 when the true value was 0, divided by the group size.

You can use some of the **Fairlearn** (https://fairlearn.github.io/) capabilities when applying model fairness to your work. You are certainly welcome to check out the FairLearn repository, including the notebooks we use in the demo, here (https://github.com/fairlearn/fairlearn).