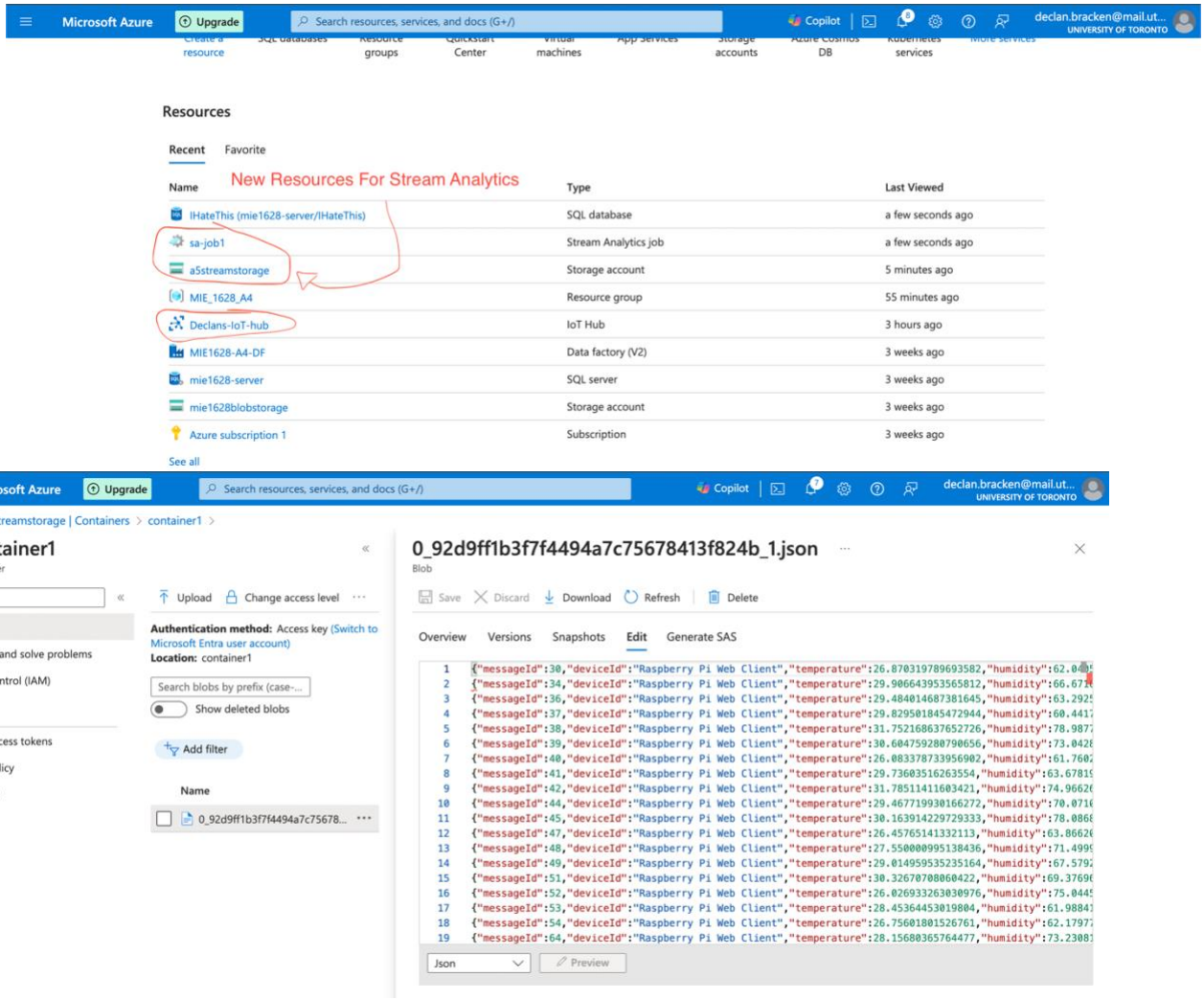
MIE 1628 - Assignment 5

Declan Bracken – 1006251324

Part A)

1. The following steps are performed in order of the image:
    - **Ingest Data**: This is the process of importing, transferring, loading, and processing data from a variety of sources. In Azure, this can be done using Azure Data Factory or Azure Event Hubs. Azure Data Factory can be used for batch data ingestion, while Azure Event Hubs can handle real-time streaming data.
    - **Data Store**: This refers to the storage solution where the ingested data is kept. For a big data scenario, Azure offers several options such as Azure Blob Storage for large amounts of unstructured data, Azure Data Lake for big data analytics, or Azure SQL Data Warehouse for large volumes of relational data.
    - **Prepare and Transform Data**: This stage involves cleaning, transforming, and preparing data for analysis. Azure Data Factory can be used again here for orchestrating and automating the data flow. Additionally, Azure Databricks can be used for more complex data processing and transformation tasks with its Apache Spark-based analytics engine.
    - **Model and Serve Data**: The final stage in the process where data is used to build models for analysis or predictions, and the results are served to users or applications. Azure Machine Learning service is ideal for building, training, and deploying machine learning models. Azure Analysis Services can be used to serve data for business intelligence purposes.
2. Azure Stream Analytics is an event-processing engine that allows you to examine high volumes of data streaming from devices, sensors, websites, social media feeds, applications, and more in real time. Stream analytics starts with a data input, ingestion can take in data from Azure services like IoT hub, event hub, or blob storage. This data is then put through a stream analytics job, which is the heart of Azure stream analytics. This job is programmable in a SQL-like query language where one can specify transformation and processing steps like aggregation, filtering, or mathematical operations. Azure uses a complex event processing engine (CEP) to process input data on the fly for real time analytics. The processed output data is then streamed to another Azure service like a SQL database, Cosmos DB, blob storage, etc...

3.



Resulting JSON:

{"messageId":30,"deviceId":"Raspberry Pi Web Client","temperature":26.870319789693582,"humidity":62.0405189789532,"EventProcessedUtcTime":"2024-04-13T18:05:29.7552156Z","PartitionId":0,"EventEnqueuedUtcTime":"2024-04-13T18:04:27.5300000Z","IoTHub":{"MessageId":null,"CorrelationId":null,"ConnectionDeviceId":"declans-device","ConnectionDeviceGenerationId":"638486169586543168","EnqueuedTime":"2024-04-13T18:04:27.5400000Z"}}

{"messageId":34,"deviceId":"Raspberry Pi Web Client","temperature":29.906643953565812,"humidity":66.67163361522863,"EventProcessedUtcTime":"2024-04-13T18:05:29.9327849Z","PartitionId":0,"EventEnqueuedUtcTime":"2024-04-13T18:04:35.5460000Z","IoTHub":{"MessageId":null,"CorrelationId":null,"ConnectionDeviceId":"declans-device","ConnectionDeviceGenerationId":"638486169586543168","EnqueuedTime":"2024-04-13T18:04:35.5420000Z"}}

{"messageId":36,"deviceId":"Raspberry Pi Web Client","temperature":29.484014687381645,"humidity":63.292505666693096,"EventProcessedUtcTime":"2024-04-13T18:05:29.9328624Z","PartitionId":0,"EventEnqueuedUtcTime":"2024-04-13T18:04:39.5460000Z","IoTHub":{"MessageId":null,"CorrelationId":null,"ConnectionDeviceId":"declans-device","ConnectionDeviceGenerationId":"638486169586543168","EnqueuedTime":"2024-04-13T18:04:39.5420000Z"}}
{"messageId":37,"deviceId":"Raspberry Pi Web Client","temperature":29.829501845472944,"humidity":60.441765775274,"EventProcessedUtcTime":"2024-04-13T18:05:29.9328976Z","PartitionId":0,"EventEnqueuedUtcTime":"2024-04-13T18:04:41.5300000Z","IoTHub":{"MessageId":null,"CorrelationId":null,"ConnectionDeviceId":"declans-device","ConnectionDeviceGenerationId":"638486169586543168","EnqueuedTime":"2024-04-13T18:04:41.5270000Z"}}
{"messageId":38,"deviceId":"Raspberry Pi Web Client","temperature":31.752168637652726,"humidity":78.9877310816304,"EventProcessedUtcTime":"2024-04-13T18:05:29.9329332Z","PartitionId":0,"EventEnqueuedUtcTime":"2024-04-13T18:04:43.5460000Z","IoTHub":{"MessageId":null,"CorrelationId":null,"ConnectionDeviceId":"declans-device","ConnectionDeviceGenerationId":"638486169586543168","EnqueuedTime":"2024-04-13T18:04:43.5420000Z"}}
{"messageId":39,"deviceId":"Raspberry Pi Web Client","temperature":30.604759280790656,"humidity":73.04288713676284,"EventProcessedUtcTime":"2024-04-13T18:05:29.9329688Z","PartitionId":0,"EventEnqueuedUtcTime":"2024-04-13T18:04:45.5460000Z","IoTHub":{"MessageId":null,"CorrelationId":null,"ConnectionDeviceId":"declans-device","ConnectionDeviceGenerationId":"638486169586543168","EnqueuedTime":"2024-04-13T18:04:45.5420000Z"}}
{"messageId":40,"deviceId":"Raspberry Pi Web Client","temperature":26.083378733956902,"humidity":61.76024501970854,"EventProcessedUtcTime":"2024-04-13T18:05:29.9330029Z","PartitionId":0,"EventEnqueuedUtcTime":"2024-04-13T18:04:47.5460000Z","IoTHub":{"MessageId":null,"CorrelationId":null,"ConnectionDeviceId":"declans-device","ConnectionDeviceGenerationId":"638486169586543168","EnqueuedTime":"2024-04-13T18:04:47.5420000Z"}}
{"messageId":41,"deviceId":"Raspberry Pi Web Client","temperature":29.73603516263554,"humidity":63.6781928554887,"EventProcessedUtcTime":"2024-04-13T18:05:29.9330399Z","PartitionId":0,"EventEnqueuedUtcTime":"2024-04-13T18:04:49.5460000Z","IoTHub":{"MessageId":null,"CorrelationId":null,"ConnectionDeviceId":"declans-device","ConnectionDeviceGenerationId":"638486169586543168","EnqueuedTime":"2024-04-13T18:04:49.5280000Z"}}

{"messageId":42,"deviceId":"Raspberry Pi Web Client","temperature":31.78511411603421,"humidity":74.9662677361384,"EventProcessedUtcTime":"2024-04-13T18:05:29.9330735Z","PartitionId":0,"EventEnqueuedUtcTime":"2024-04-13T18:04:51.5460000Z","IoTHub":{"MessageId":null,"CorrelationId":null,"ConnectionDeviceId":"declans-device","ConnectionDeviceGenerationId":"638486169586543168","EnqueuedTime":"2024-04-13T18:04:51.5430000Z"}}
{"messageId":44,"deviceId":"Raspberry Pi Web Client","temperature":29.467719930166272,"humidity":70.07108740416393,"EventProcessedUtcTime":"2024-04-13T18:05:29.9331491Z","PartitionId":0,"EventEnqueuedUtcTime":"2024-04-13T18:04:55.5300000Z","IoTHub":{"MessageId":null,"CorrelationId":null,"ConnectionDeviceId":"declans-device","ConnectionDeviceGenerationId":"638486169586543168","EnqueuedTime":"2024-04-13T18:04:55.5280000Z"}}
{"messageId":45,"deviceId":"Raspberry Pi Web Client","temperature":30.163914229729333,"humidity":78.08688733496851,"EventProcessedUtcTime":"2024-04-13T18:05:29.9331841Z","PartitionId":0,"EventEnqueuedUtcTime":"2024-04-13T18:04:57.4830000Z","IoTHub":{"MessageId":null,"CorrelationId":null,"ConnectionDeviceId":"declans-device","ConnectionDeviceGenerationId":"638486169586543168","EnqueuedTime":"2024-04-13T18:04:57.4810000Z"}}
{"messageId":47,"deviceId":"Raspberry Pi Web Client","temperature":26.45765141332113,"humidity":63.866207837043845,"EventProcessedUtcTime":"2024-04-13T18:05:29.9332554Z","PartitionId":0,"EventEnqueuedUtcTime":"2024-04-13T18:05:01.5480000Z","IoTHub":{"MessageId":null,"CorrelationId":null,"ConnectionDeviceId":"declans-device","ConnectionDeviceGenerationId":"638486169586543168","EnqueuedTime":"2024-04-13T18:05:01.5430000Z"}}
{"messageId":48,"deviceId":"Raspberry Pi Web Client","temperature":27.550000995138436,"humidity":71.499988071787,"EventProcessedUtcTime":"2024-04-13T18:05:29.9332894Z","PartitionId":0,"EventEnqueuedUtcTime":"2024-04-13T18:05:03.5480000Z","IoTHub":{"MessageId":null,"CorrelationId":null,"ConnectionDeviceId":"declans-device","ConnectionDeviceGenerationId":"638486169586543168","EnqueuedTime":"2024-04-13T18:05:03.5440000Z"}}
{"messageId":49,"deviceId":"Raspberry Pi Web Client","temperature":29.014959535235164,"humidity":67.57922778994426,"EventProcessedUtcTime":"2024-04-13T18:05:29.9333221Z","PartitionId":0,"EventEnqueuedUtcTime":"2024-04-13T18:05:05.5320000Z","IoTHub":{"MessageId":null,"CorrelationId":null,"ConnectionDeviceId":"declans-device","ConnectionDeviceGenerationId":"638486169586543168","EnqueuedTime":"2024-04-13T18:05:05.5280000Z"}}

{"messageId":51,"deviceId":"Raspberry Pi Web Client","temperature":30.32670708060422,"humidity":69.37696947221734,"EventProcessedUtcTime":"2024-04-13T18:05:29.9333896Z","PartitionId":0,"EventEnqueuedUtcTime":"2024-04-13T18:05:09.5320000Z","IoTHub":{"MessageId":null,"CorrelationId":null,"ConnectionDeviceId":"declans-device","ConnectionDeviceGenerationId":"638486169586543168","EnqueuedTime":"2024-04-13T18:05:09.5290000Z"}}
{"messageId":52,"deviceId":"Raspberry Pi Web Client","temperature":26.026933263030976,"humidity":75.04459872600493,"EventProcessedUtcTime":"2024-04-13T18:05:29.9334224Z","PartitionId":0,"EventEnqueuedUtcTime":"2024-04-13T18:05:11.5320000Z","IoTHub":{"MessageId":null,"CorrelationId":null,"ConnectionDeviceId":"declans-device","ConnectionDeviceGenerationId":"638486169586543168","EnqueuedTime":"2024-04-13T18:05:11.5290000Z"}}
{"messageId":53,"deviceId":"Raspberry Pi Web Client","temperature":28.45364453019804,"humidity":61.98841811936888,"EventProcessedUtcTime":"2024-04-13T18:05:29.9334549Z","PartitionId":0,"EventEnqueuedUtcTime":"2024-04-13T18:05:13.5630000Z","IoTHub":{"MessageId":null,"CorrelationId":null,"ConnectionDeviceId":"declans-device","ConnectionDeviceGenerationId":"638486169586543168","EnqueuedTime":"2024-04-13T18:05:13.5600000Z"}}
{"messageId":54,"deviceId":"Raspberry Pi Web Client","temperature":26.75601801526761,"humidity":62.179777392786654,"EventProcessedUtcTime":"2024-04-13T18:05:29.9334899Z","PartitionId":0,"EventEnqueuedUtcTime":"2024-04-13T18:05:15.5320000Z","IoTHub":{"MessageId":null,"CorrelationId":null,"ConnectionDeviceId":"declans-device","ConnectionDeviceGenerationId":"638486169586543168","EnqueuedTime":"2024-04-13T18:05:15.5320000Z"}}
{"messageId":64,"deviceId":"Raspberry Pi Web Client","temperature":28.15680365764477,"humidity":73.23081662901828,"EventProcessedUtcTime":"2024-04-13T18:05:35.7768247Z","PartitionId":0,"EventEnqueuedUtcTime":"2024-04-13T18:05:35.5320000Z","IoTHub":{"MessageId":null,"CorrelationId":null,"ConnectionDeviceId":"declans-device","ConnectionDeviceGenerationId":"638486169586543168","EnqueuedTime":"2024-04-13T18:05:35.5320000Z"}}
{"messageId":67,"deviceId":"Raspberry Pi Web Client","temperature":31.628980939795376,"humidity":79.59418985883843,"EventProcessedUtcTime":"2024-04-13T18:05:41.6991175Z","PartitionId":0,"EventEnqueuedUtcTime":"2024-04-13T18:05:41.5470000Z","IoTHub":{"MessageId":null,"CorrelationId":null,"ConnectionDeviceId":"declans-device","ConnectionDeviceGenerationId":"638486169586543168","EnqueuedTime":"2024-04-13T18:05:41.5480000Z"}}

{"messageId":68,"deviceId":"Raspberry Pi Web Client","temperature":27.062397926716056,"humidity":64.49261003948416,"EventProcessedUtcTime":"2024-04-13T18:05:43.7772004Z","PartitionId":0,"EventEnqueuedUtcTime":"2024-04-13T18:05:43.5320000Z","IoTHub":{"MessageId":null,"CorrelationId":null,"ConnectionDeviceId":"declans-device","ConnectionDeviceGenerationId":"638486169586543168","EnqueuedTime":"2024-04-13T18:05:43.5330000Z"}}
{"messageId":74,"deviceId":"Raspberry Pi Web Client","temperature":26.714301181650445,"humidity":77.69563665753999,"EventProcessedUtcTime":"2024-04-13T18:05:55.6066853Z","PartitionId":0,"EventEnqueuedUtcTime":"2024-04-13T18:05:55.5470000Z","IoTHub":{"MessageId":null,"CorrelationId":null,"ConnectionDeviceId":"declans-device","ConnectionDeviceGenerationId":"638486169586543168","EnqueuedTime":"2024-04-13T18:05:55.5510000Z"}}
{"messageId":75,"deviceId":"Raspberry Pi Web Client","temperature":27.000529580056934,"humidity":63.70064753496623,"EventProcessedUtcTime":"2024-04-13T18:05:57.7932502Z","PartitionId":0,"EventEnqueuedUtcTime":"2024-04-13T18:05:57.5470000Z","IoTHub":{"MessageId":null,"CorrelationId":null,"ConnectionDeviceId":"declans-device","ConnectionDeviceGenerationId":"638486169586543168","EnqueuedTime":"2024-04-13T18:05:57.5350000Z"}}
{"messageId":80,"deviceId":"Raspberry Pi Web Client","temperature":28.96267698041912,"humidity":64.86865803459291,"EventProcessedUtcTime":"2024-04-13T18:06:07.7618053Z","PartitionId":0,"EventEnqueuedUtcTime":"2024-04-13T18:06:07.5310000Z","IoTHub":{"MessageId":null,"CorrelationId":null,"ConnectionDeviceId":"declans-device","ConnectionDeviceGenerationId":"638486169586543168","EnqueuedTime":"2024-04-13T18:06:07.5350000Z"}}
{"messageId":83,"deviceId":"Raspberry Pi Web Client","temperature":28.559585023456187,"humidity":77.22225167755144,"EventProcessedUtcTime":"2024-04-13T18:07:31.7629817Z","PartitionId":0,"EventEnqueuedUtcTime":"2024-04-13T18:07:31.5340000Z","IoTHub":{"MessageId":null,"CorrelationId":null,"ConnectionDeviceId":"declans-device","ConnectionDeviceGenerationId":"638486169586543168","EnqueuedTime":"2024-04-13T18:07:31.5280000Z"}}
{"messageId":84,"deviceId":"Raspberry Pi Web Client","temperature":31.5789939283847,"humidity":78.53645412248034,"EventProcessedUtcTime":"2024-04-13T18:08:18.4667883Z","PartitionId":0,"EventEnqueuedUtcTime":"2024-04-13T18:08:18.3170000Z","IoTHub":{"MessageId":null,"CorrelationId":null,"ConnectionDeviceId":"declans-device","ConnectionDeviceGenerationId":"638486169586543168","EnqueuedTime":"2024-04-13T18:08:18.3120000Z"}}

{"messageId":86,"deviceId":"Raspberry Pi Web Client","temperature":27.87426960061821,"humidity":77.85790781286164,"EventProcessedUtcTime":"2024-04-13T18:08:21.6387641Z","PartitionId":0,"EventEnqueuedUtcTime":"2024-04-13T18:08:21.4730000Z","IoTHub":{"MessageId":null,"CorrelationId":null,"ConnectionDeviceId":"declans-device","ConnectionDeviceGenerationId":"638486169586543168","EnqueuedTime":"2024-04-13T18:08:21.4680000Z"}}
{"messageId":89,"deviceId":"Raspberry Pi Web Client","temperature":26.644603869716587,"humidity":74.97705222666764,"EventProcessedUtcTime":"2024-04-13T18:08:27.6546655Z","PartitionId":0,"EventEnqueuedUtcTime":"2024-04-13T18:08:27.4730000Z","IoTHub":{"MessageId":null,"CorrelationId":null,"ConnectionDeviceId":"declans-device","ConnectionDeviceGenerationId":"638486169586543168","EnqueuedTime":"2024-04-13T18:08:27.4680000Z"}}
{"messageId":91,"deviceId":"Raspberry Pi Web Client","temperature":27.908676389338538,"humidity":75.03748011067242,"EventProcessedUtcTime":"2024-04-13T18:08:31.7014515Z","PartitionId":0,"EventEnqueuedUtcTime":"2024-04-13T18:08:31.4730000Z","IoTHub":{"MessageId":null,"CorrelationId":null,"ConnectionDeviceId":"declans-device","ConnectionDeviceGenerationId":"638486169586543168","EnqueuedTime":"2024-04-13T18:08:31.4680000Z"}}
{"messageId":92,"deviceId":"Raspberry Pi Web Client","temperature":31.304989220689027,"humidity":71.7004849565448,"EventProcessedUtcTime":"2024-04-13T18:08:33.7795209Z","PartitionId":0,"EventEnqueuedUtcTime":"2024-04-13T18:08:33.5510000Z","IoTHub":{"MessageId":null,"CorrelationId":null,"ConnectionDeviceId":"declans-device","ConnectionDeviceGenerationId":"638486169586543168","EnqueuedTime":"2024-04-13T18:08:33.5460000Z"}}

Part B)

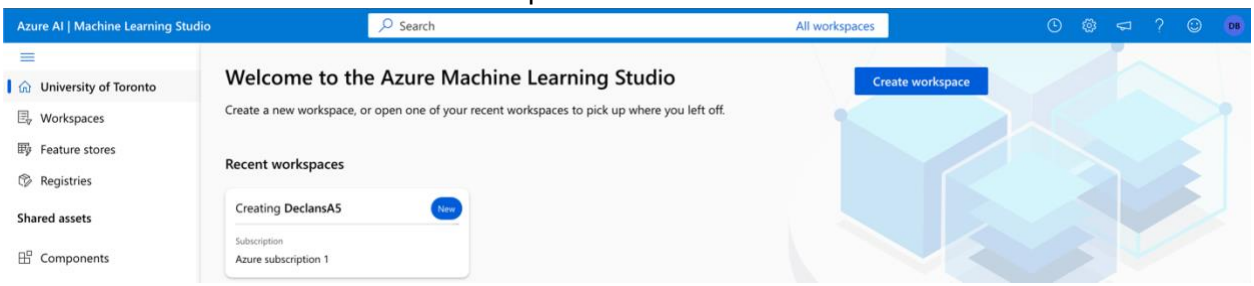
1. In my final assignment for MIE1628, I will be exploring a dataset that captures a range of biomedical voice measurements from 42 people with early-stage Parkinson's disease. These individuals were part of a six-month trial utilizing a telemonitoring device designed for the remote monitoring of symptom progression. The voice recordings were automatically captured in the patients' homes, providing a practical dataset that reflects real-world conditions.

   The dataset contains several attributes per recording, including the subject's number, age, gender, and the time elapsed since their baseline recruitment. It also details both motor UPDRS and total UPDRS scores (tools to measure the severity

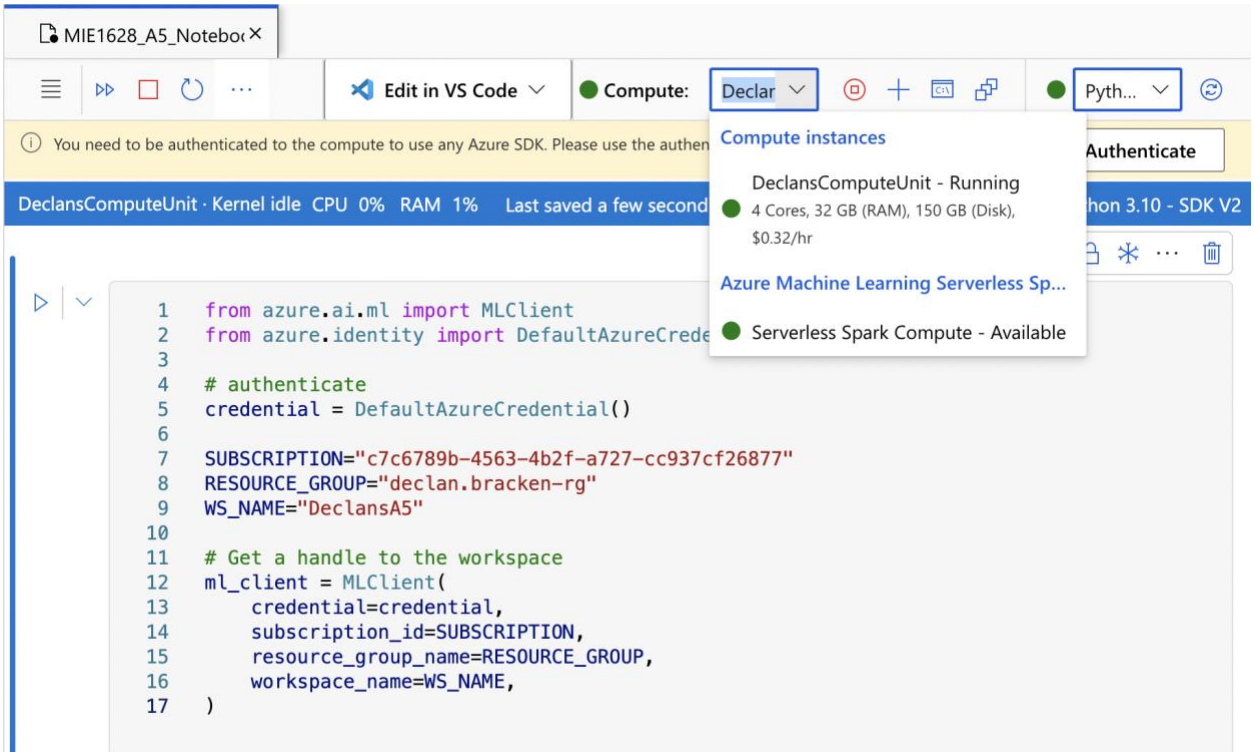
and progression of Parkinson disease), alongside 16 biomedical voice measures. There are 5,875 voice recordings in total. My main task is to develop a predictive model using these voice measures to estimate the motor and total UPDRS scores. This model could greatly aid in the remote healthcare domain by enabling early detection and continuous monitoring of Parkinson's disease progression through non-invasive methods. This project is an excellent opportunity to apply machine learning techniques in a meaningful way, potentially improving quality of life for individuals with Parkinson's disease by enhancing the capabilities of telemonitoring systems.

Link to dataset: LINK

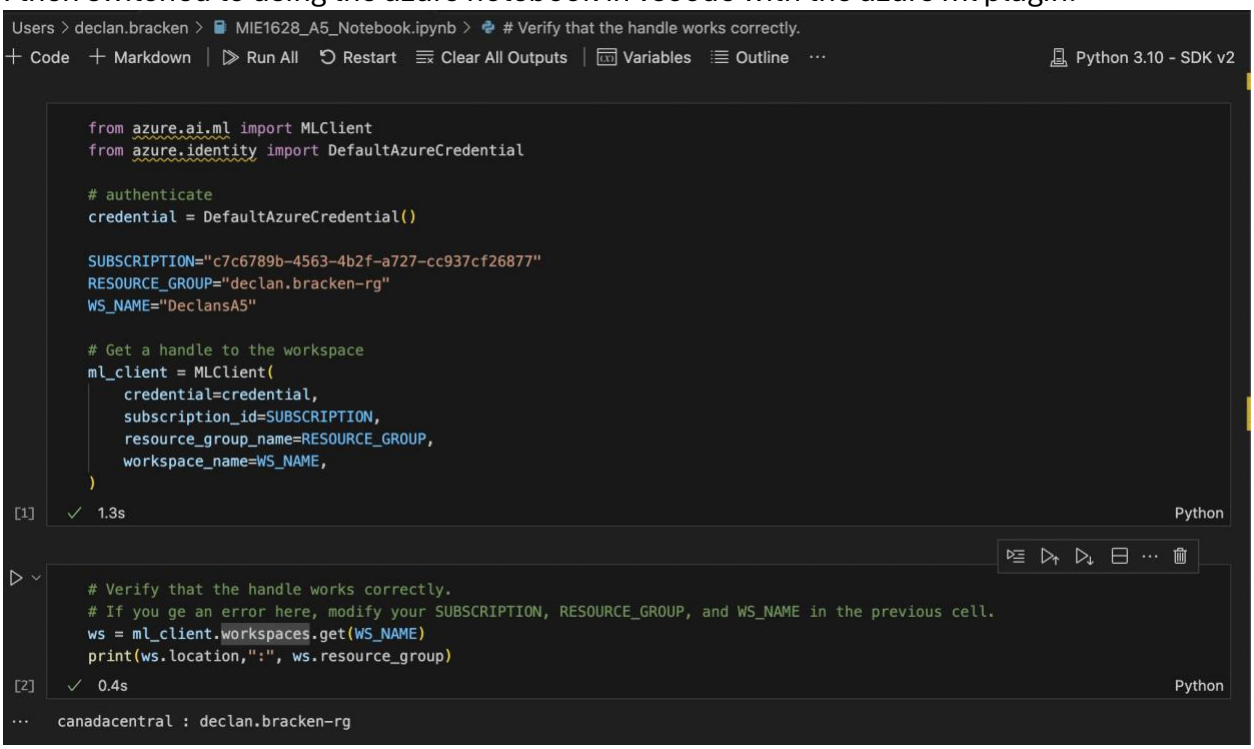
2. I created a new Azure ML workspace:



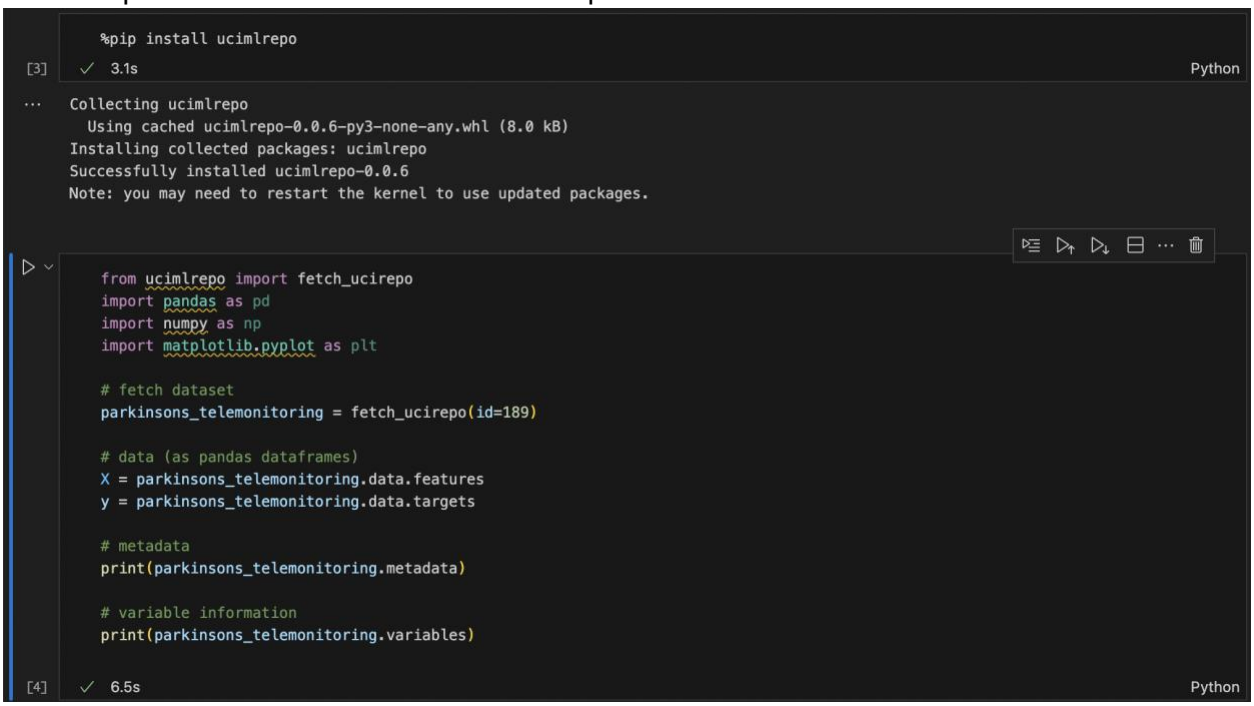
Then I created a new notebook and a new compute instance. I then go ahead and setup my subscription, resource group and name.



I then switched to using the azure notebook in vscode with the azure ml plugin:



Now I import the data from the UC Irvine repo:



The dataset I am working with is a structured collection of features and targets associated with a study of early-stage Parkinson's disease through voice measurements. It includes a

unique identifier for each subject (**subject#**) and a variety of both voice-related features and UPDRS scores, which are the primary targets of interest.

The dataset contains 22 columns, starting with the subject's age (**age**) and the time since recruitment into the trial (**test_time**). The majority of the features are continuous variables related to various measures of voice frequency and amplitude variability—these include **Jitter(%)**, **Jitter(Abs)**, **Jitter:RAP**, **Jitter:PPQ5**, **Jitter:DDP**, **Shimmer**, **Shimmer(dB)**, **Shimmer:APQ3**, **Shimmer:APQ5**, **Shimmer:APQ11**, and **Shimmer:DDA**. Two features, **NHR** and **HNR**, are ratios indicating the presence of noise in the voice. Additional features include **RPDE**, a non-linear dynamical complexity measure, **DFA**, a signal fractal scaling exponent, and **PPE**, a measure of fundamental frequency variation.

For prediction targets, there are two continuous scores: the **motor_UPDRS**, which is a motor examination score, and the **total_UPDRS**, which aggregates both motor and non-motor aspects of disability.

There is also a binary demographic feature, **sex**, indicating the subject's sex, where '0' denotes male and '1' denotes female.

All features are labeled as having no missing values, suggesting a complete dataset without the need for imputation (thankfully). Units are unspecified, implying standardization or that the measures are unitless ratios or scores. A full description and list is seen in the printed output which I've taken the liberty of presenting below.

The printed variables:

	name	role	type	demographic
0	subject#	ID	Integer	None
1	age	Feature	Integer	Age
2	test_time	Feature	Continuous	None
3	Jitter(%)	Feature	Continuous	None
4	Jitter(Abs)	Feature	Continuous	None
5	Jitter:RAP	Feature	Continuous	None
6	Jitter:PPQ5	Feature	Continuous	None
7	Jitter:DDP	Feature	Continuous	None
8	Shimmer	Feature	Continuous	None
9	Shimmer(dB)	Feature	Continuous	None
10	Shimmer:APQ3	Feature	Continuous	None
11	Shimmer:APQ5	Feature	Continuous	None
12	Shimmer:APQ11	Feature	Continuous	None
13	Shimmer:DDA	Feature	Continuous	None
14	NHR	Feature	Continuous	None
15	HNR	Feature	Continuous	None
16	RPDE	Feature	Continuous	None
17	DFA	Feature	Continuous	None

18	PPE	Feature	Continuous	None		
19	motor_UPDRS	Target	Continuous	None		
20	total_UPDRS	Target	Continuous	None		
21	sex	Feature	Binary	Sex		

	description	units	missing_values
0	Integer that uniquely identifies each subject	None	no
1	Subject age	None	no
2	Time since recruitment into the trial. The integer part is the number of days since recruitment.	None	no
3	Several measures of variation in fundamental frequency	None	no
4	Several measures of variation in fundamental frequency	None	no
5	Several measures of variation in fundamental frequency	None	no
6	Several measures of variation in fundamental frequency	None	no
7	Several measures of variation in fundamental frequency	None	no
8	Several measures of variation in amplitude	None	no
9	Several measures of variation in amplitude	None	no
10	Several measures of variation in amplitude	None	no
11	Several measures of variation in amplitude	None	no
12	Several measures of variation in amplitude	None	no
13	Several measures of variation in amplitude	None	no
14	Two measures of ratio of noise to tonal components in the voice	None	no
15	Two measures of ratio of noise to tonal components in the voice	None	no
16	A nonlinear dynamical complexity measure	None	no
17	Signal fractal scaling exponent	None	no
18	A nonlinear measure of fundamental frequency variation	None	no
19	Clinician's motor UPDRS score, linearly interpolated	None	no
20	Clinician's total UPDRS score, linearly interpolated	None	no
21	Subject sex '0' - male, '1' - female	None	no

13 None no
14 None no
15 None no
16 None no
17 None no
18 None no
19 None no
20 None no
21 None no

To further analyze the data I've chosen the following figures:

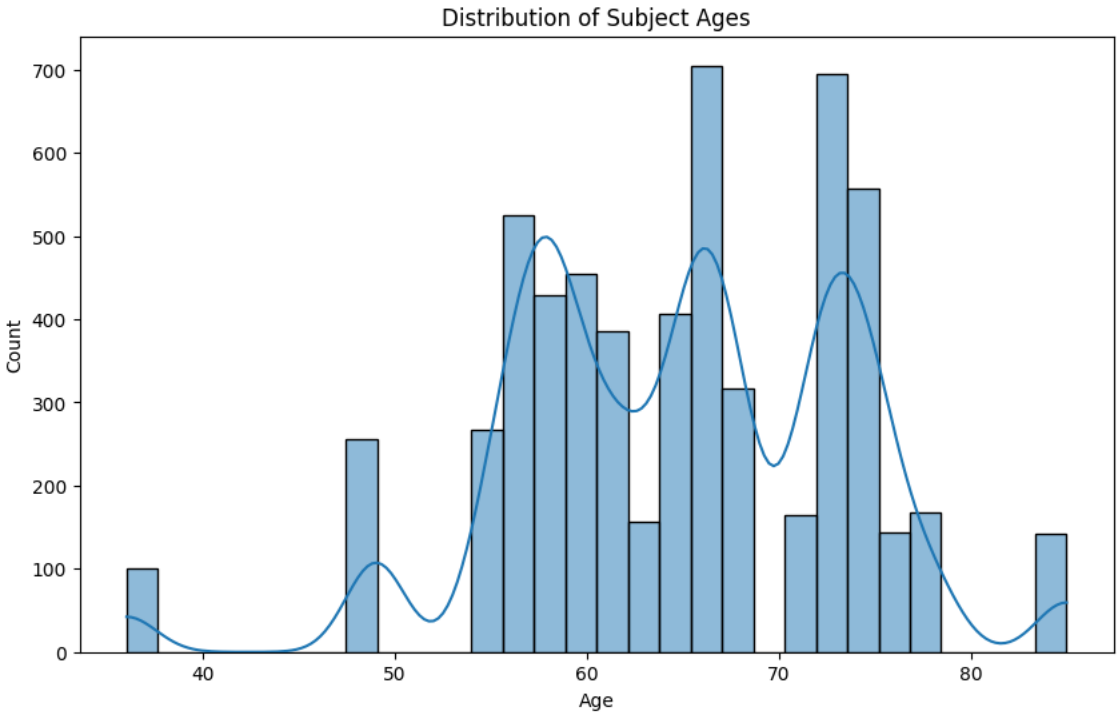


Figure 1: Histogram of patient age.

The distribution of subject age shows a fairly wide spread between the ages of candidates, with some as young as 36, and others as old as 85. Because of the relatively few number of candidates, there are several gaps in the distribution.

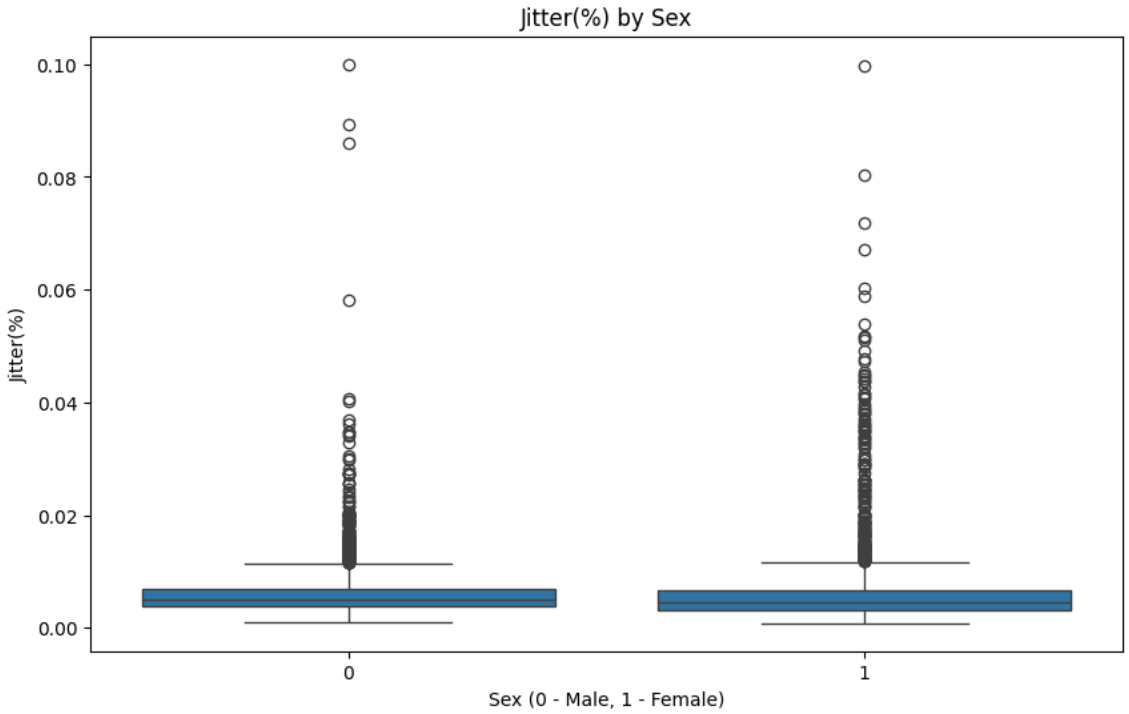


Figure 2: Jitter (%) as a function of sex.

Jitter, which as the name suggests is a measure of vibrational control in the vocal chords, is a common indicator used for tonal and acoustic voice analysis. This graph shows no significant differences between the average or max/min jitter between male and female patients.

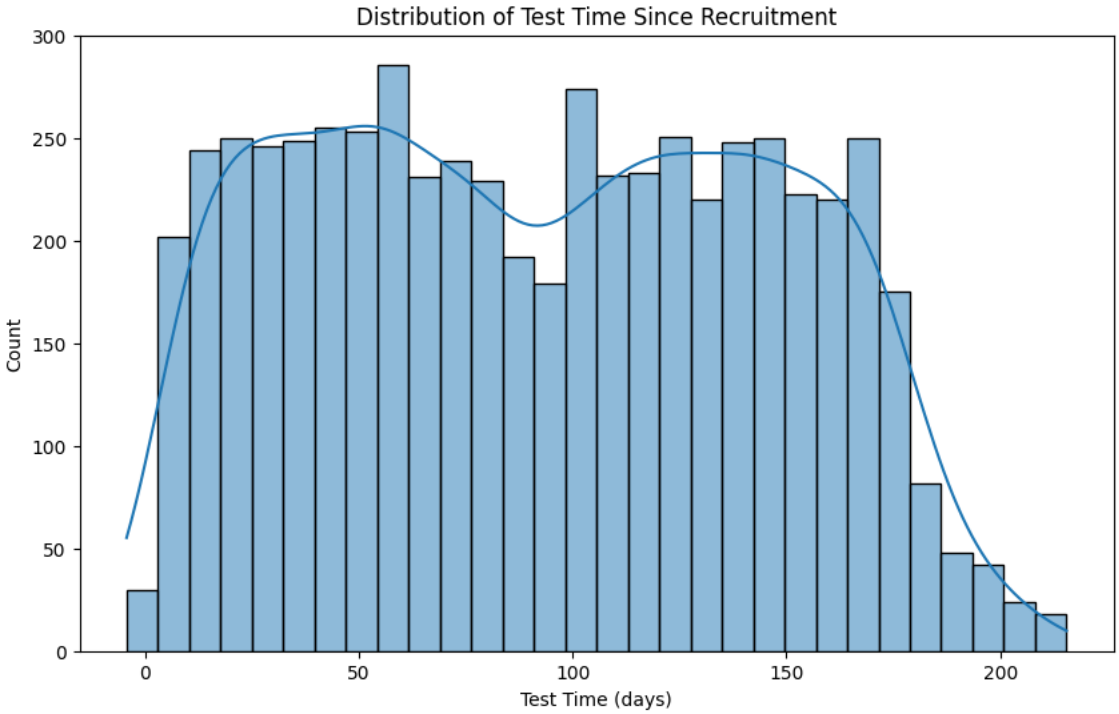


Figure 3: Histogram of the test time since patient recruitment to the study.

The test time since recruitment indicates the number of days since the client was recruited into the trial that have passed before the test sample was taken. As we can see, the distribution is primarily even, but slightly bimodal with a dip near the midway point. This could indicate that samples were timed with a first vs second half staging in mind. One might predict that the patients UPDRS scores will increase as time since recruitment increases, since that will give more time for symptom development.

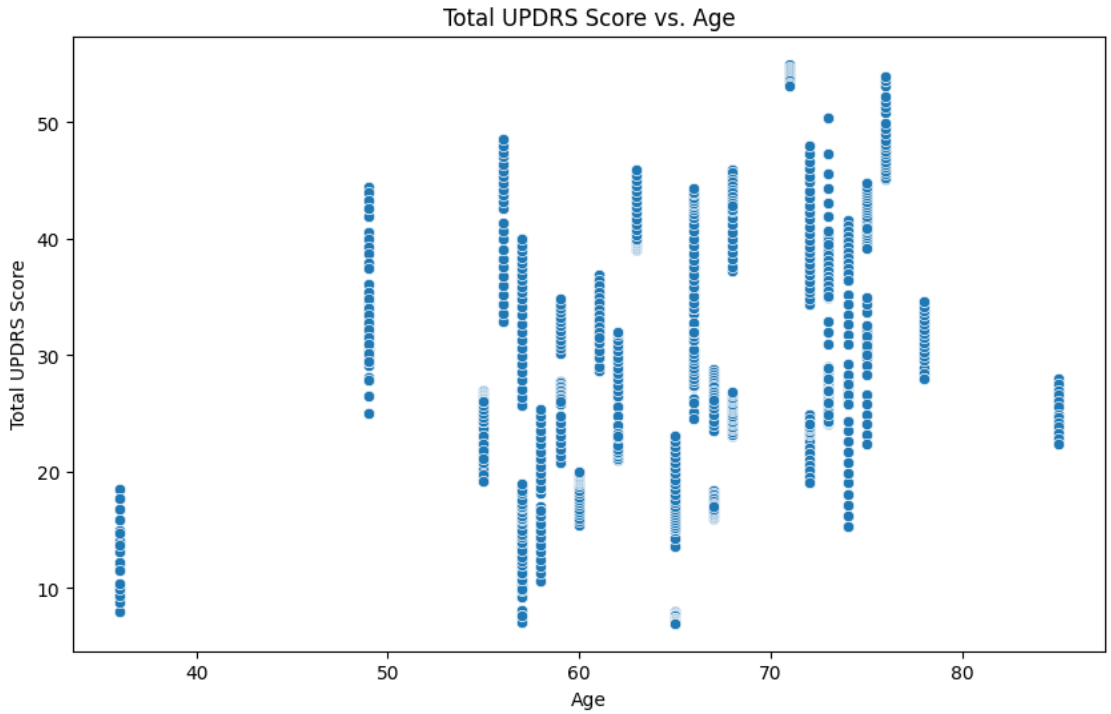


Figure 4: Total UPDRS scores as a function of patient age.

The Total UPDRS is the Unified Parkinson's Disease Rating Scale. It incorporates a series of factors, observations about the patient, and tests to create an index rating which utilizing the following components:
UPDRS components.

- Part I: Mentation, Behavior, Mood
- Part II: Activities of Daily Living (Determine for "on" or "off", indicating either a "good" or "bad" day, respectively.)
- Part III: Motor Examination
- Part IV: Complications of Therapy (in the past week)
- Part V: H&Y staging scale
- Part VI: S&E ADL scale

Source: American Physical Therapy Association (APTA Link

My final graph is a feature-wise pairplot showing the relationships between different features and also the targets (total and motor UPDRS scores). The goal of this figure is to allow us to examine many differenct feature relationships at once to find trends.
For the pairplot, you can examine the axes labels and plots more closely by zooming in (the resolution should dynamically improve)

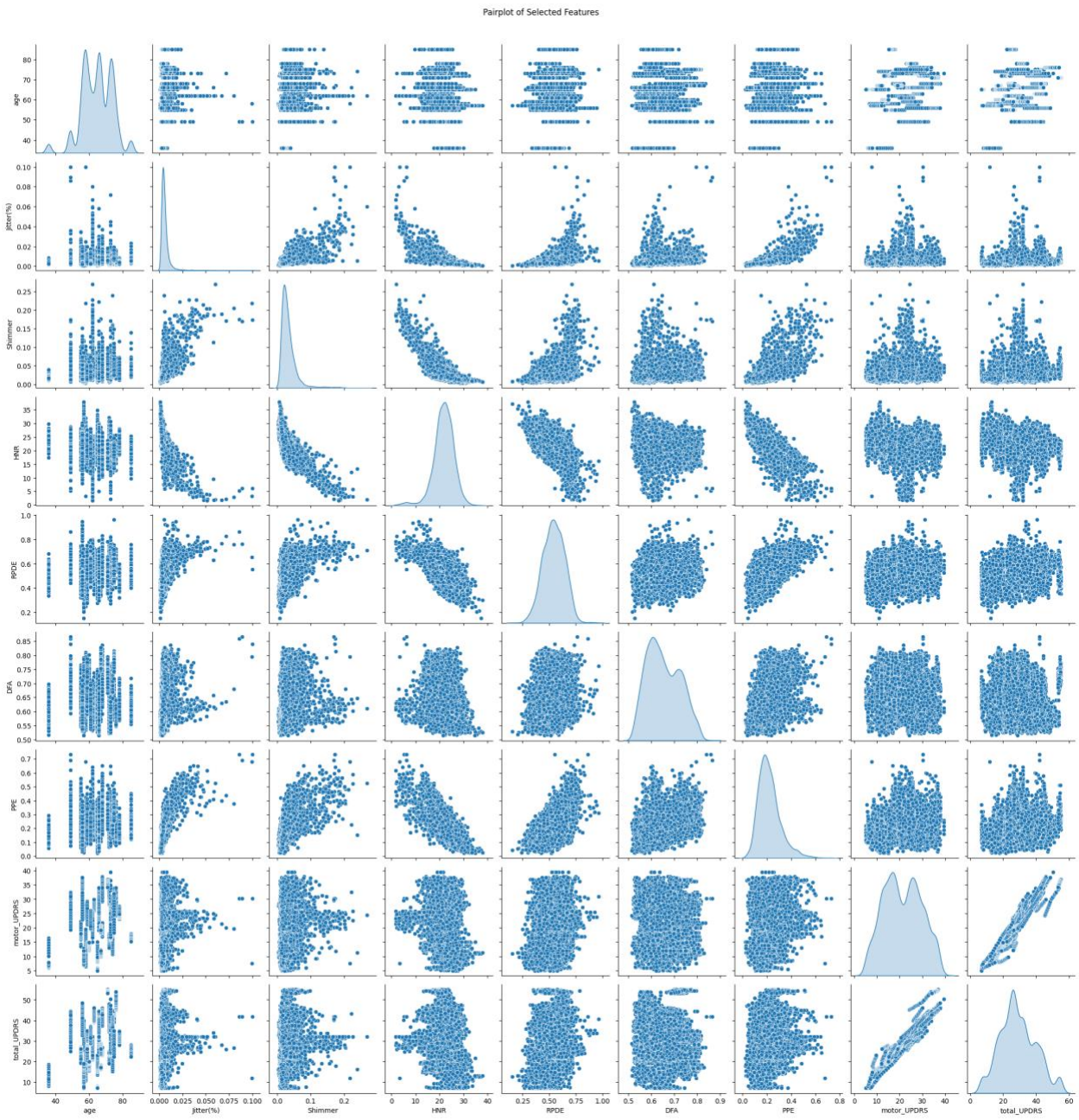


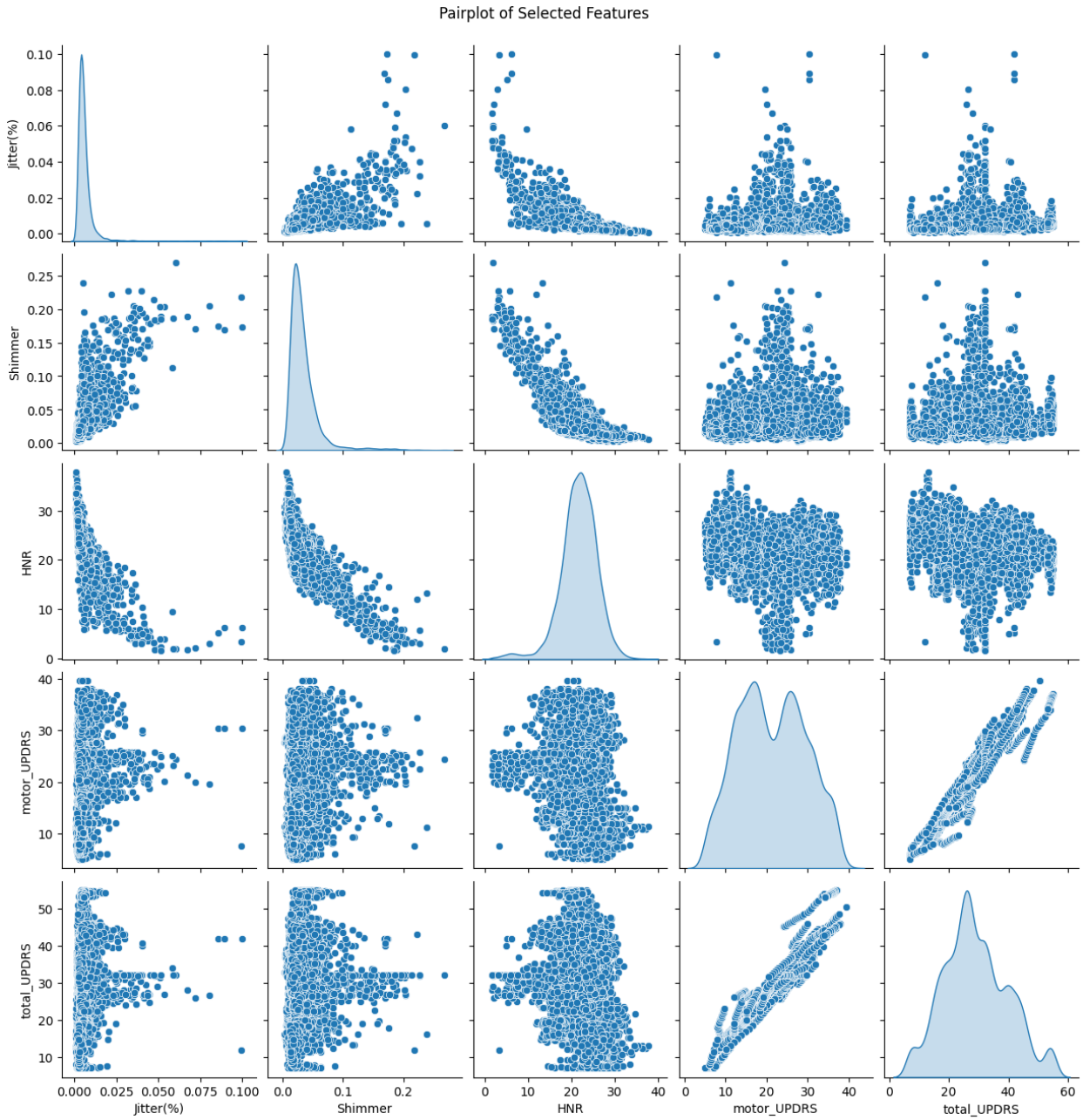
Figure 5: Pair plot for several key features and the targets. Please zoom in to inspect with greater detail.

Note that the diagonal shows the variance of the feature (like a covariance matrix).

Preprocessing and Feature Engineering

Since there are no missing values in this dataset, imputing is unnecessary. I understand there are marks for this section, so instead of just cleaning, I'll perform some feature engineering, to do this let's examine the pair plots. We notice some interesting relationships, and one of the more subtle, but important one's is the way in which features

like HNR, Shimmer, and Jitter either spike or drop on the midway point of the UPDRS scores.



Here's what I want to do. To encode some meaning in when the shimmer or jitter is above a certain value, I'll create either ordinal or binary indicators at some observed cutoff/range. This will hopefully help the model recognize some that the majority of high jitter/shimmer observations fall within the median UPDRS range. I decided to create an ordinal indicator for Jitter between 0 – 0.015, 0.015 – 0.3, and 0.3+. I then made a binary indicator for Shimmer, either above or below 0.09, and a binary indicator for HNR, either above or below 10.

Additionally, we notice that the relationships between jitter, shimmer and HNR are non-linearly correlated. To capitalize on this, I'm going to create new features acting as the logarithmic transformation of the original 3 features previously mentioned. Comparing before and after log transformation:

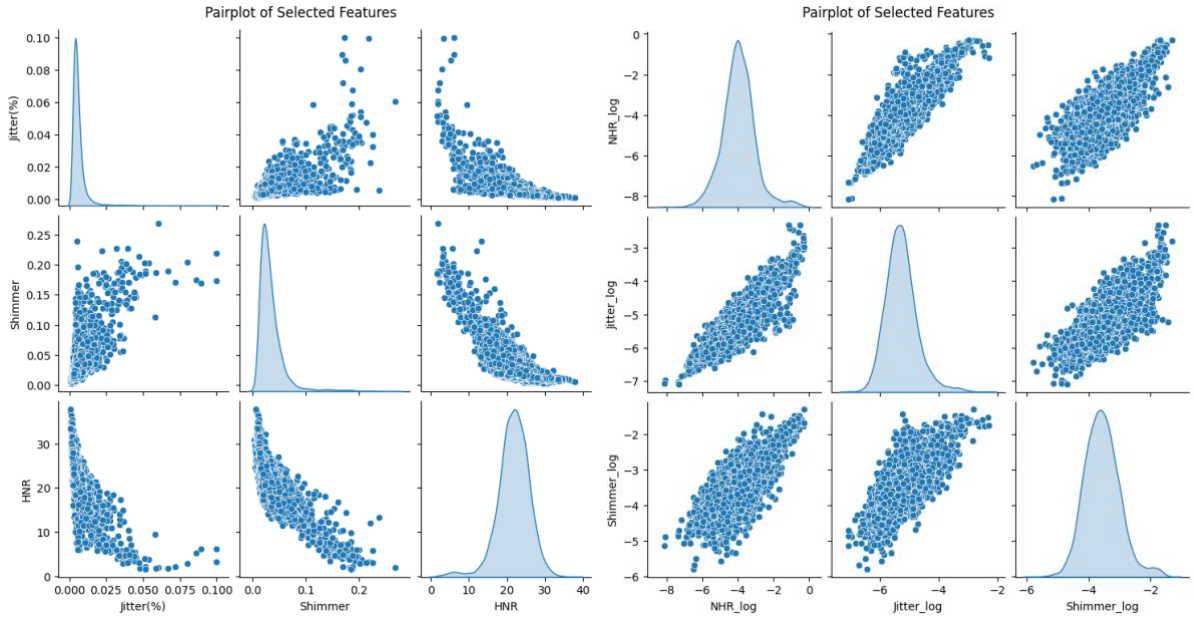


Figure 6: Left: Original Jitter, Shimmer, and HNR pairplots. Right: Log-transformed Jitter, Shimmer, and HNR pairplots.

We can see that by performing logarithmic transformations of the data, the resulting distributions become linearized, and the intra-feature distributions also become more normal with less skewness and kurtosis. Though this doesn't necessarily mean that these features become more correlated to the targets, hopefully this leads to improved model accuracy.

I want to also mention that I had considered performing data standardization on all my non-unit specific features (everything apart from the test time, participant age, and participant sex). I found that this ruined some of the correlations I found between features (such as the ones above) and ended up deciding to leave out standard scaling for simplicity and preservation.

Modelling

I chose to use a Random Forest and XGBoost model for my dataset. These are two sophisticated machine learning algorithms that are widely used for regression tasks due to their robustness, versatility, and generally strong performance.

Random Forest operates by constructing a multitude of decision trees at training time and outputting the average prediction of the individual trees. This ensemble approach helps to reduce the risk of overfitting, which is common with single decision trees. For regression tasks, this means that Random Forest can capture complex, non-linear relationships without being too sensitive to noise in the training data. It works well with a mix of numerical and categorical features and can handle large datasets efficiently. Furthermore, Random Forest has the added benefit of being relatively easy to tune and often performs quite well with default hyperparameters. Its ensemble nature makes it more robust than individual decision trees and less likely to be thrown off by outliers.

XGBoost, which stands for Extreme Gradient Boosting, is an implementation of gradient boosted decision trees designed for speed and performance. XGBoost is particularly valued for its performance in predictive accuracy and computational efficiency. It applies the principle of boosting, where trees are added one at a time, and each tree tries to correct the mistakes of the previous one. In a regression context, XGBoost is capable of capturing complex non-linear patterns in the data by optimizing its loss function using gradient descent. The model is highly customizable with a rich set of hyperparameters that can be fine-tuned for better performance. One of the key advantages of XGBoost is that it includes a regularization term in the loss function, which helps prevent overfitting, a common challenge in regression tasks.

Both models also provide feature importance scores, which can be incredibly insightful when trying to understand the driving factors behind the predictions. This can be particularly useful in a medical context where interpretability is as important as predictive power. This proves to be critical during my modelling process…

I instantiated baseline hyperparameters for either model and trained to incredibly high accuracy immediately, yielding and RMSE for predicted UPDRS of 1.9 and 3.4, with R squared values of 0.97 and 0.96 for the random forest and XGboost models respectively. This was somewhat shocking, because in analyzing the feature graphs, I didn't see many correlations between feature values and the targets. I decided to check on the feature importances for either model, which is when I figured out that *age* made up most of the explained variance… then I realized that with such a small sample of patients (under 50), the ages for each patient effectively acted as a patient ID, and that allowed the model to memorize which age roughly corresponded to which UPDRS score, based on the patient. Since I didn't split my train, validation, and test sets by patient I effectively created data leakage.

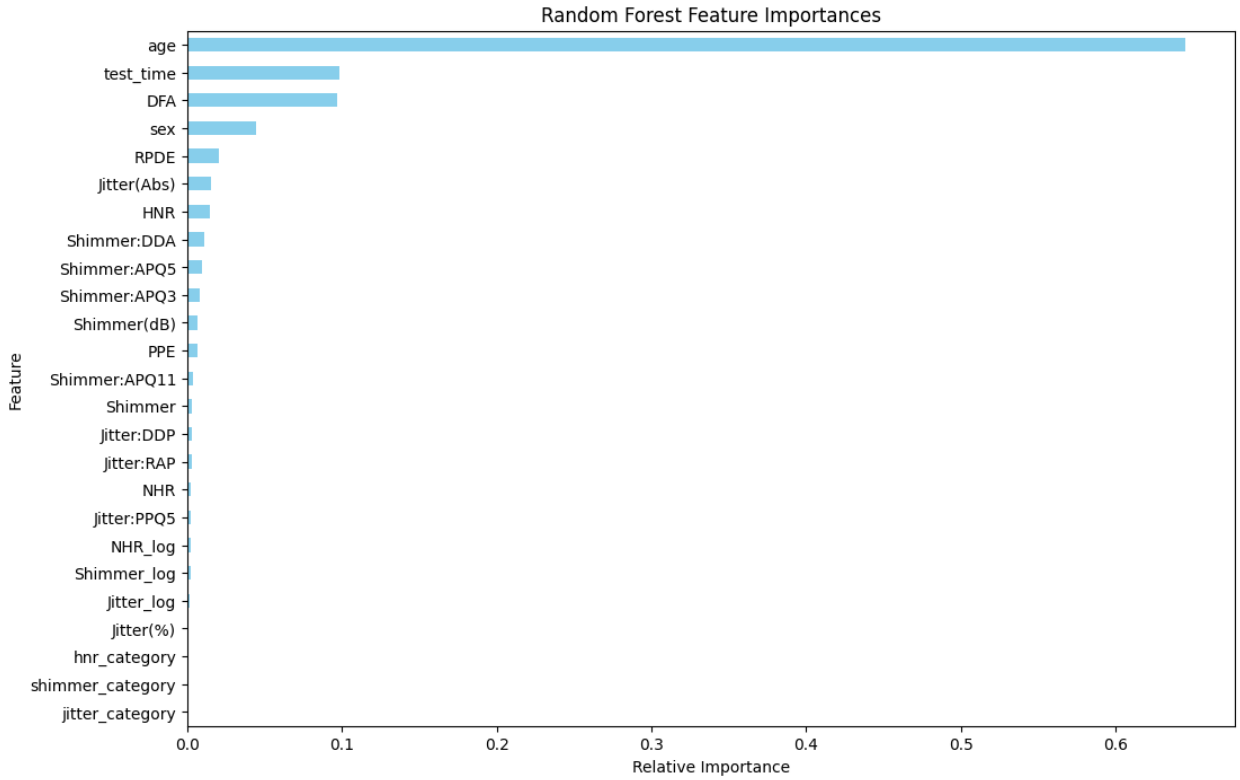


Figure 7: Feature importance for Random Forests Algorithm.

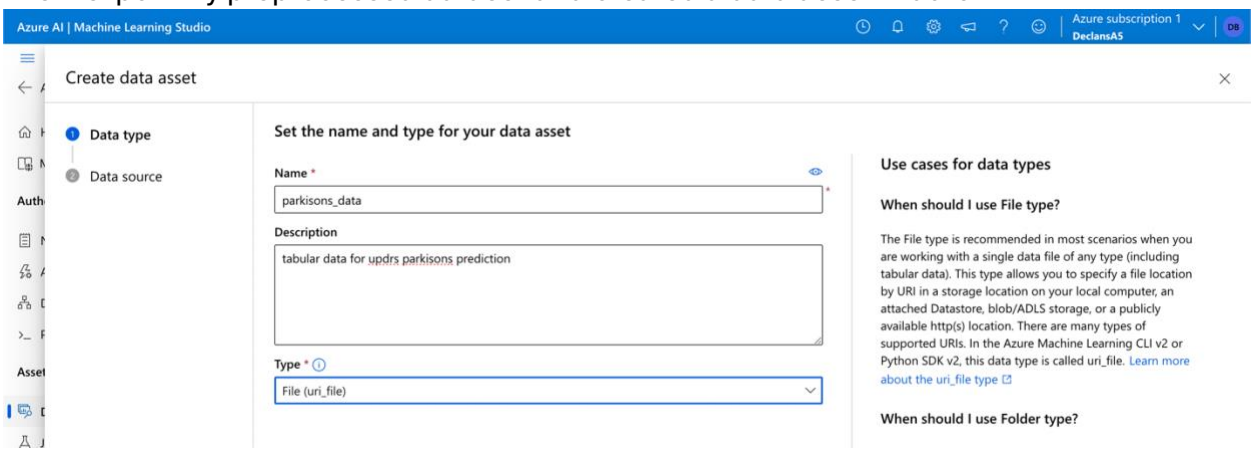
I decided to perform grouped splitting based on patient ID, this resolves the issue of having unique ages correspond to unique patients, and therefore unique UPDRS scores. Upon performing modelling however, I'm hit with a stark result; The random forest yields an RMSE of 13 and R squared of -0.44, and the XGBoost algorithm results in nearly the exact same values. What does this mean? Essentially this would indicate that there is no correlation, non-linear or simple, between the acoustical characteristics of patients and their UPDRS scores. I tried a third modelling method as well to see if perhaps the random forest ensemble methods were the problem but to no avail: using linear regression also yielded a disturbing RMSE of 11.5 and R squared of -0.12. The negative R squarred value for all 3 models indicates that the model is somehow less accurate than just taking the mean of the feature data and drawing a horizontal line.

Now you must understand that I've spent a long time on this problem, and I cannot resolve it. When analyzing figure 5, it becomes apparent that either there's simply not enough individual patients to create meaningful relationships between features and resulting UPDRS, or there is no relationship anyways. In order to determine whether I've just made a mistake somewhere, or if there's really no correlation to be found in this dataset, I searched for an original paper which analyzed this exact dataset and was published to the Journal of Cloud Computing (LINK). In this paper, they find the same result: "it can be seen that there is a strong correlation between motor UPDRS and total UPDRS; shimmer and jitter parameters have a moderate correlation with each other. In contrast, there is a low correlation between all other parameters and total UPDRS". Unfortunately for me, this

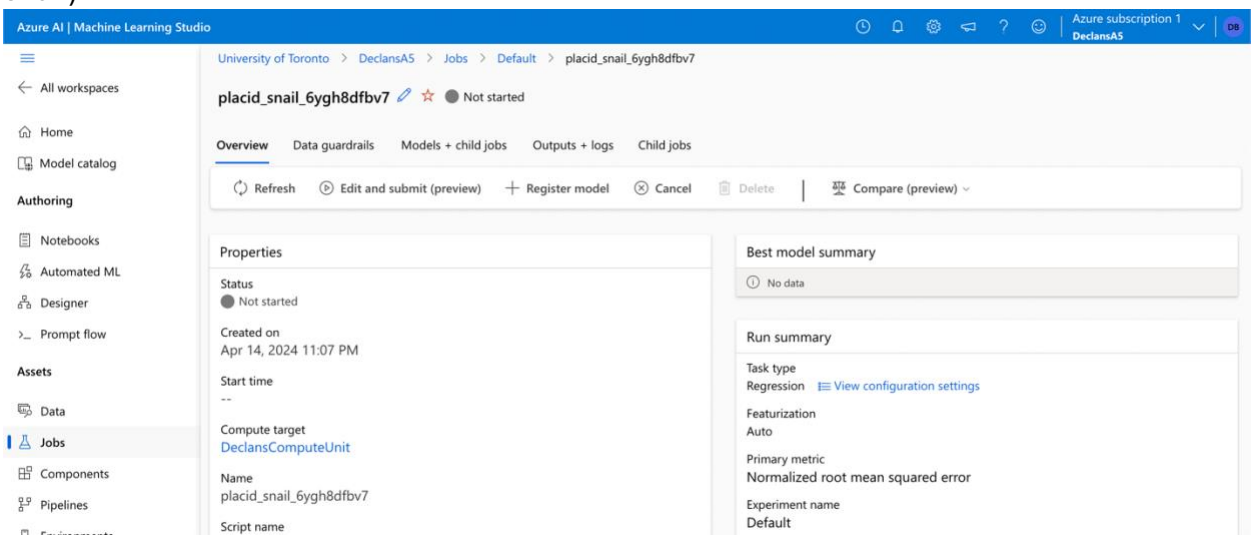
means that the modelling has been unfruitful. I hope that whoever is marking can understand my frustration in this matter, and not mark me too harshly on this section. I truly believe I chose a bad dataset for this assignment, based purely on the results.

Automated ML in Azure

To truly determine whether or not my dataset can yield good results, I'm going to implement an automated ML task in Azure.
First I export my preprocessed dataset and created a data asset in autoML:



I specify the computational limits, my compute resource, and basic info. I then run my job for a maximum of 15 minutes, since the dataset isn't very large (my job is called placid snail):



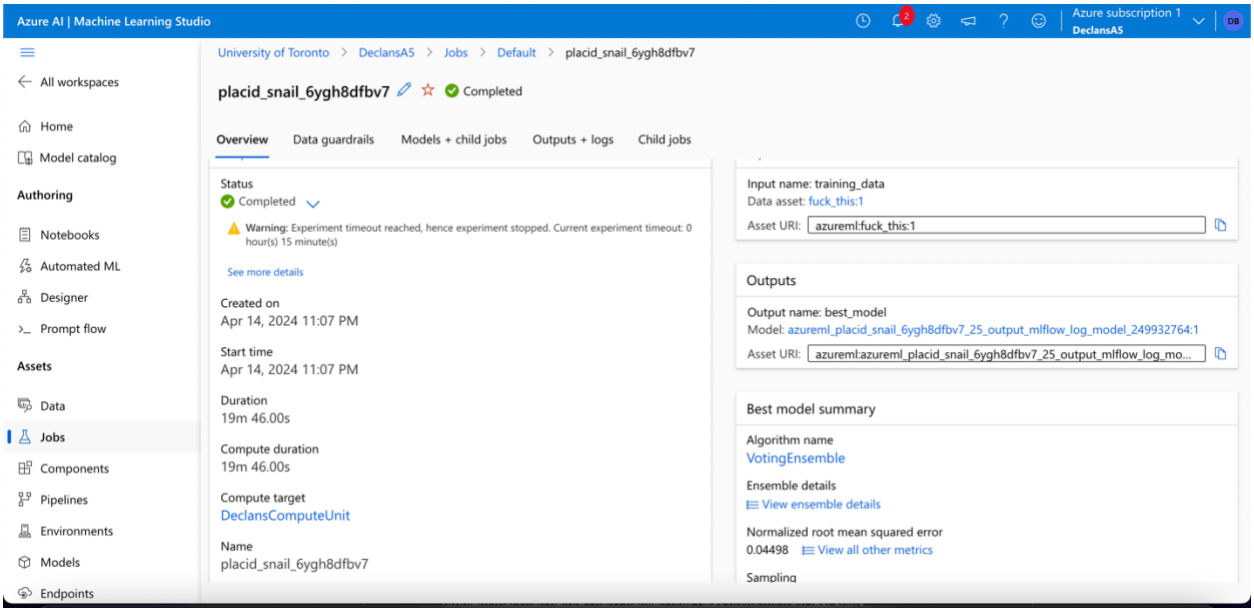
I then ran the model for nearly 20 minutes which resulted in the best model utilizing an XGBoost Regressor (like what I used in my own code) with the following resulting metrics:

Run Metrics

Explained variance0.95955

Mean absolute error1.3724
Mean absolute percentage error6.3005
Median absolute error0.87068
Normalized mean absolute error0.028597
Normalized median absolute error0.018142
Normalized root mean squared error0.044980
Normalized root mean squared log error0.052889
R2 score0.95951
Root mean squared error2.1587
Root mean squared log error0.10291
Spearman correlation0.98040

Screenshot:



Based on the results from Azure's Automated ML for predicting total UPDRS scores, it appears that the model performed exceptionally well on the given dataset, with an R2 score of 0.95951, which indicates that 95.951% of the variance in UPDRS scores is predictable from the features used by the model. The Spearman correlation coefficient of 0.98040 supports this strong relationship, suggesting a high correlation between the predicted values and the actual values. However, the exceptionally low errors, such as a mean absolute error (MAE) of 1.3724 and a root mean squared error (RMSE) of 2.1587, need to be critically examined, given the understanding that the model may have overfit to the age feature.

The issue still seems to arise from the inclusion of patient age, which, in the context of this dataset, is acting as a proxy for the patient identifier. If age is not appropriately randomized across patients and is instead a static variable (for example, if a patient's age does not

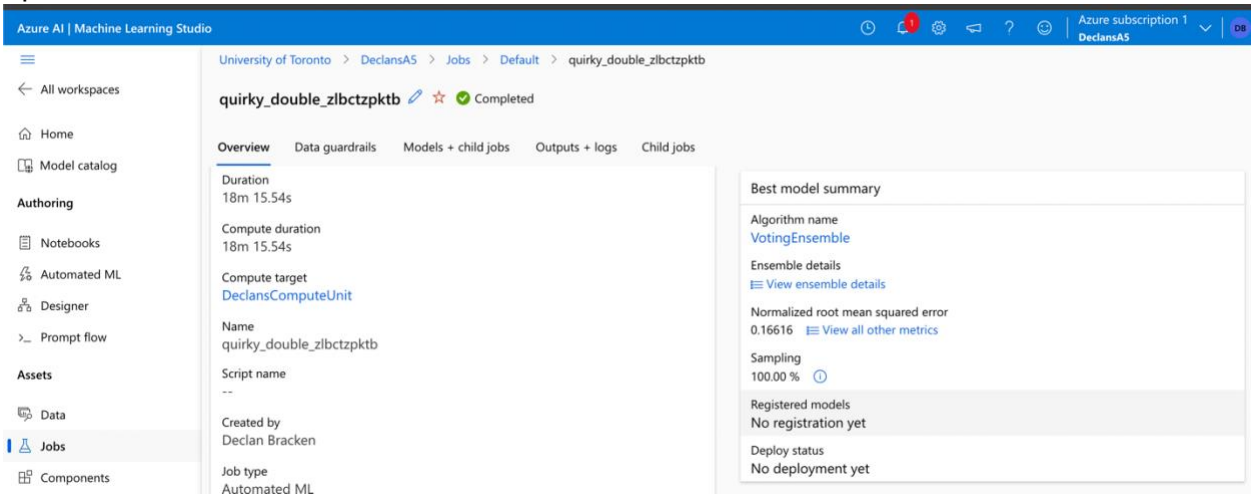
change over the course of different records), the model will learn to associate this age with the progression of the disease as measured by the UPDRS score. This would not be a generalizable finding; rather, it would be an artifact of this dataset's structure where multiple records for a patient have the same age value. In essence, the model is memorizing UPDRS scores based on age, rather than learning meaningful patterns from the acoustical features that are supposed to be predictive of the disease's progression.

To ensure that the model is capturing the true underlying patterns and not simply memorizing scores, age should either be removed from the feature set or the data should be split in such a way that ensures the model cannot simply learn to predict UPDRS scores based on patient age. Moreover, further validation using an unseen dataset or a cross-validation approach that includes age variation for each patient across folds would be necessary to assess the model's ability to generalize.

So with this in mind, I ran another ML job, this time I removed age as a feature, which resulted in significantly worse results:

Run Metrics

Explained variance0.44856
Mean absolute error6.1860
Mean absolute percentage error27.247
Median absolute error5.0913
Normalized mean absolute error0.12890
Normalized median absolute error0.10609
Normalized root mean squared error0.16616
Normalized root mean squared log error0.15704
R2 score0.44805
Root mean squared error7.9745
Root mean squared log error0.30556
Spearman correlation0.64357



Upon removing the age feature, which previously allowed the model to overfit to patient-specific data, the Automated ML results indicate a significant change in performance. The R2 score dropped to 0.44805, suggesting that now only approximately 44.805% of the variance in the UPDRS scores can be explained by the model, which is a notable decrease from the initial results. The Spearman correlation coefficient also decreased to 0.64357, reflecting a weaker monotonic relationship between the predicted and actual values.

The mean absolute error (MAE) increased to 6.1860, and the root mean squared error (RMSE) is now 7.9745, both of which are considerably higher than before. This is expected as the model is no longer leveraging the age feature, which likely accounted for a significant portion of the variance in UPDRS scores due to the dataset's structure rather than genuine predictive capability.

These results highlight the challenge of creating a model that accurately predicts clinical outcomes based solely on non-invasive biomedical voice measurements without overfitting to demographic features. The lower performance metrics may present a more realistic picture of the predictive power of the model using only the given acoustical features. It emphasizes the importance of having a diverse set of features and the need for careful consideration of feature selection to avoid overfitting and to build models that generalize well to new data.