AWS re:Invent

NOV. 29 – DEC. 3, 2021 | LAS VEGAS, NV

AUT301

Toyota Racing uses deep learning to predict race outcomes

Steve Miller
Principal Consultant
AWS Professional Services

Derek Jones
Data Manager
Toyota Racing Development

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Speakers



- Steve Miller (ssmiller@amazon.com)
- Principal Consultant
- Focus on motorsports and automotive customers
- Passion for data science and data engineering



- Derek Jones (derek.v.jones@toyota.com)
- Data Manager
- Focus on delivering new solutions to help win races
- Previous experience implementing ML models in the entertainment industry

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Introduction to stock car racing and TRD

Discussion on what makes race car drivers fast

How we translated this into an ML problem

Deep dive on neural networks and embeddings

Behind-the-scenes look at our model and code

How we use dashboards to make decisions

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Introduction

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Stock Car Racing 101



- Popular car racing series in the USA
- Most tracks are oval
- Marketing tool for automakers
- More laps (200+) than F1 and also more cars (40+)
- Rules are very restrictive
- Chassis and engine design have big impact on winning

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Toyota Racing Development USA





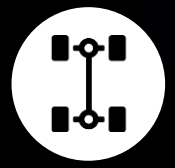
- Commonly referred to as TRD
- Subsidiary of Toyota Motors North America (TMNA)
- Part of Toyota's broader marketing strategy
- Primary business outcome: Winning races
- Race in NASCAR, NHRA, IMSA, and many other series
- Capabilities to design, manufacture, and build engines, and provide aero modeling, engineering, and software implementation to Toyota-sponsored teams
- Apps help drivers and pit crews make the right decisions

aws

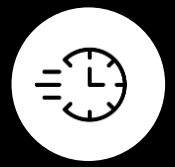
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

TRD championship-winning app suite

Racing applications*



Simulation apps to help determine the **chassis setup** that will yield the best lap time



Situational awareness and race prediction apps that provide **real-time scoring** and telemetry



Historical data mining apps to **compare performance** between drivers

Results

- ✓ **Won** 75+ NASCAR cup races since 2015
- ✓ **Won** 3 of the last 6 manufacturer championships for NASCAR cup series
- ✓ 600+ weekly users
- ✓ 30+ TB data lake

*Representative sample of 20+ applications

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Chalk Talk

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Driver development pipeline



- The more races you win, the more cars you sell, the more money you make
- Winning races year after year is a function of the driver development pipeline
- Identifying the best young drivers and getting them into the pipeline is critical
- Drivers can come from any series and any type of racing

aws

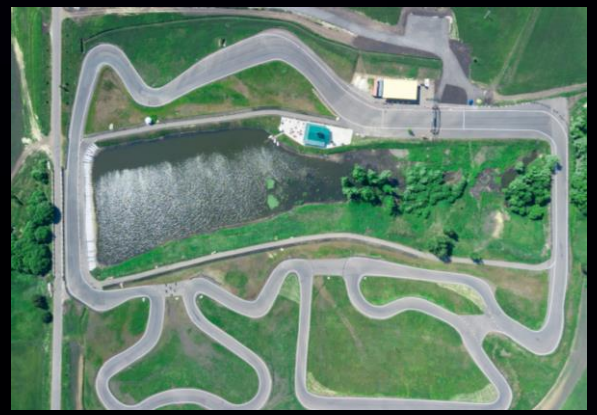
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

But what makes a driver good?

Starting position?



Track type?



Series?



Money?

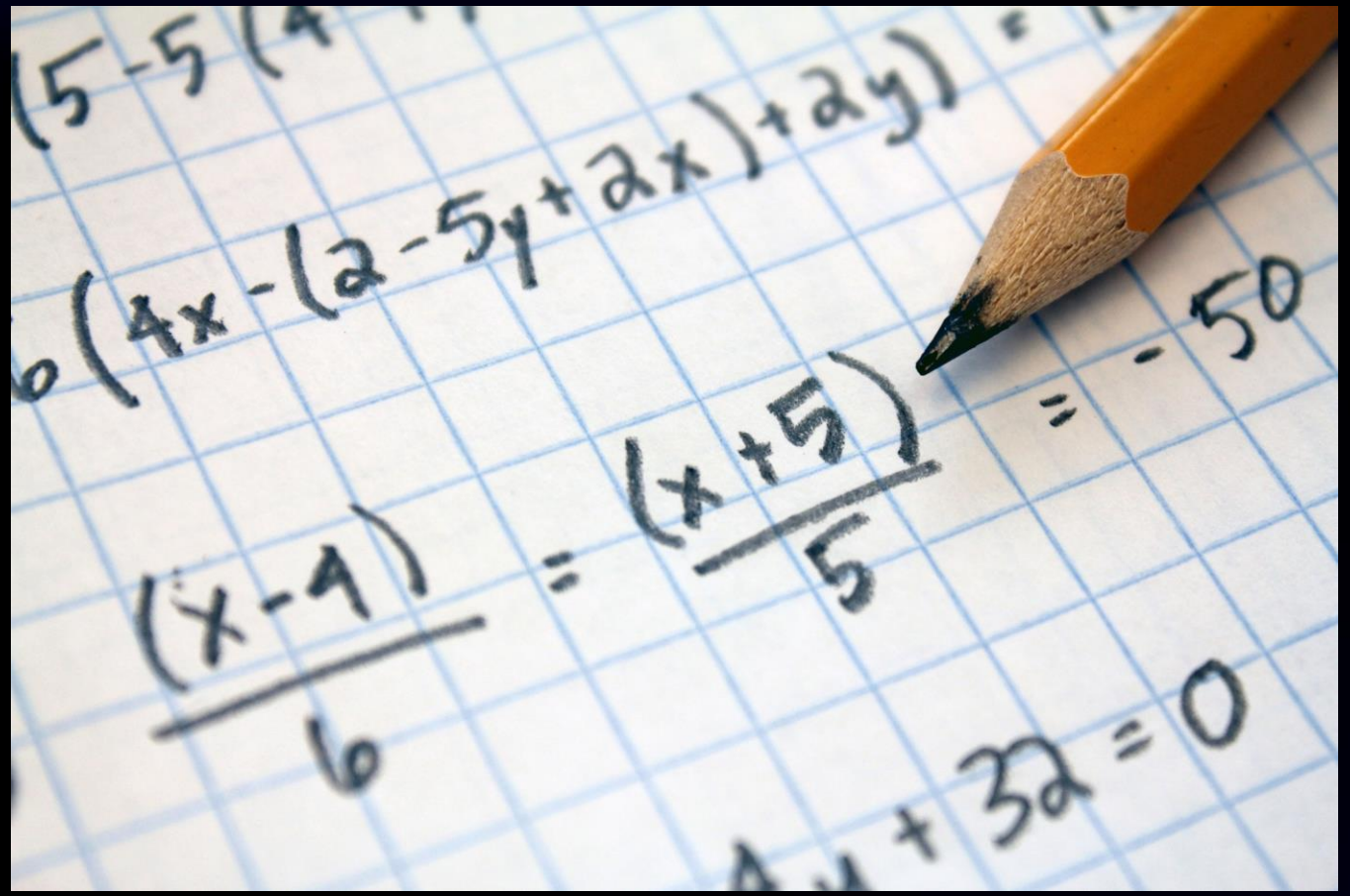


Thoughts from the audience?

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How do we translate this into an ML problem?



Problem

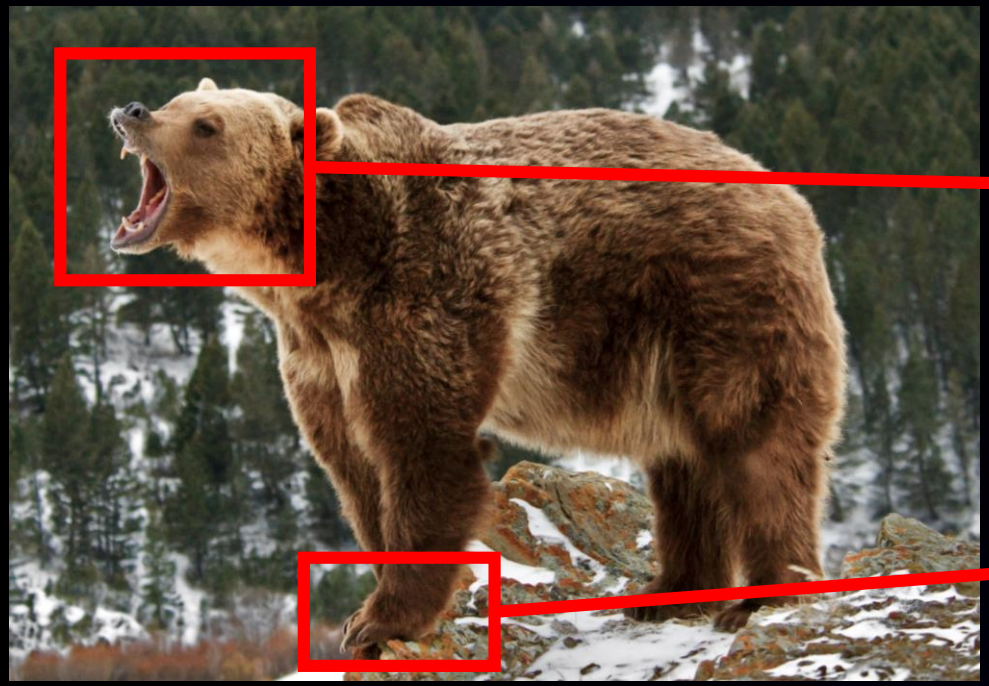
How do you numerically describe and normalize so many different categorical variables and all the possible interactions they have with each other?

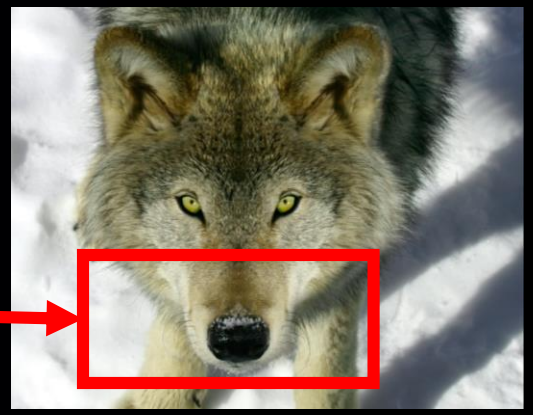
aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Our brains are really good at this

If you never saw a bear before, how would you know that it's dangerous?







We can identify the type of a new animal because it has features from animals that we already know

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

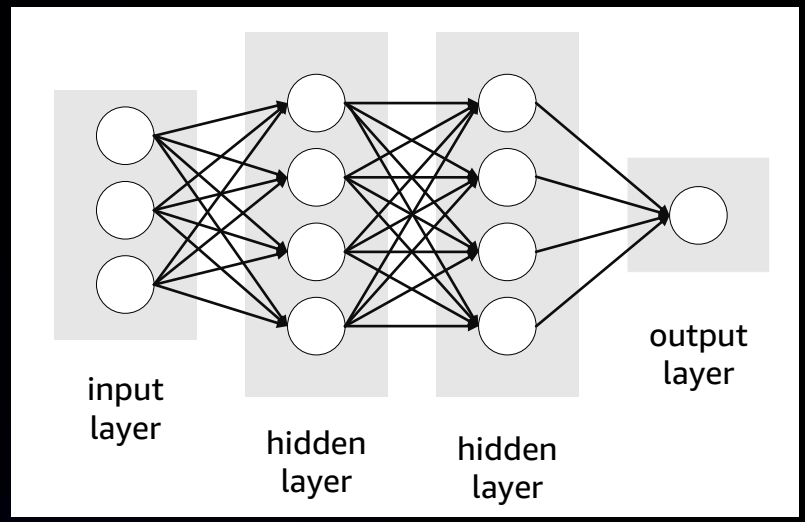
Neural networks do the same thing

Human brain



Our brains have a bunch of neurons that receive inputs from other neurons and fire in a unique sequence to achieve an outcome

Neural network



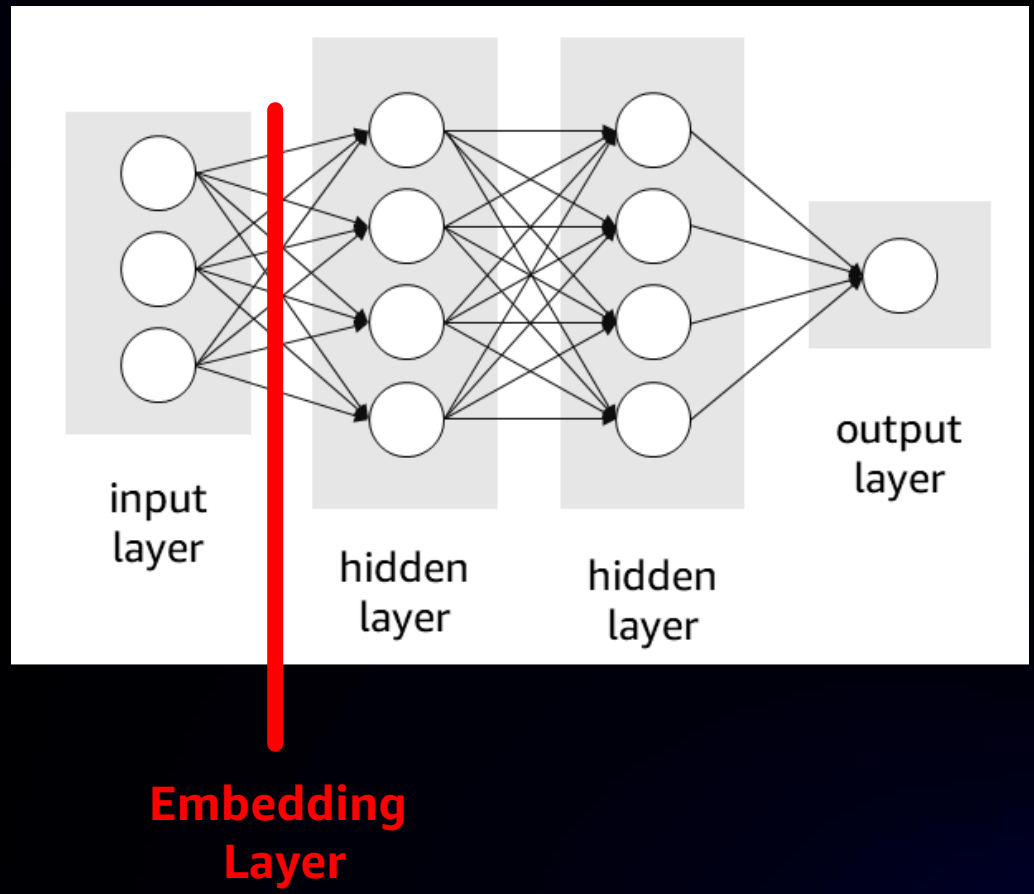
We do a very similar thing in machine learning where we receive a bunch of inputs and pass them through a series of nodes which delivers an output

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

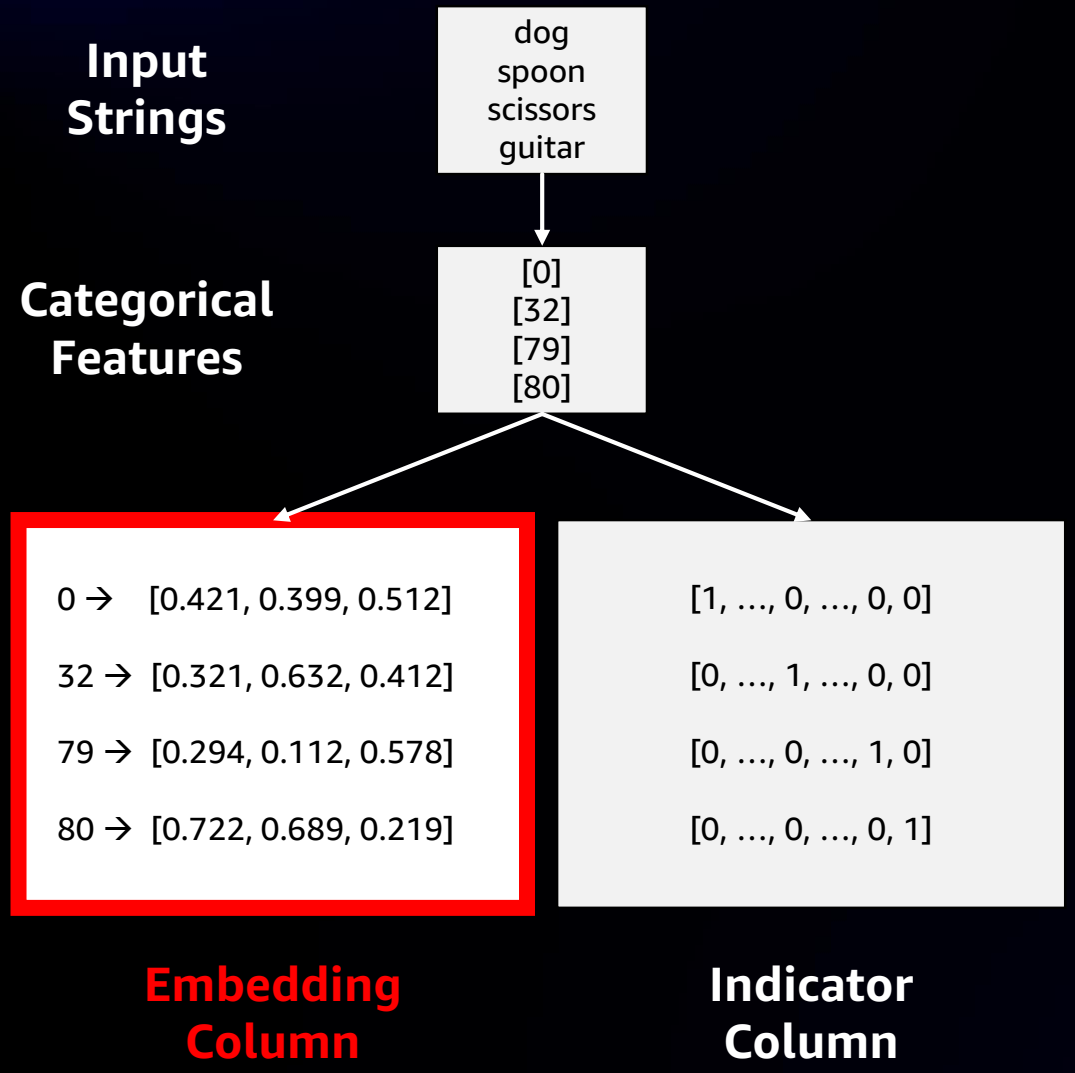
Our neural net includes an embedding layer

Neural network w/embedding layer



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What is an embedding?

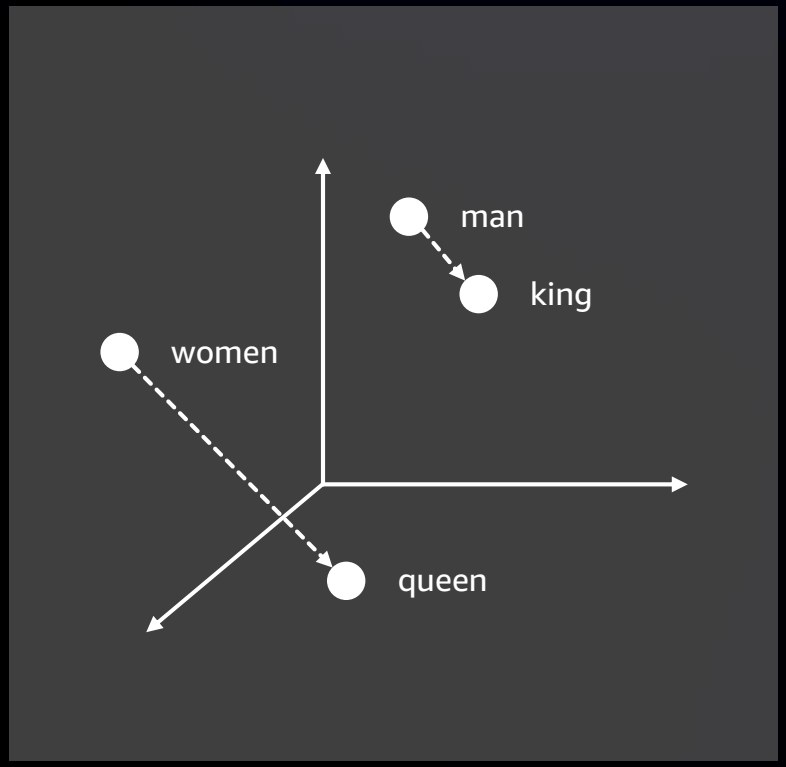


- Embeddings are a multi-dimensional representation of how the neural network sees each item in a category
- Easy-to-visualize groupings
- Trained through all the choices the driver makes and what the outcomes were
- Contains much more information useful to the model than just a 0 or 1 representation of which driver is being used
- Describes the driver numerically for the neural network

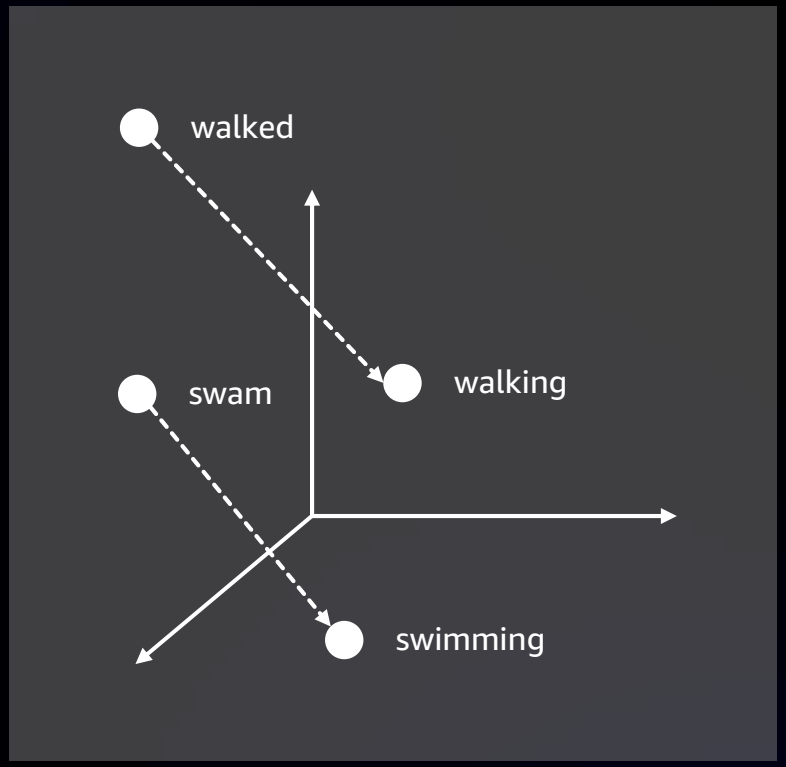
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Embeddings show similarities

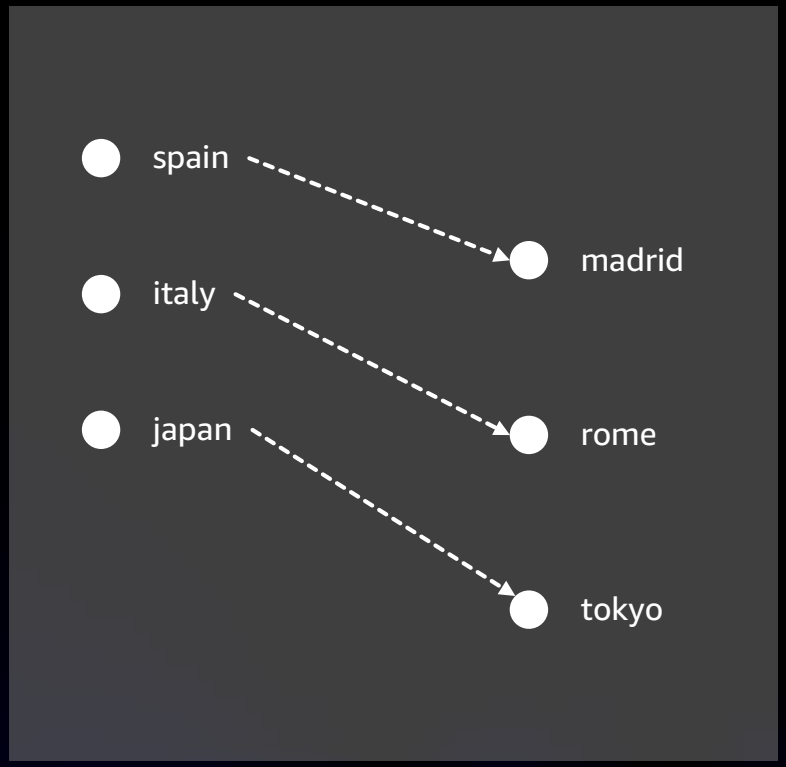
Male-Female



Verb Tense



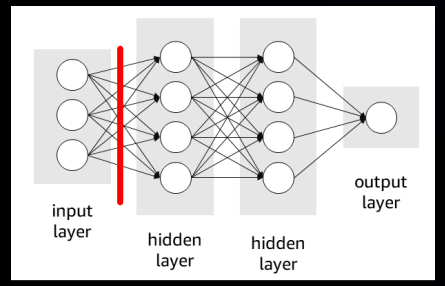
Country-Capital



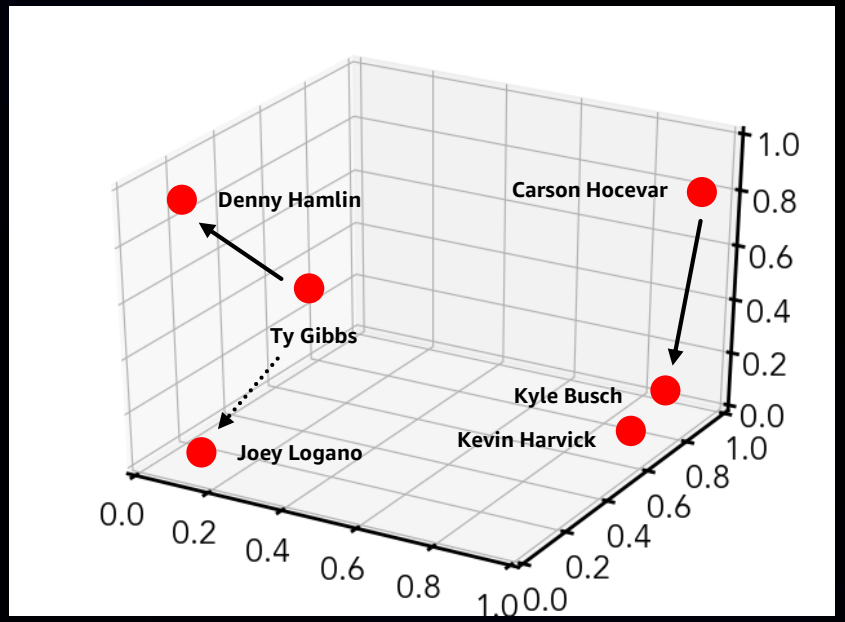
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

We can plot embeddings in 3D space

Embeddings



Visual representation



Decisions

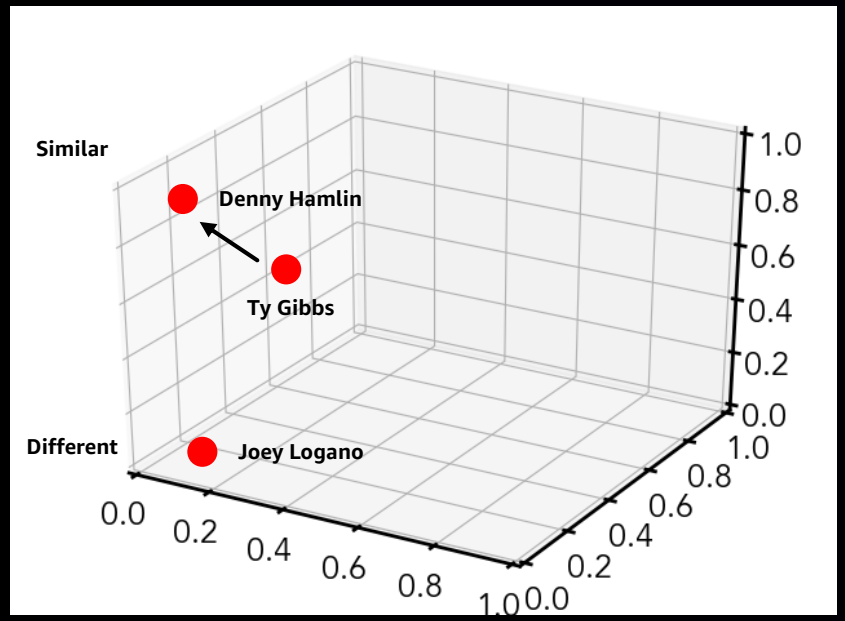


- The embeddings capture all of the qualitative things about a driver
- And we can plot them!
- We can put junior drivers and NASCAR drivers on the same chart
- Then we can use the chart to make decisions

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Let’s look at Ty Gibbs as an example

Embeddings chart



ARCA



Ty Gibbs

Most similar

NASCAR

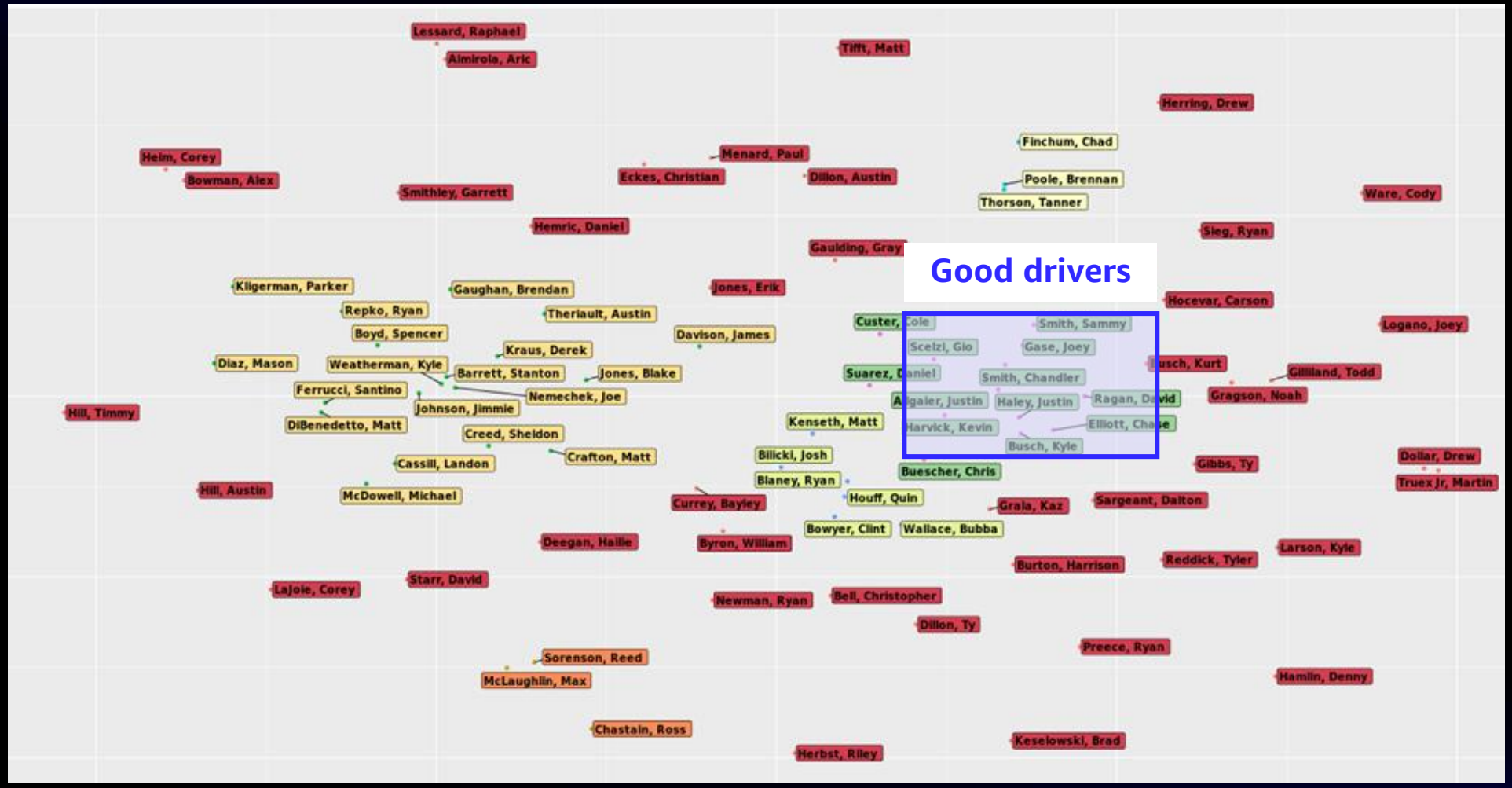


Denny Hamlin

- The embeddings show Ty Gibb’s tendencies
- Over time we learned that he drives like Denny Hamlin
- Now we know what tracks he will do well on

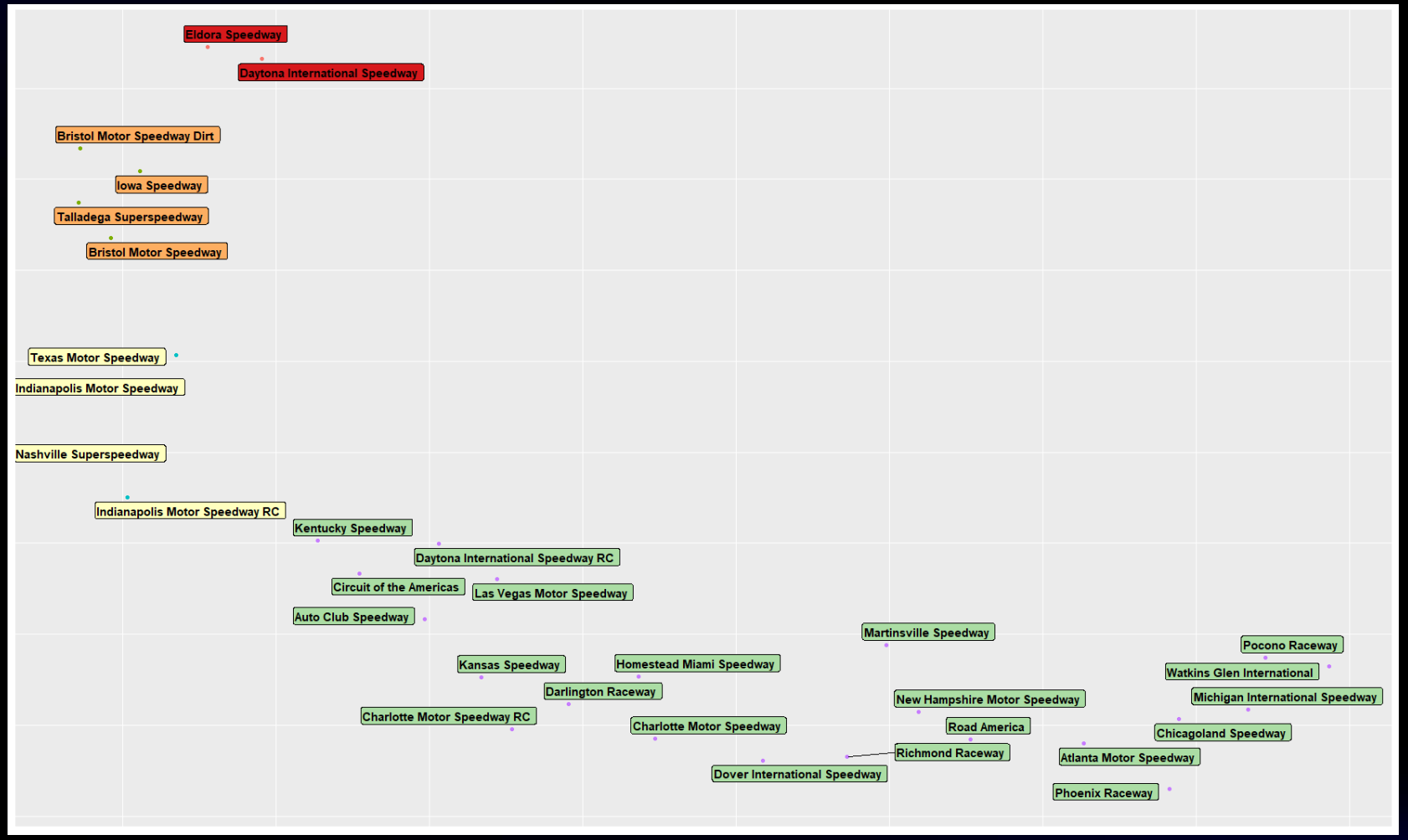
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Let’s look at all drivers



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

And tracks as well

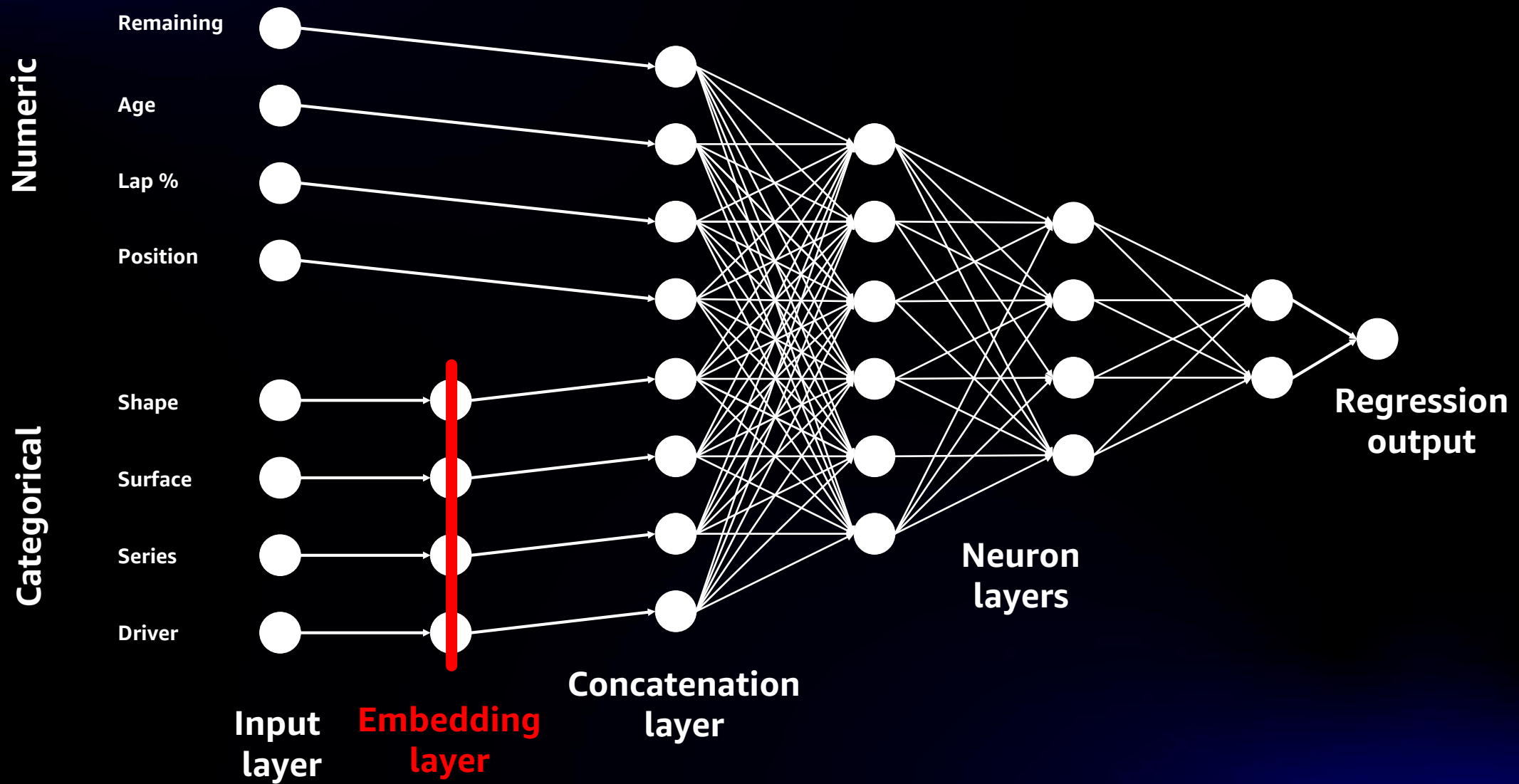


© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

So how do we model this in TensorFlow?

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Here’s what our model looks like



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

We can easily add new embedding layers

Embeddings logic

```
for categorical_var in categorical_columns:
    # Name of the categorical variable that will be used in the Keras Embedding layer
    cat_emb_name = categorical_var.replace(" ", "") + '_Embedding'

    # Define the embedding_size
    no_of_unique_cat = dat[categorical_var].nunique()
    embedding_size = int(min(np.ceil((no_of_unique_cat) ** .25), 10))

    # One Embedding Layer for each categorical variable
    input_model = Input(shape=(1,), name=categorical_var)
    output_model = Embedding(no_of_unique_cat, embedding_size, name=cat_emb_name)(input_model)
    output_model = Reshape(target_shape=(embedding_size,))(output_model)

    # Appending all the categorical inputs
    input_models.append(input_model)

    # Appending all the embeddings
    output_embeddings.append(output_model)

for i in other_cols:
    input_numeric = Input(shape=(1,), name=i)
    embedding_numeric = Dense(1)(input_numeric)
    input_models.append(input_numeric)
    output_embeddings.append(embedding_numeric)
```

Designed to expand over time, all you need is a list of the categorical and numerical columns

Able to tune the embedding size as a hyper parameter in the model

One embedding layer for each categorical variable . . . and numerical variable (further down)

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Our code is very adaptable

Pre-processing logic

```
def preproc(X_train, X_test):
    input_list_train = []
    input_list_test = []

    # the cols to be embedded: rescaling to range [0, # values)
    for c in categorical_columns:
        input_list_train.append(X_train[c].values)
        input_list_test.append(X_test[c].fillna(0).values)

    print(other_cols)
    for i in other_cols:
        input_list_train.append(X_train[i].values)
        input_list_test.append(X_test[i].values)

    return input_list_train, input_list_test
```

Keep pre-processing logic simple

Weights

```
embed_df = pd.DataFrame(model.get_weights()[0])
embed_df['driver_key'] = dat_driver_key
```

Extract weights from the model and then apply dimensionality reduction

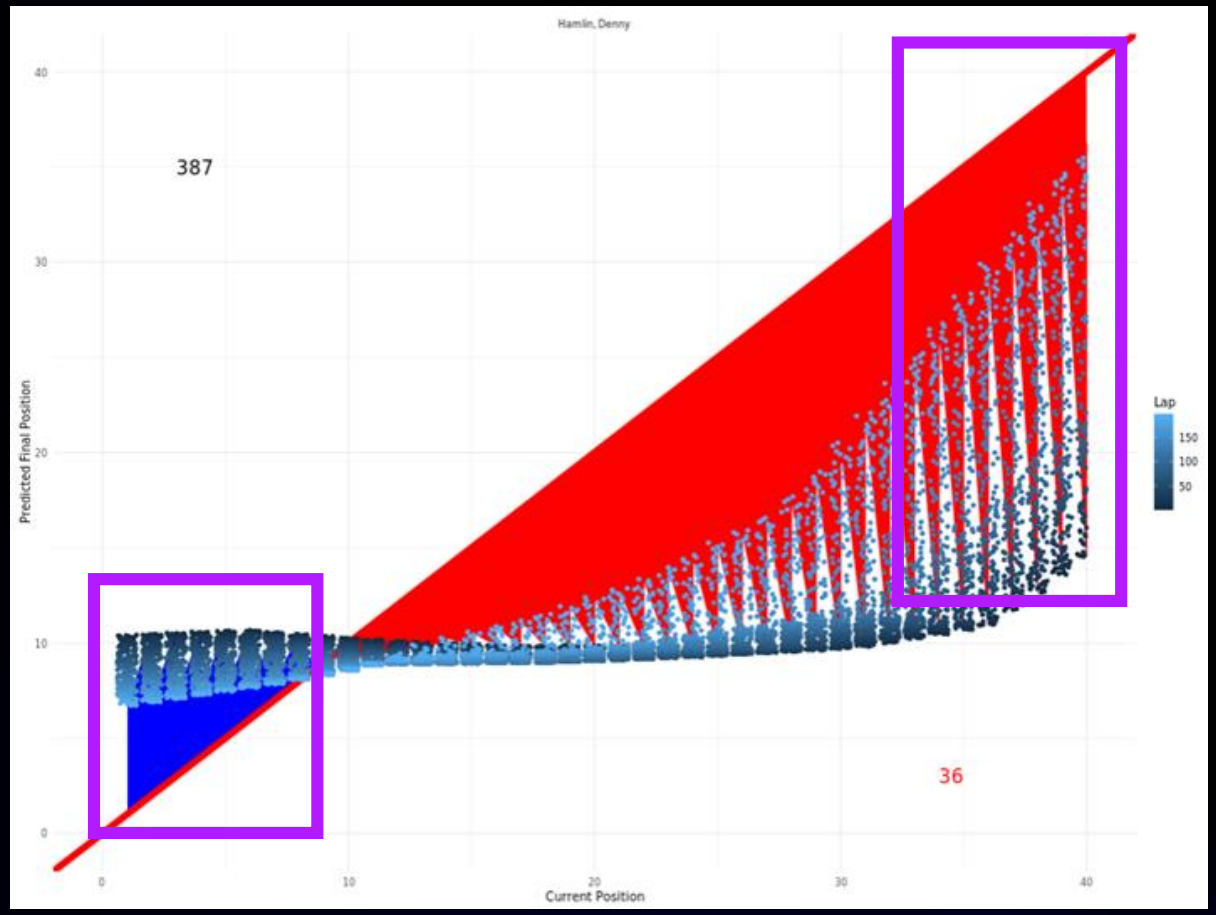
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Here’s what the end result looks like

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

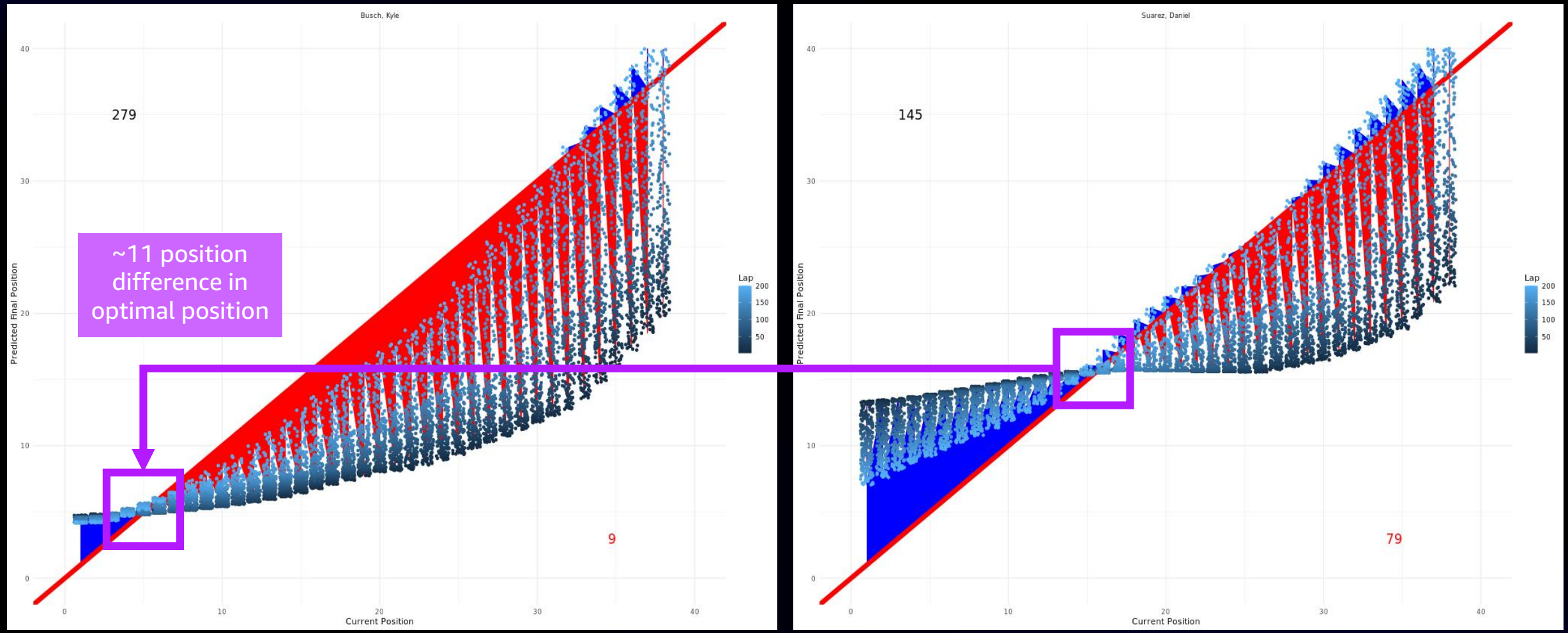
Driver competitiveness

- Where the lines cross is essentially the limit of the driver's competitiveness
- When black is highest, that means the more laps that are available, the more likely the driver will perform worse
- A driver at position 40 is expected to be able to make up ground versus lesser drivers
- A driver at position 1 is at risk of losing ground to stronger drivers



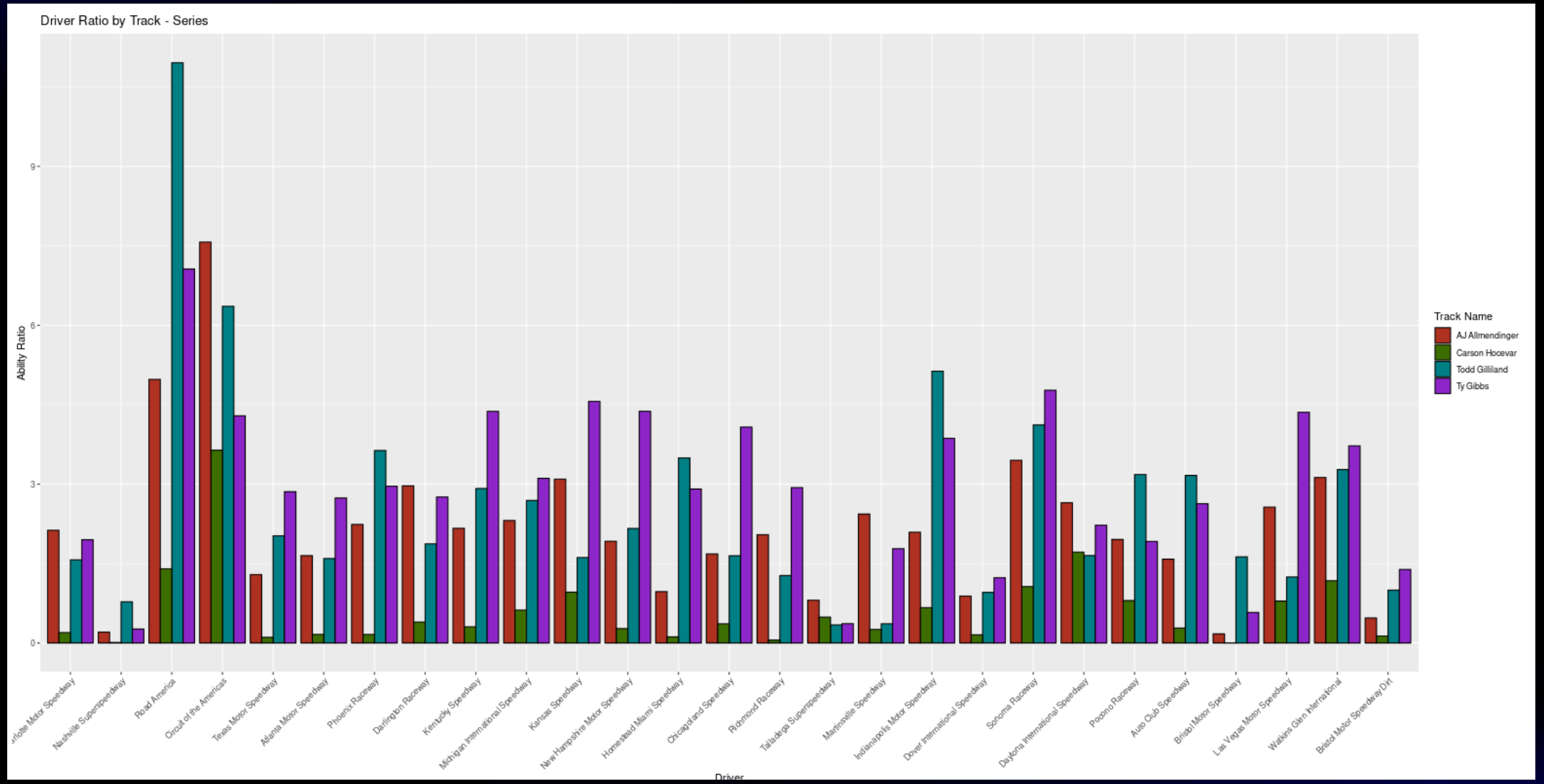
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Compare final position by current lap/pos



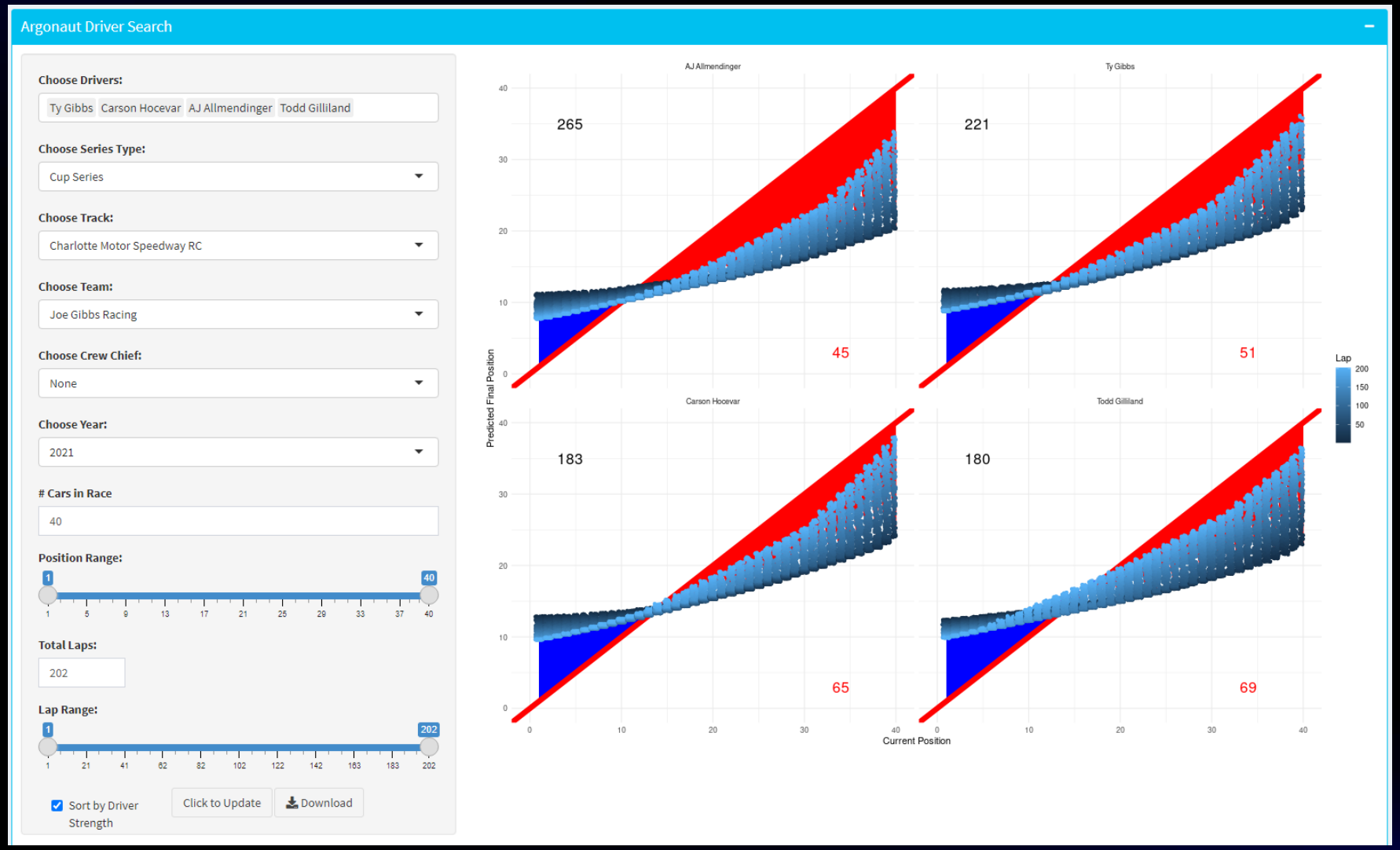
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Compare driver ability by track



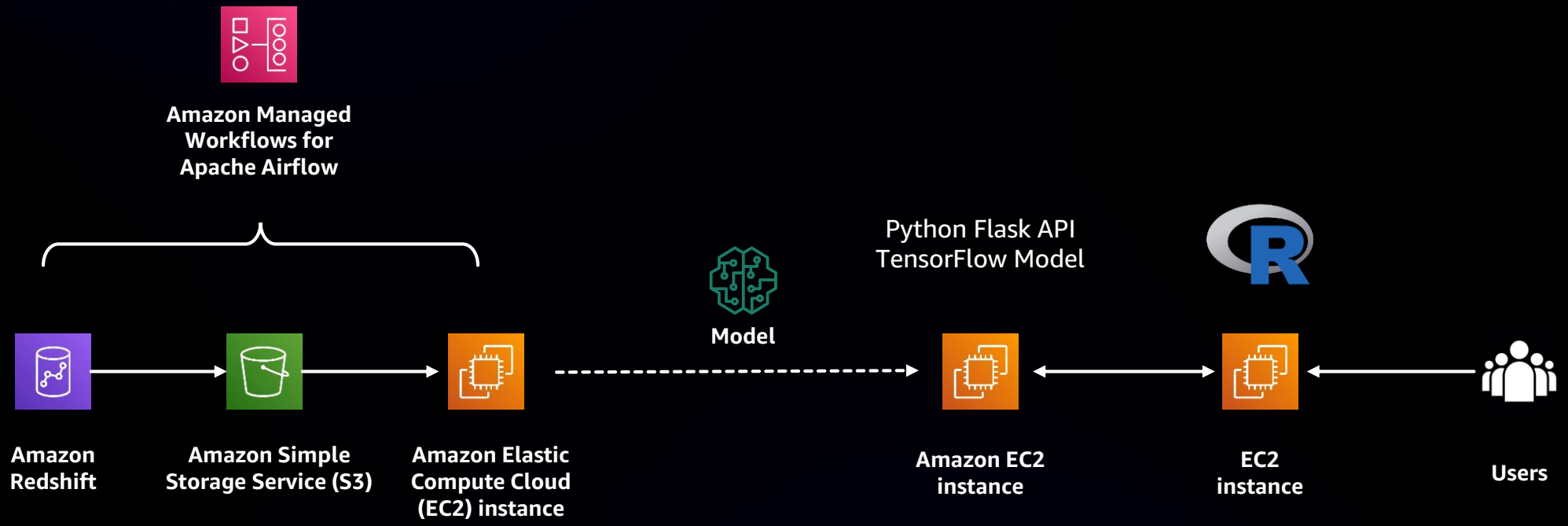
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Dashboard



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Finally, here's how we're running this on AWS



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Steve Miller
ssmiller@amazon.com

Derek Jones
derek.v.jones@toyota.com

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.