Mark
20 seconds

Hi everyone, and welcome to our presentation on Predicting Turnover through Machine Learning.
Our project team - Jordan, Ash, myself, Aysun, and Ryan, will walk you through our project, which is a tool for HR professionals to get predictive turnover insights on their workforce.

The Problem:
Unexpected Employee Turnover

Mark
35 seconds.

Our project looks at the problem of unexpected employee turnover, referred to in the industry as “regrettable” turnover. This is when an employee resigns from their job, surprising their company with a vacant position. This is incredibly costly for businesses - a 2019 Gallup study estimated US businesses lost $1 TRILLION to regrettable turnover that year, and the Society for Human Resource Management estimates that on average, the cost of replacing an employee is 6-9 months of their salary.

The Problem:
HR Teams Need Analytical Support

Mark
25 seconds.

HR departments are often tasked with analyzing turnover trends and producing insights and forecasting for executives. While they typically have access to basic reporting such as lists of terminated employees or historical turnover averages, they often lack the resources or expertise to conduct predictive analysis. Thus, their ability to think about talent retention strategically suffers.

Mark
65 seconds

Globally, the market for HR analytics tools was about $2.5 BILLION in 2022 and expected to reach over $6 BILLION by 2030.

Our target customers are medium sized companies—those big enough to have a business need for predictive turnover but not so big they have in-house resources to conduct this type of work—and according to Crunchbase, this gives us a huge market to pursue.

Through our market research and discussions with industry experts, our team discerned a path to stand out in this crowded market by prioritizing simplicity and actionable analysis. While most competing products require costly subscriptions and integrations, and overwhelm users with too many charts and not enough insight, we distinguish ourselves by offering a significantly lighter weight (and free!) tool that makes understanding predictive turnover easy.

URL: https://apps-summer.ischool.berkeley.edu/EmployeeTurnoverPredictionsApp/

Ryan
2 minutes 20 Seconds

< Zoom 200%>
<Transitioning from Slide 4 to the Home Page>
In light of this, we have engineered an interactive dashboard application aptly titled "Predicting Turnover Through Machine Learning." Let’s walk through a demonstration of the application's analytics and predictions functionality.

<Go to About Page, scrolling down slowly>
New users should start on the About page, which shows an overview of the site's functionality, instructions for uploading data, and a detailed description of the dashboard and what insights it can provide.

<Go to fresh Document Upload page.>
To use the application, we will need to upload a file of employee data. After reading and agreeing to the Terms of Service, you can click Browse and upload the data file.

<Switch to Dashboard page>
After uploading the file, the dashboard will be populated.

<Turnover Rate by Month graph>

The first module shows the monthly turnover rates from the submitted data, which provides an overall historical view of turnover rates over time. (Zoom in 2020-2022) For example, we can view trends in turnover during the COVID pandemic, including the peak turnover in October 2021 during the “Great Resignation”.

<Forecasted Turnover Rates for Remaining Months>

The next module uses Meta's Prophet tool to analyze past turnover rates in order to forecast potential turnover scenarios in the future. The historical data- shown as the blue line- fits *fairly* well within the purple bands that are our model’s predictions. Thus we can have confidence in the model’s future predictions, shown to the right of the dotted line. This forecast will be valuable for talent planning and budgeting in the future.

<Turnover Rate Comparisons>

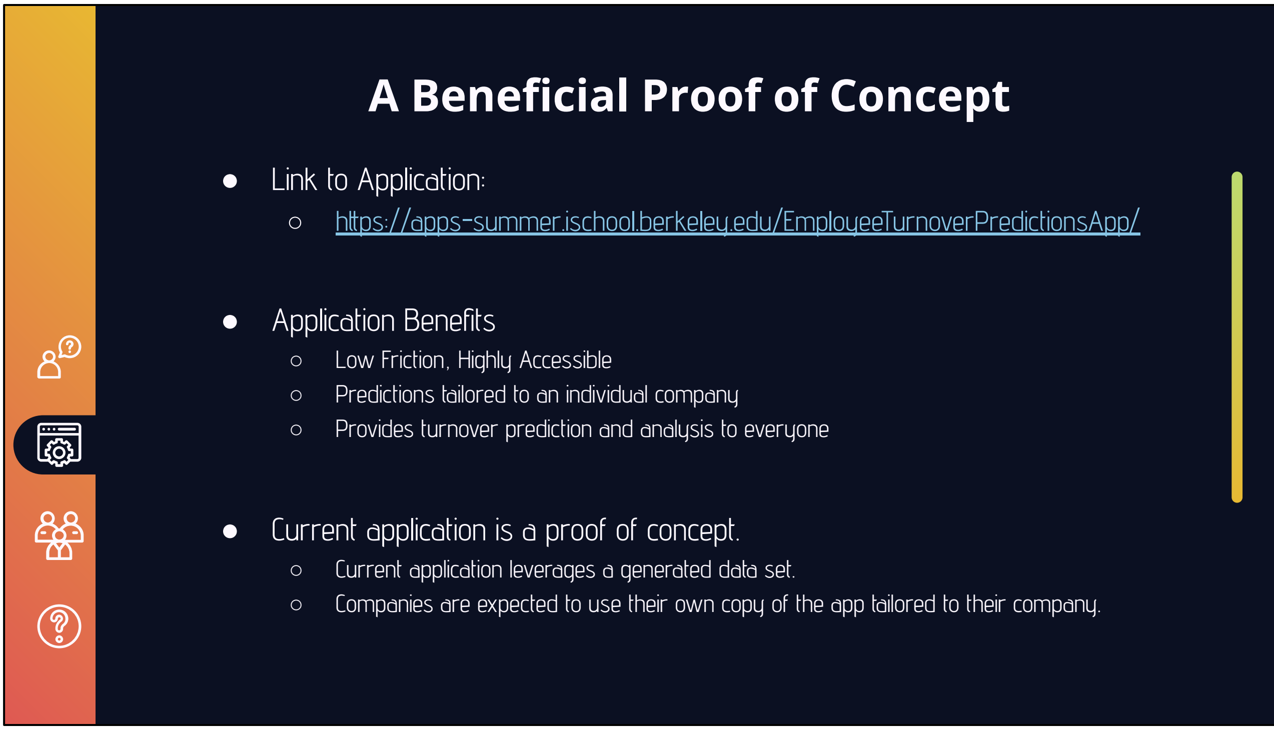
The next module uses chi-squared tests in order to determine whether sub-groups within each feature have significantly different turnover rates. As shown here, the turnover rates between any of the sub-groups are not significantly different from each other.

<Feature Importance>

The feature importance table shows how each feature is associated with turnover. As can be seen, tenure is significantly associated with low turnover, indicating that long-serving employees are unlikely to quit.

<Predictions Table>

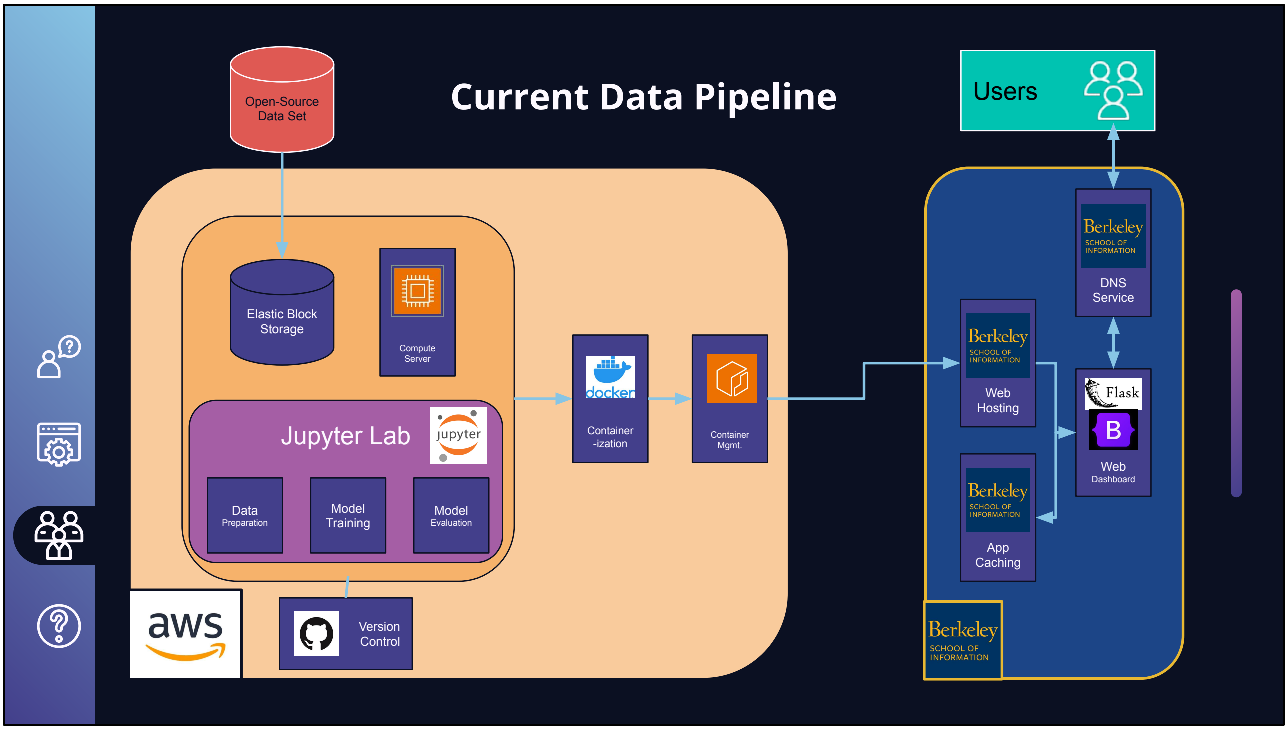
And lastly, we have our main predictions table. This is how we can see which current employees are predicted to stay (as marked by a 0 prediction), and who is predicted to turnover (as marked by a 1 prediction).



Ryan

30 Seconds

Our application provides many valuable insights and benefits to HR departments, and they could start using it right away with their own data to craft their own employee retention strategies. But we consider this application to be a proof of concept, an example of how the app can be used. For the best results, companies would need to use a copy of this application tailored to their particular company.



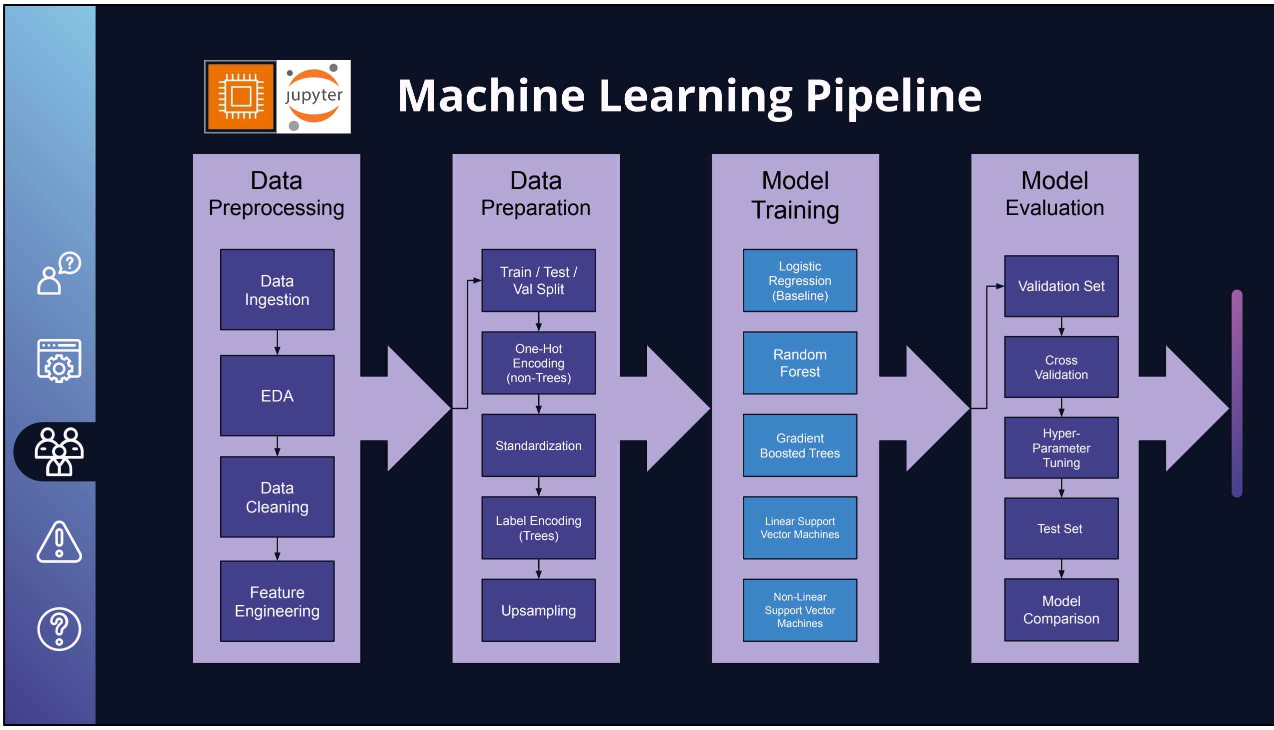
Jordan
35 seconds

The data pipeline that makes this application possible is pictured here.

We utilized an Amazon Web Services EC2 instance for processing the data and creating our models, and Docker to containerize the application. Afterwards, we then transfer that application to an iSchool server for hosting and deployment.

We designed our application to be simple in order to promote an aggressive development strategy.

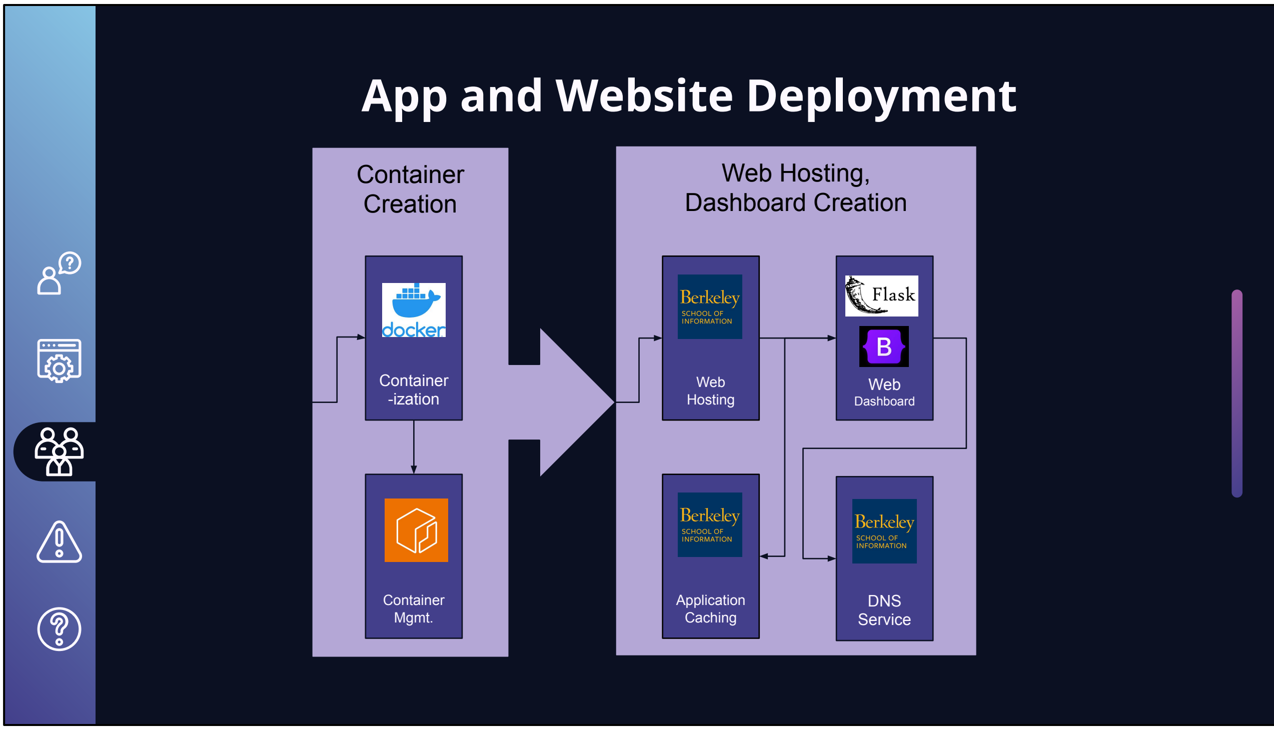
We consolidated our platforms, used familiar tools, and kept things lean in order to build as fast as possible.



Jordan
15 seconds

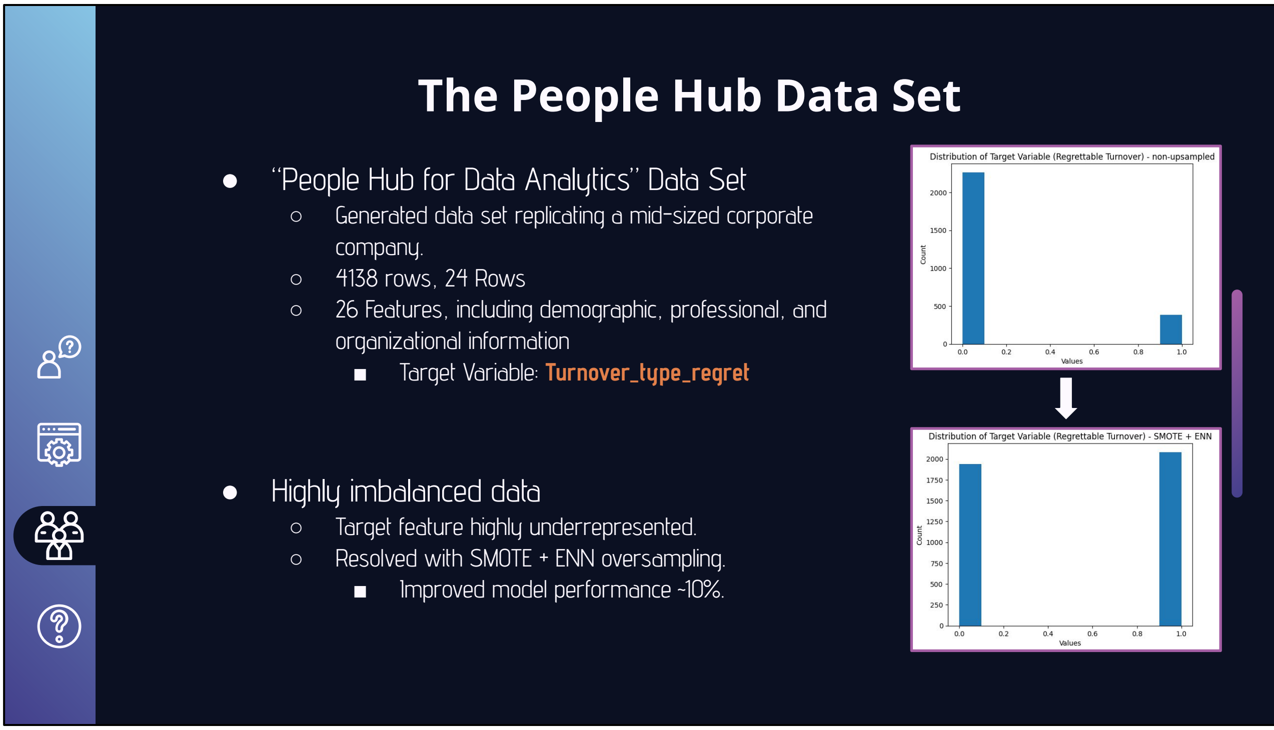
Here is a general overview of the steps within our machine learning pipeline.

We first process and prepare the data, before using it to train and evaluate our machine learning models.



Jordan
10 seconds

Once the model is selected, it gets added to the app, containerized with Docker, and then sent to the server for hosting and deployment.



Jordan
40 Seconds

With that in mind, let's dive deeper into our pipeline's components and considerations.

Our project's first challenge was finding a data set.

We couldn't use real employee data, BUT

- we did want data that seemed realistic to a business,
- had the features that our users would be interested in,
- and had enough rows to generate predictions with.

We found these qualities in the People Hub for Data Analytics data set, which is an open-source data set that was generated specifically to emulate a company's employee data.

It was highly imbalanced, but we handled that by applying Synthetic Minority Oversampling Technique and Edited Nearest Neighbor to oversample the data without generating too much noise.

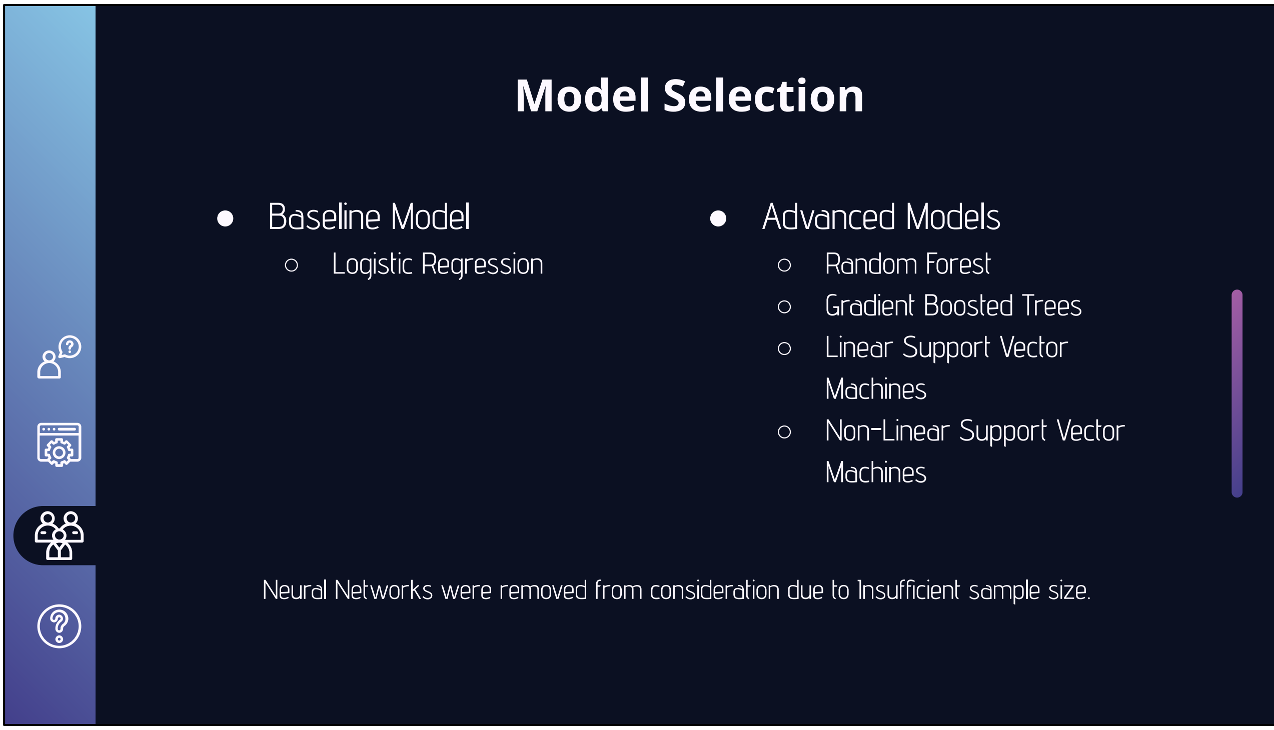
Feature Engineering

- "**Turnover_type_regret**": regrettable turnover indicator feature.
  - Determined by stated termination reason.
  - Our primary target variable.
- "**Salary_adj**": salary value adjusted for current level of inflation.
- "**Salary_adj_cohort_percentile**": the percentile ranking of salary that a person makes within their cohort.
  - Cohort: the same department, job level, and city
  - 30% indicates the employee is at the 30th percentile of their cohort's salary range
- Reduced 26 original features down to 16 used in analysis.
  - Feature selection was informed by EDA (i.e. features that have higher correlation with target) and SME inputs
  - Feature list given in appendix

Jordan
20 Seconds

Next, we generated new features and we're listing three of them that we were interested in here.

Furthermore, we reduced the original 26 features down to 16 used in our analysis based on findings in our EDA and subject matter expert insight.



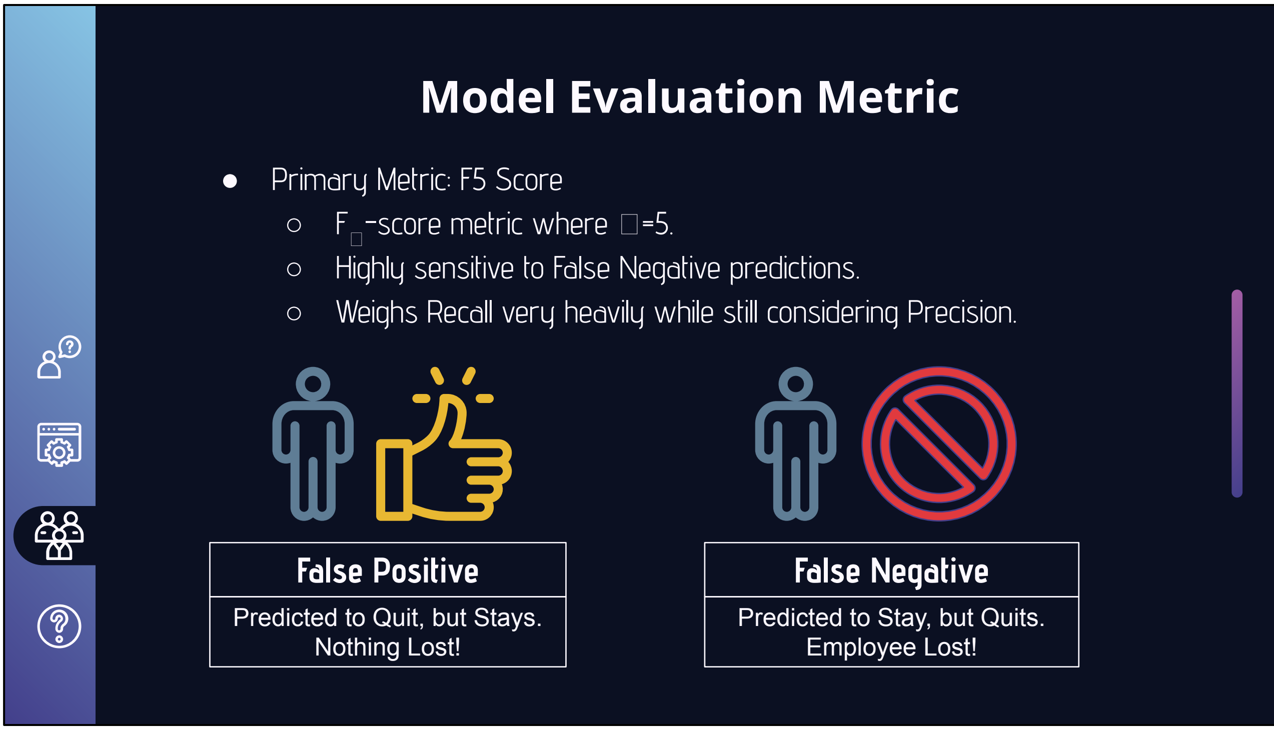
Jordan
25 seconds

When selecting our models, we wanted one basic model to act as our baseline, plus a selection of more sophisticated models that were known to perform well in classification tasks.

As such, we chose the basic Logistic Regression model as our baseline, and we chose the four models on the right as our advanced models.

We had originally considered including a Neutral Network model as well.

But that was dropped due to concerns about our data set's sample size.

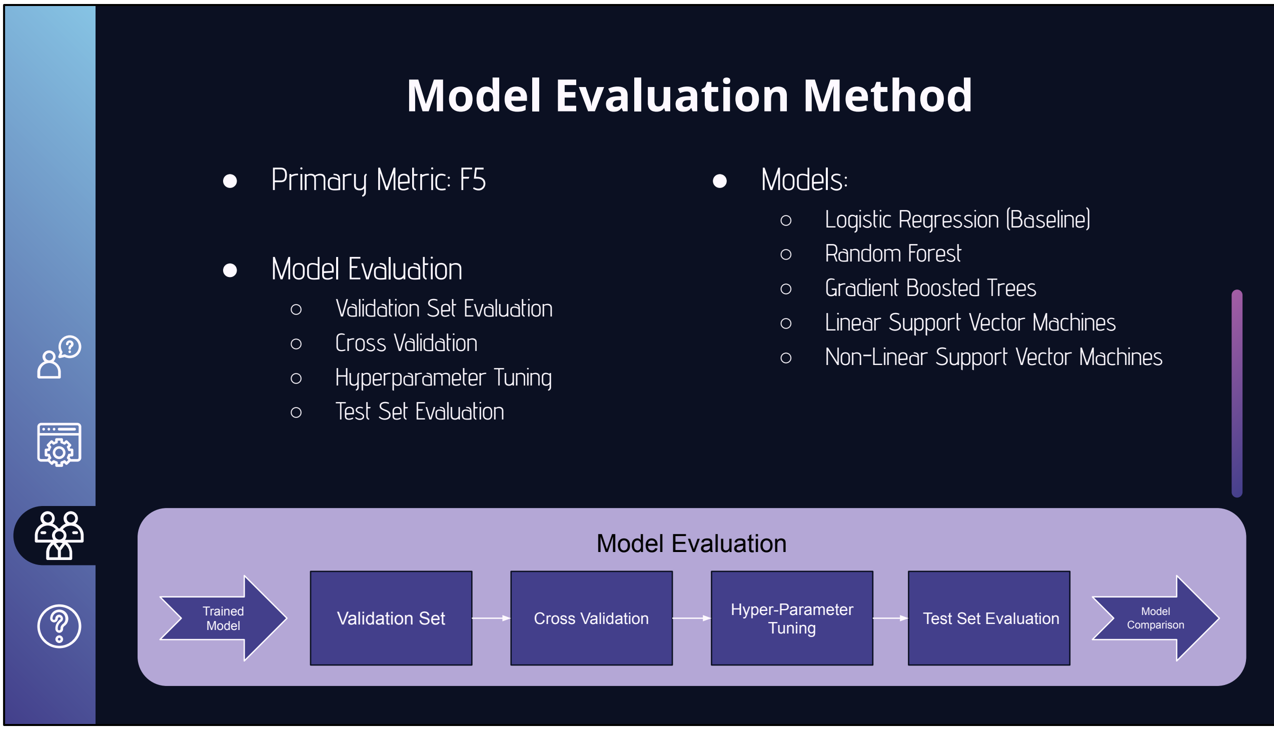


Aysun

35 seconds

When choosing a metric to use for measuring model performance, we decided to use an F5 score as our primary metric. This metric weighs False Negative predictions heavily while still considering the number of False Positives predictions.

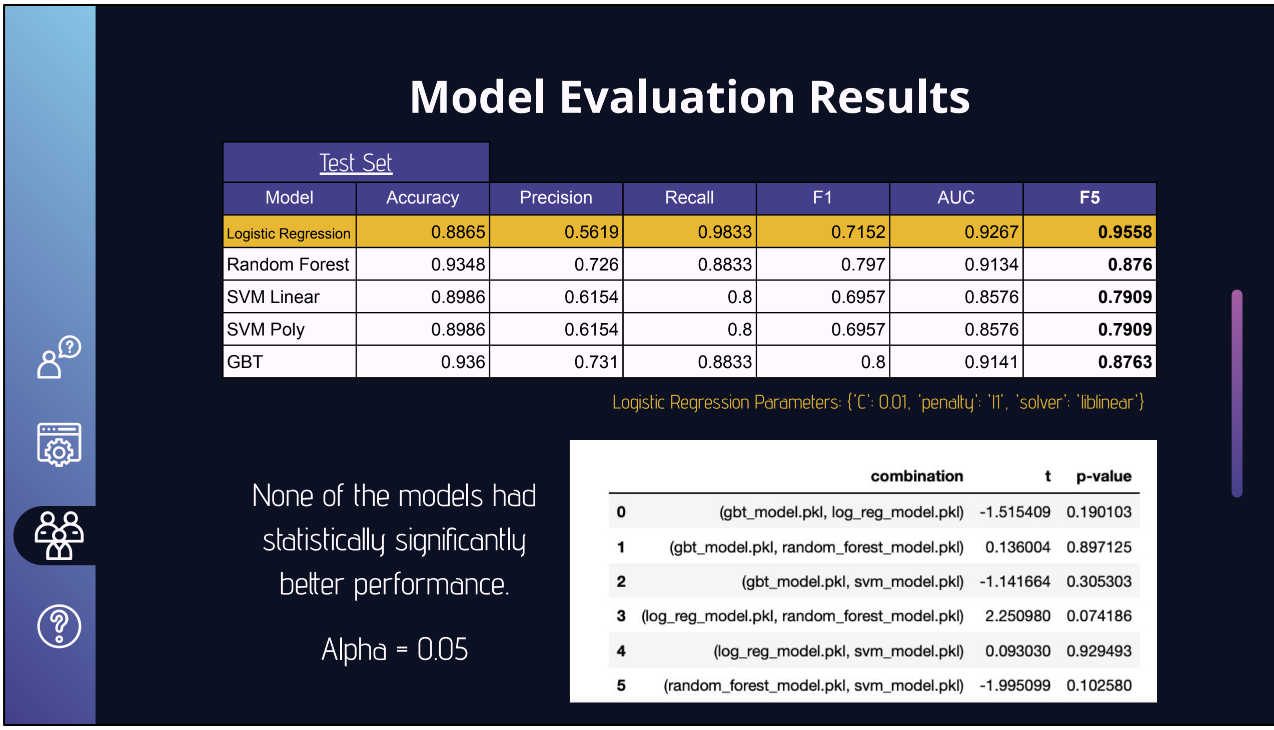
This is important when you consider our application's business case. It is acceptable to predict that a few employees will turnover but don't. However, it is not acceptable to predict an employee would stay but actually leaves. We felt like F5 balanced both outcomes evenly for the business needs.



Aysun

15 seconds

With the models and the primary metric chosen, we conducted model evaluation leveraging validation, cross-validation to avoid overfitting (to promote generalization), hyper parameter tuning and test set evaluation. Once a model has been fully trained, optimized, and evaluated, we then did the same steps for the other models. After all of that, we moved to model comparison stage.



Aysun

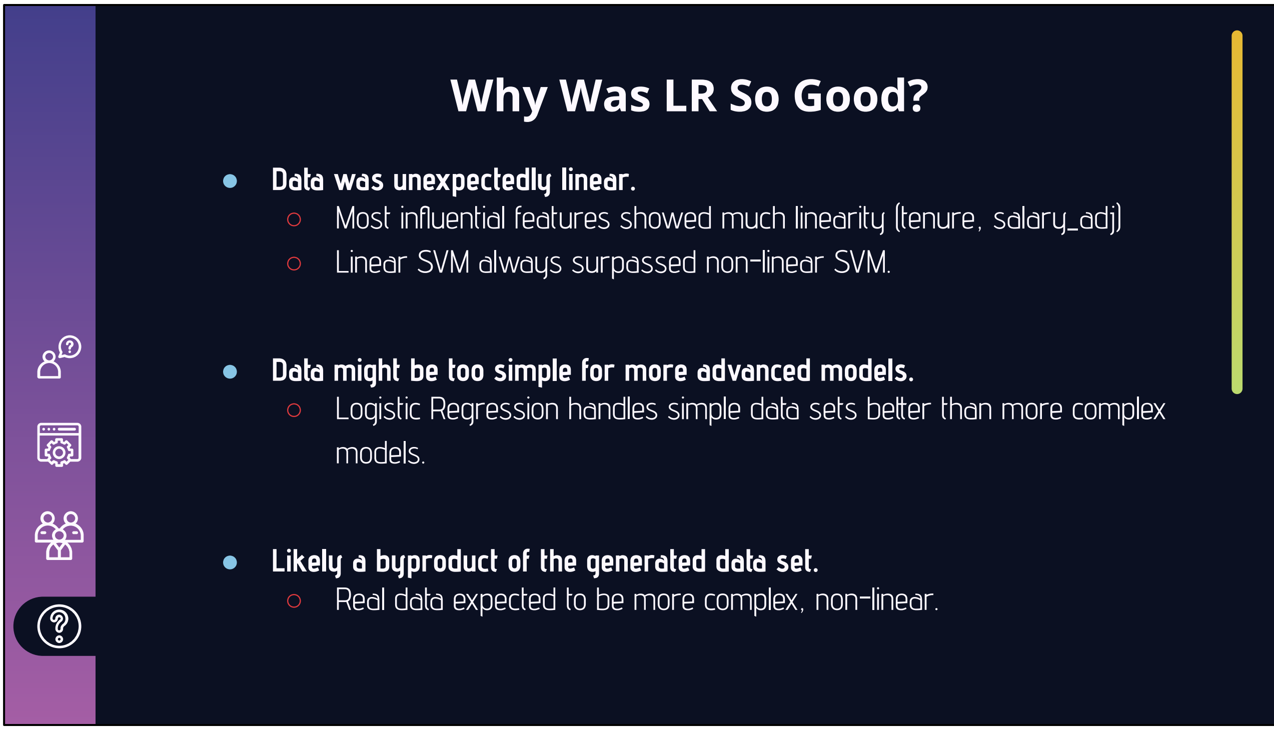
25 seconds.

And here are the model evaluation results. Unexpectedly, Logistic Regression had the highest F5 score when we eyeball the scores. But, we went one step further and implemented a 5x2 cross validation modified t-test to compare model performance statistically, we discovered that none of the models were different from each other to a statistically significant degree.

Aysun

15 seconds.

In light of that, we decided to pick Logistic Regression as our model of choice for the application. It had the best F5 score of all the models when we eyeball them, but not statistically significant from other models. So, we chose the simplest model for ease of application and explainability.



Aysun

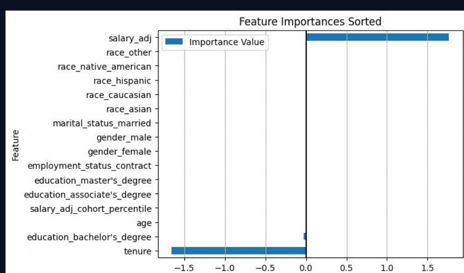
30 seconds.

Logistic Regression's performance was unexpectedly good, and we have a few ideas why. Our data was unexpectedly linear, with the tenure and adjusted salary features having outsized influence on employee turnover. Our data might also be too simple for the more sophisticated models, but just right for basic Logistic Regression. These considerations might be consequences of the generated data structure.

Feature Importances

- Feature Importances from Logistic Regression Feature Weights (Coefficients)

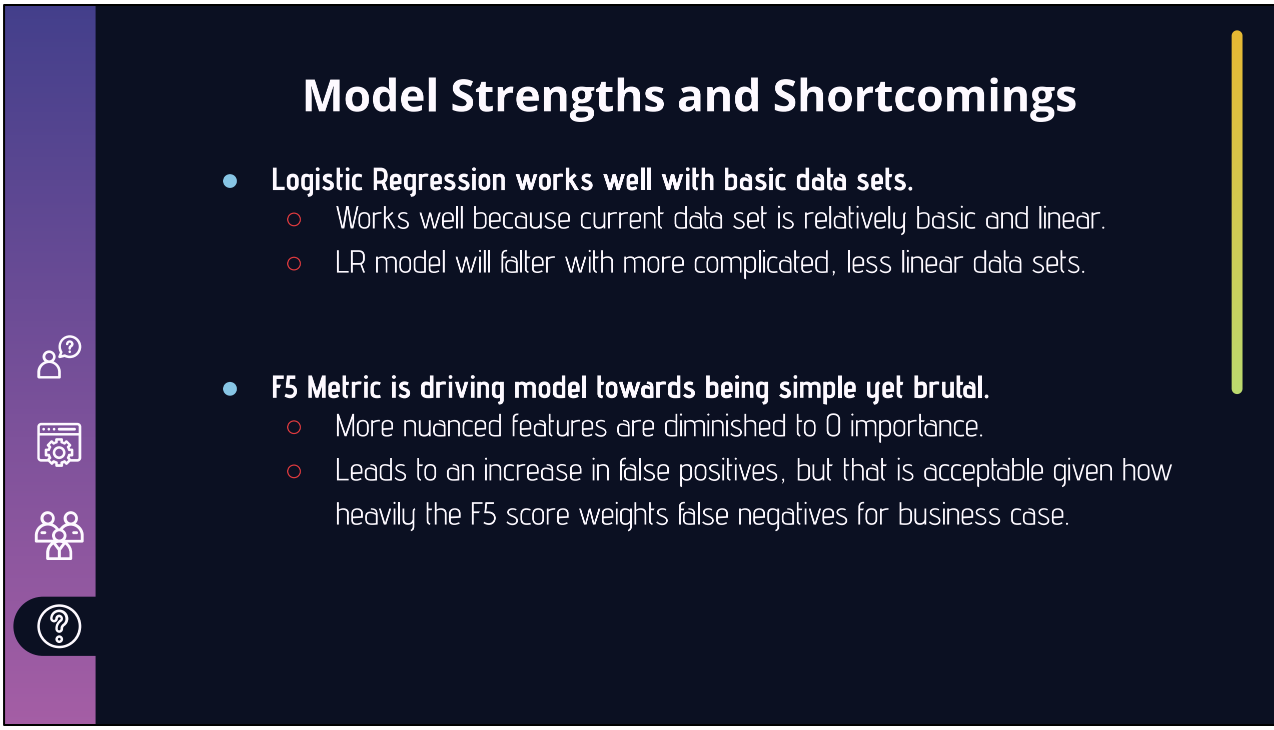
- Most influential feature importances:
  - Most Negative: Tenure
  - Most Positive: Salary_adj



Variable of Interest	Importance Value	p-value	Significant Result?
tenure	-1.7	9e-57	Yes
education_bachelor's_degree	-0.028	0.011	Yes
age	0.00	0.059	No
salary_adj_cohort_percentile	0.00	0.00028	Yes
education_associate's_degree	0.00	0.00018	Yes
education_master's_degree	0.00	0.56	No
employment_status_contract	0.00	0.22	No
gender_female	0.00	0.68	No
gender_male	0.00	0.29	No
marital_status_married	0.00	0.003	Yes
race_asian	0.00	7.5e-10	Yes
race_caucasian	0.00	4.1e-09	Yes
race_hispanic	0.00	5.4e-06	Yes
race_native_american	0.00	0.11	No
race_other	0.00	0.05	No
salary_adj	1.8	3.8e-08	Yes

Ash
35 Seconds

Feature importance was very important for our users to know, and we calculated it using the model's feature coefficient weights. In our logistic regression model, Tenure and Adjusted Salary demonstrated the largest impact on predicting turnover. As it turns out, people who have been at a company for a long time are less likely to quit; perhaps they are loyal to the company. At the same time, those with a high salary are more likely to quit; these individuals might be top performers that are sought out by recruiters.



Ash
25 seconds.

Our Logistic Regression model worked well with our current dataset because it is simple and linear. But we recognize that it might not generalize well with other, more complex data sets like those in real life.

Furthermore, the model's interaction with the F5 metric lead to a large increase in false positive rates which is actually considered acceptable given our business case.

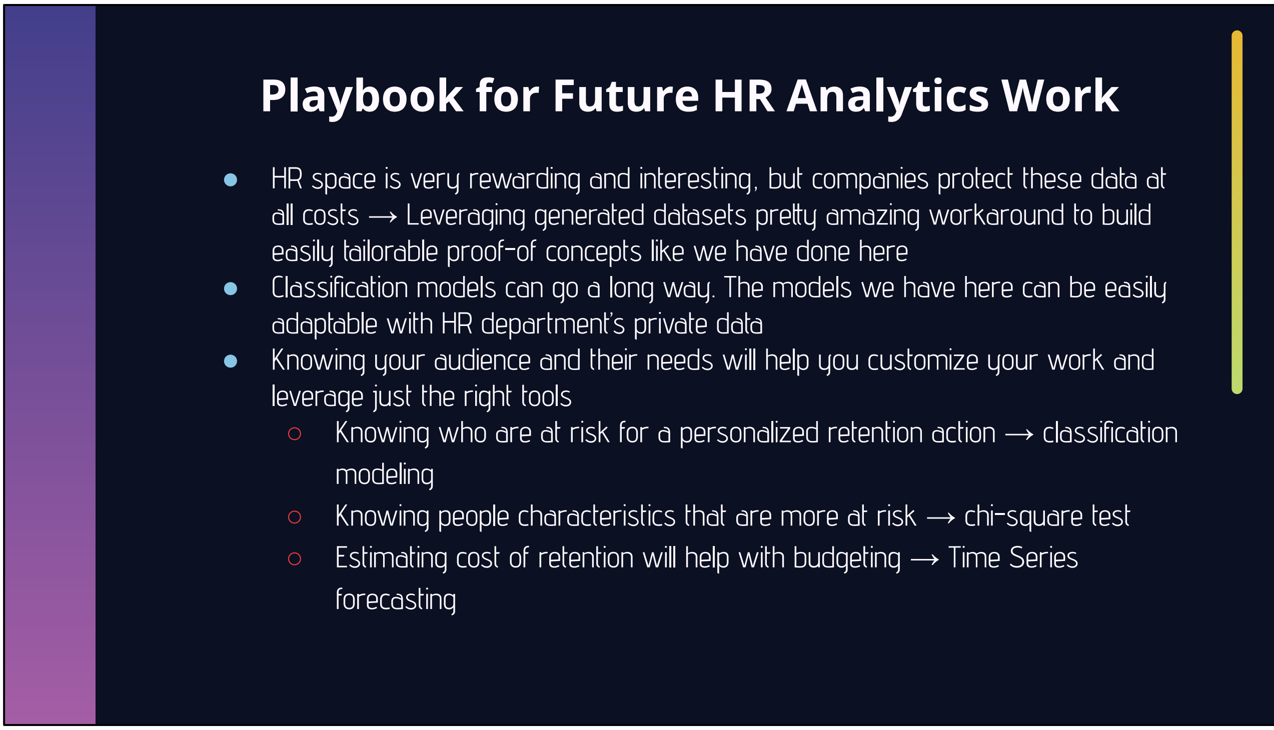
Ash
40 seconds

For those interested in following in our footsteps, here are the most important technical lessons we learned:

Because employee data sets may vary wildly, we built a pipeline that is flexible and adaptive with its data, features, and models.

Applying statistics to our pipeline was a major boon for us. The additional analysis provided valuable insight regarding how our users should apply their employee retention efforts more effectively.

And lastly, we learned how to build an application with a very intentional and mindful design in order to meet the unique needs of non-technical HR users.



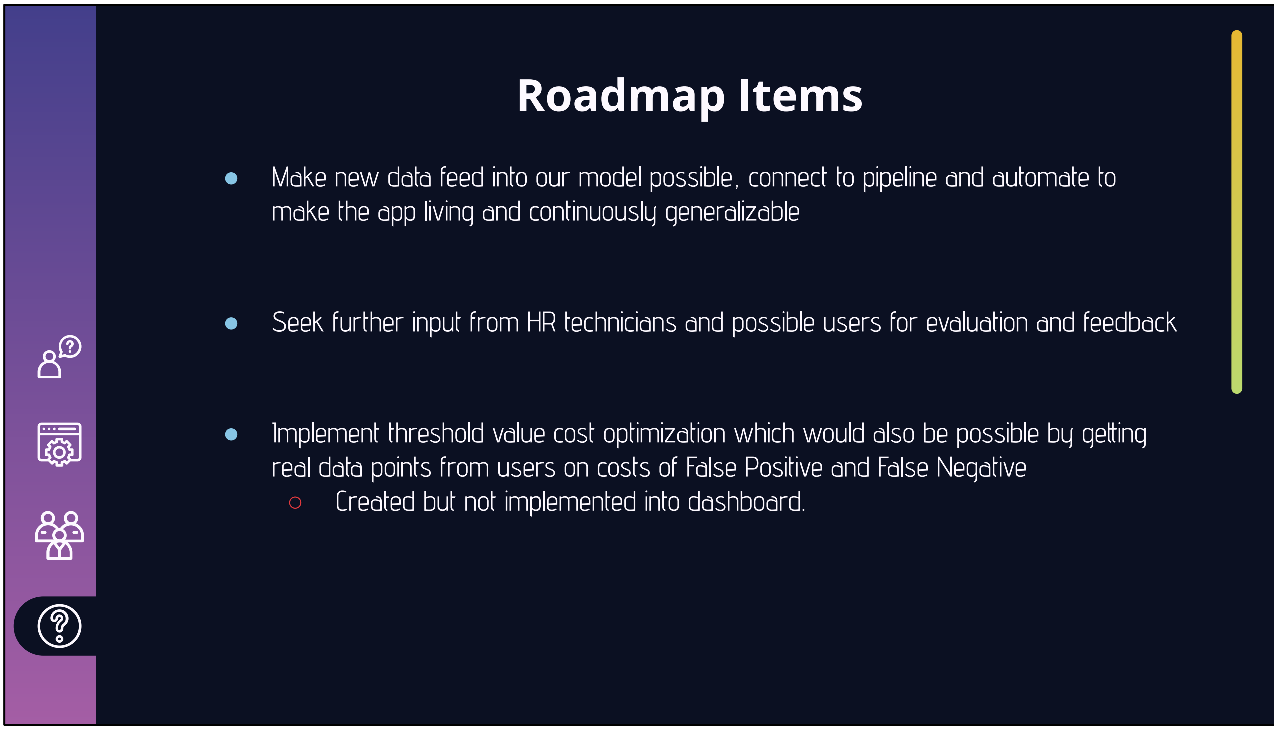
Ash
50 Seconds

Beyond our technical takeaways, we also wanted to share our learnings as a playbook for anyone else interested in working in the challenging yet rewarding field of HR Analytics.

Firstly, getting data can be a challenge, as companies will protect their employee information closely. As such, leveraging generated datasets can be a good workaround when building applications like ours.

Secondly, the classification models can go a long way, as they are easily applicable to an HR department's data.

And lastly, it is essential to deeply understand both the business case and the available analytics tools. Whether you're finding at-risk employees using classification modelling or estimating headcount budgeting with time series forecasting, you must synergize the essential HR need with the best analytics tool available.

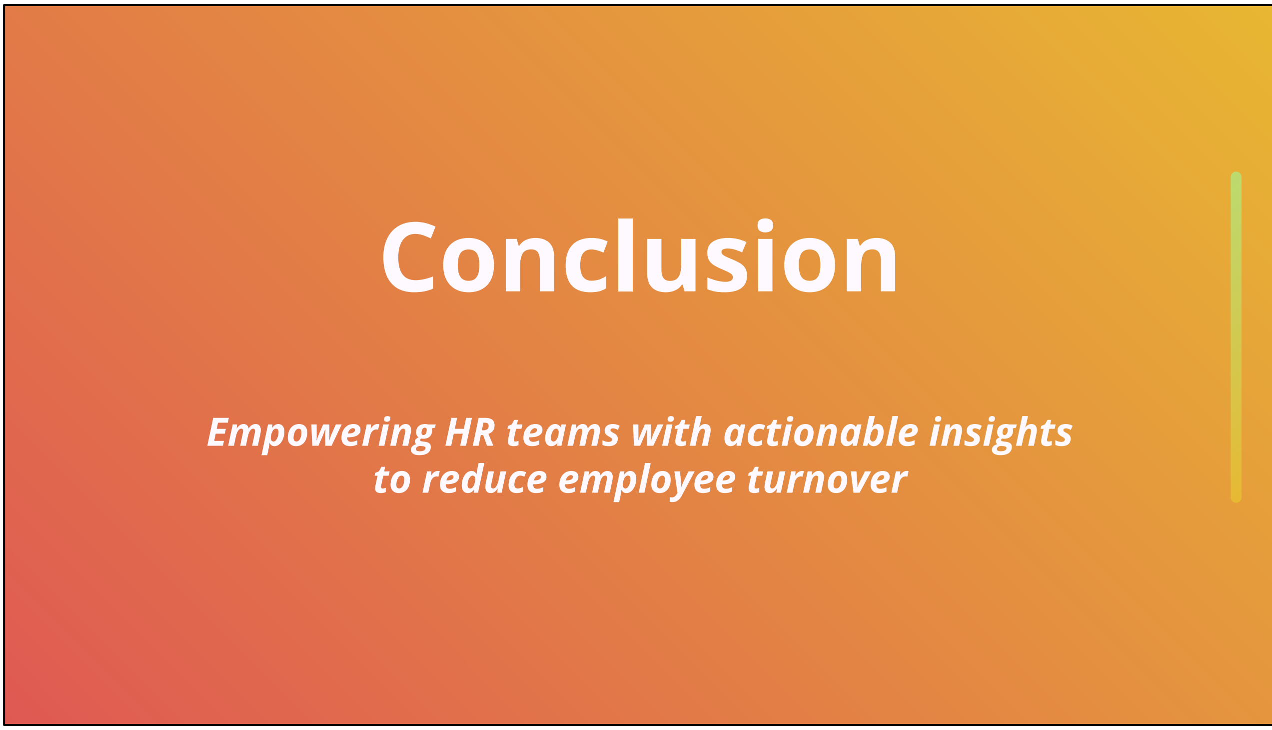


Ash
30 seconds

This proof of concept application could be just the start. If we had more time, we would expand the functionality of our application with these additions:

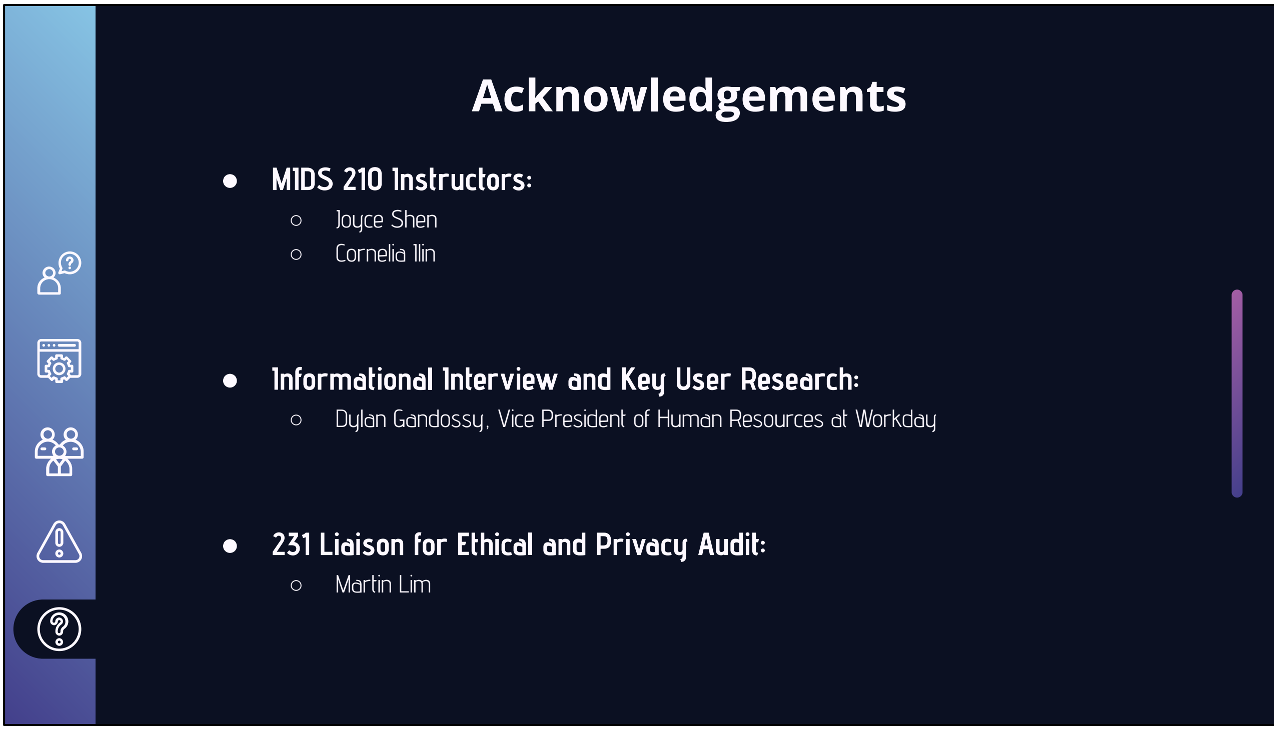
We would make our pipeline truly end-to-end, thus making the app automated, living, and continuously generalizable.

And, we would seek to improve the app with feedback from our target audience of HR technicians, and implement additional engineered features.



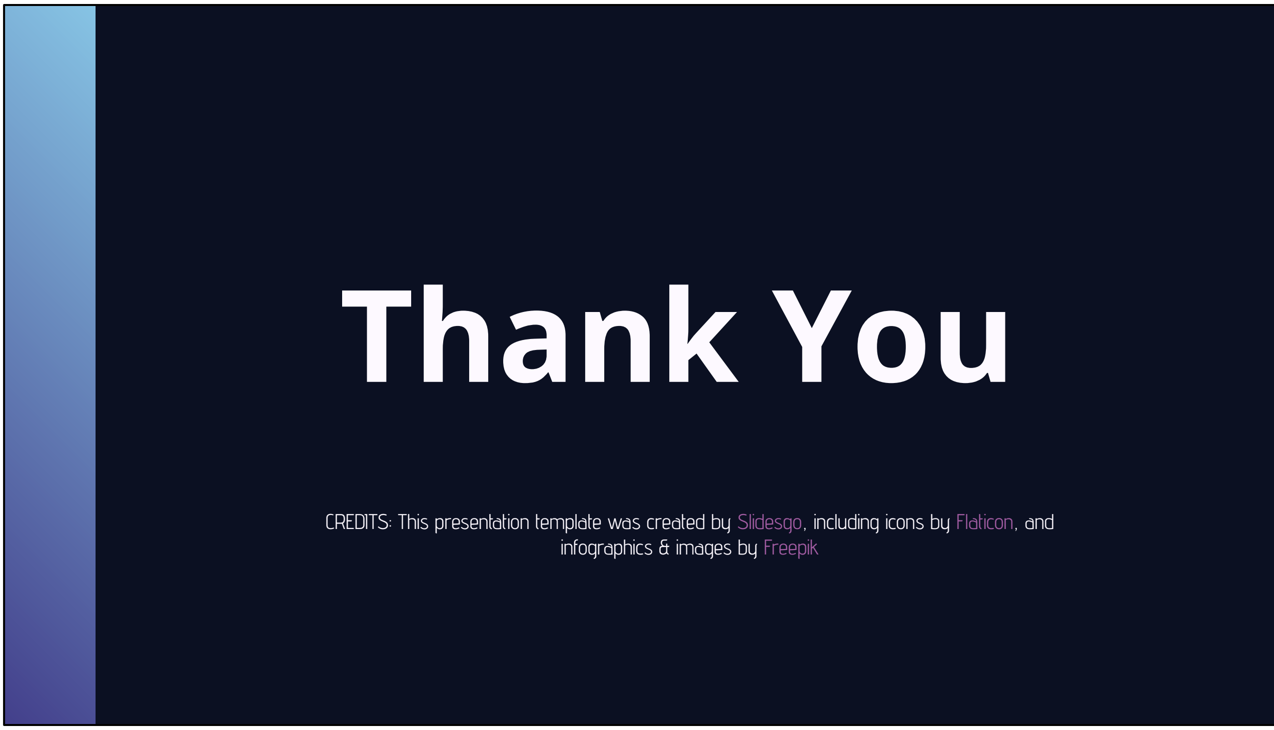
Ash
30 seconds

In conclusion, our application provides an insightful tool for HR departments to predict regrettable employee turnover. Companies that can predict unwanted turnover can save significant resources by protecting their most valuable assets - being their employees. By leveraging simple employee datasets, machine learning, and analytics tailored to HR use cases, our application helps HR teams dramatically increase the sophistication and effectiveness of their talent retention and headcount planning strategies.



Ash
5 Seconds

To adjourn, we would like to thank everyone listed on this slide for making this project possible.



Ash
5 seconds.

And finally, we thank you for your time, and invite all of your questions..

Appendix

List of Features Used

- Age
- Tenure
- Salary_adj
- Salary_adj_cohort_percentile
- Education (one hot encoded - OHE)
- Employment Status (OHE)
- Gender (OHE)
- Marital Status (OHE)
- Race (OHE)

Key Lessons Learned

- **Always Build for the Business Case**
  - Every choice made had to be justified as useful for the end user.
  - Developments done that didn't support business case were a waste of time.
- **Be prepared for things to go wrong**
  - Keras Logistic Regression model couldn't be implemented into app.
  - Logistic Regression results were initially deemed suspicious.
  - MIDS server would up not having expected compute resources, dependencies.

Ash
40 seconds

This project included a number of technical triumphs for us.

We went from zero web experience to creating a full dashboard, using all the tools and platforms listed here, to host and deploy the app on a server.

We learned how to apply statistics within a model pipeline in tasks like model performance testing, time series analysis, feature importance analysis, and chi-squared feature analysis.

And lastly, we learned how to support the business case by designing specifically for non-technical HR users, namely with a stateless application and a dashboard focused on usability and actionality.

—--
Throughout this project, we learned new technical skills and also leveraged what we learned throughout this program.

For web development tools, that includes flask, bootstrap UI, Pydantic evaluations as well as hosting/deploying content to a server.

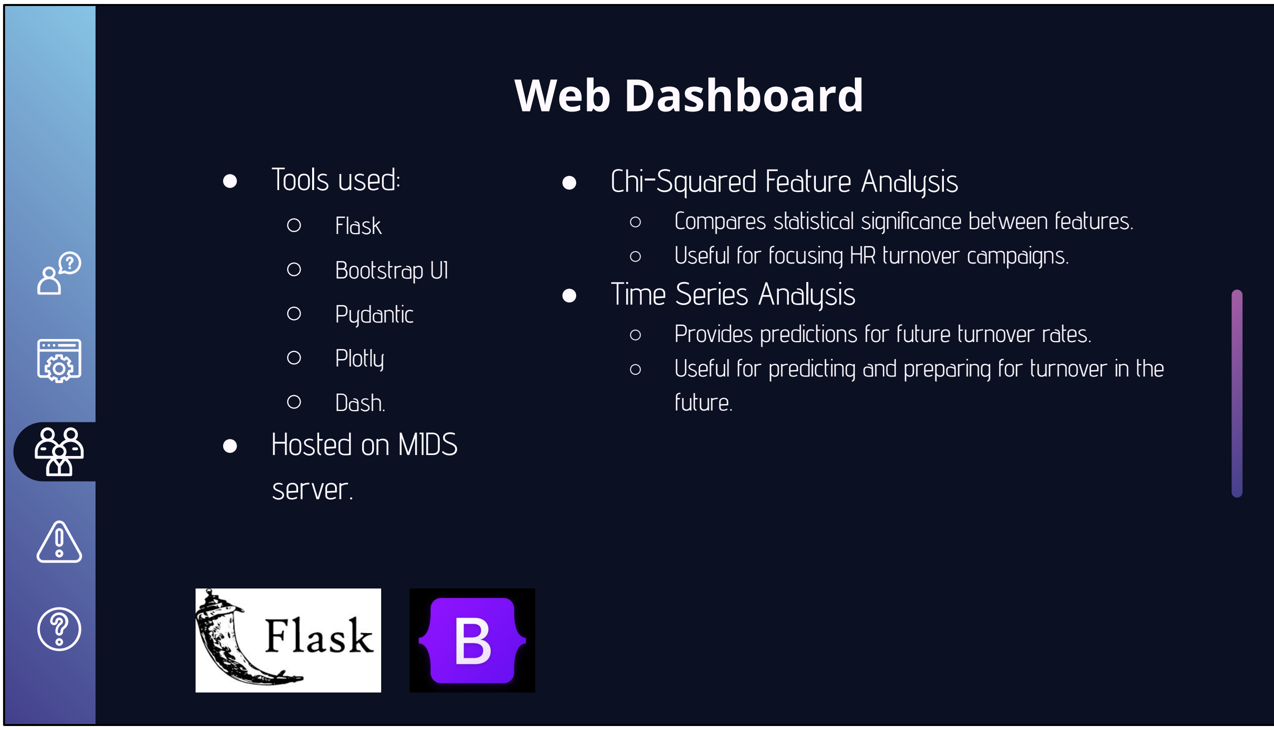
We applied statistics within modeling pipeline where appropriate such as model performance comparison via 5x2 cross validation modified t-test, implemented wald-test to test feature's statistical significance, compared turnover rates within different groups via chi-square test. And last but not least, time series turnover prediction analysis.

All this work was geared towards supporting our business case on helping HR professionals make informed decisions on tackling regrettable turnovers. We made this possible with stateless design intended to serve our users where CSV upload was the easiest method for them.

Application Design Priorities

- **Minimum Viable Product:**
  - Create ML model for predicting employee turnover
  - Dashboard that provides explainable, actionable analysis
  - Stateless design

- **Development Strategy: Move As Fast As Possible**
  - Use familiar and functional tools rather than learn new ones.
  - Consolidate tools and platforms to maximize functionality.
  - Create bare functionality first, then iterate afterwards.



With the data pipeline now completed, let's talk about the infrastructure for deploying the application.

We had very little experience in web development in our team, but we needed a way to make a Python-powered dashboard with a pleasant user interface.

We utilized Flask API's to make our Python code accessible, which in turn allowed us to integrate Pydantic file validation, file ingestion via a Preprocessing Workflow, and chart plotting with Plotly.

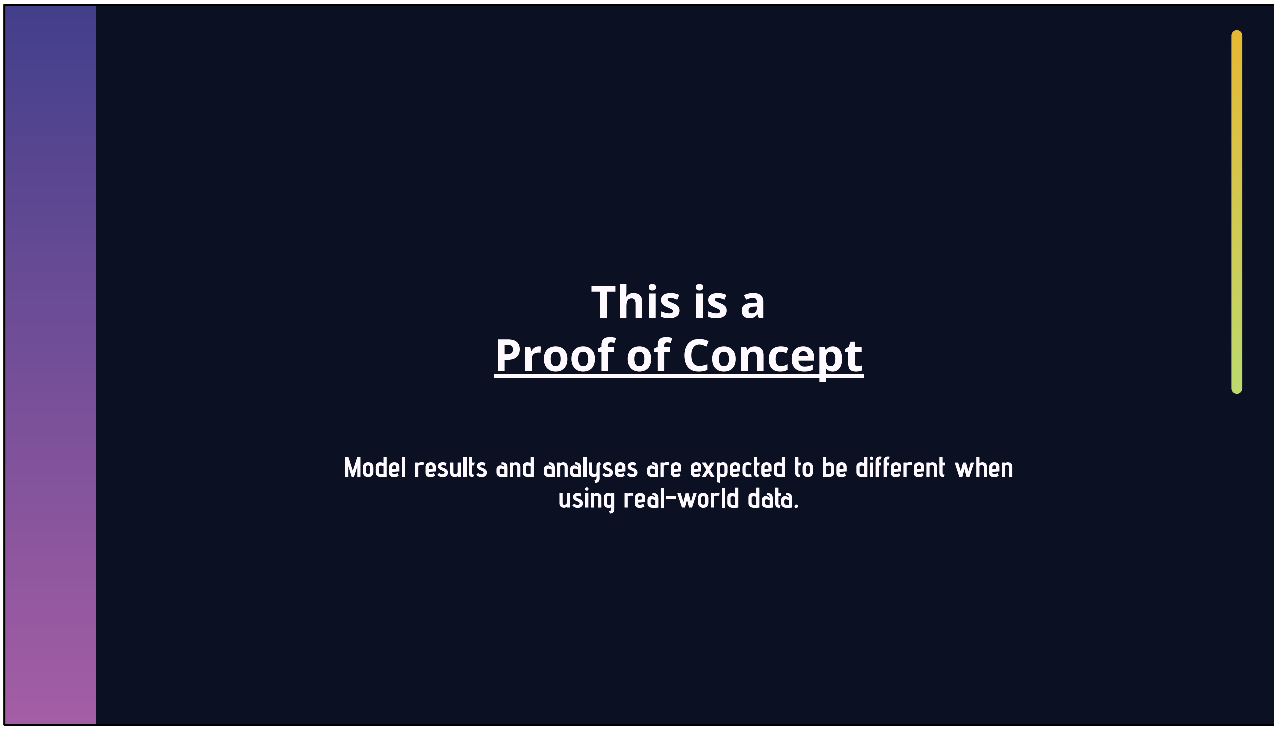
For the user interface, we utilized Bootstrap UI, which offered many useful and visually-pleasing HTML templates that we could easily adapt to our needs.

What's Out There Now?

"Data is treated as a shortcut, not a solution."

"Data without insight can be possibly a dangerous distraction."

- Dylan Gandossy
VP of HR, Strategy, PMO, and Accelerator Team at Workday



Ash
10 Seconds

But remember: this application is a proof of concept that used synthetic data during training. As such, we expect the results to be different when using real-world data.