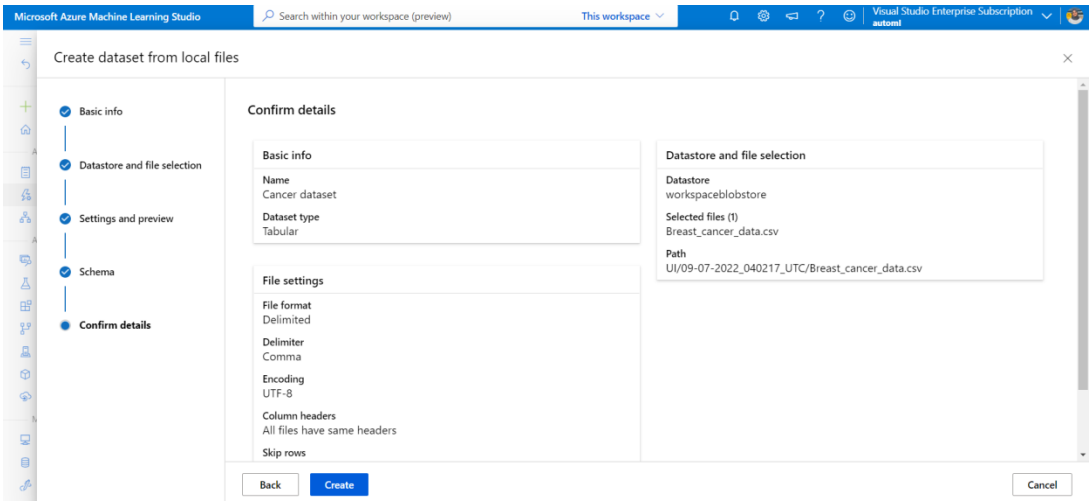
Mini-Project steps briefly

1. **Create new automatedML job using Azure Machine Learning Studio :**
    - Load and configure the Breast cancer dataset
    - Choose the type of problem (classification), the target column(diagnosis) and the validation type. I worked with «Monte Carlo»
    - Creat a compute cluster.
    - Use Azure Container Instance ACI not Azure Kubernetes service AKS since it offers an easy way to run containers in the Azure cloud, eliminating the need to manage virtual machines (VMs) or using more complex container orchestration services and It is ideal for simple container-based workloads like smaller-scale apps, build jobs, and task automation.



2. **Explore models, explanations, metrics :**
    - Analyze the features' importance
    - Visualize the confusion matrix

Cancer prediction

Completed

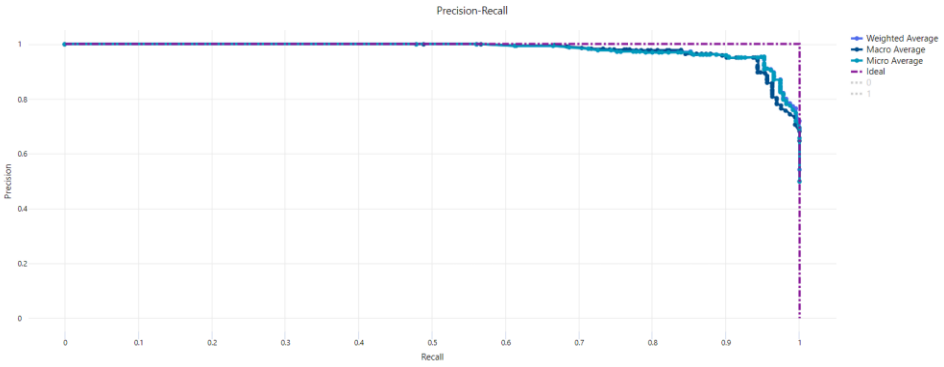
Overview | Data guardrails | **Models** | Outputs + logs | Child jobs

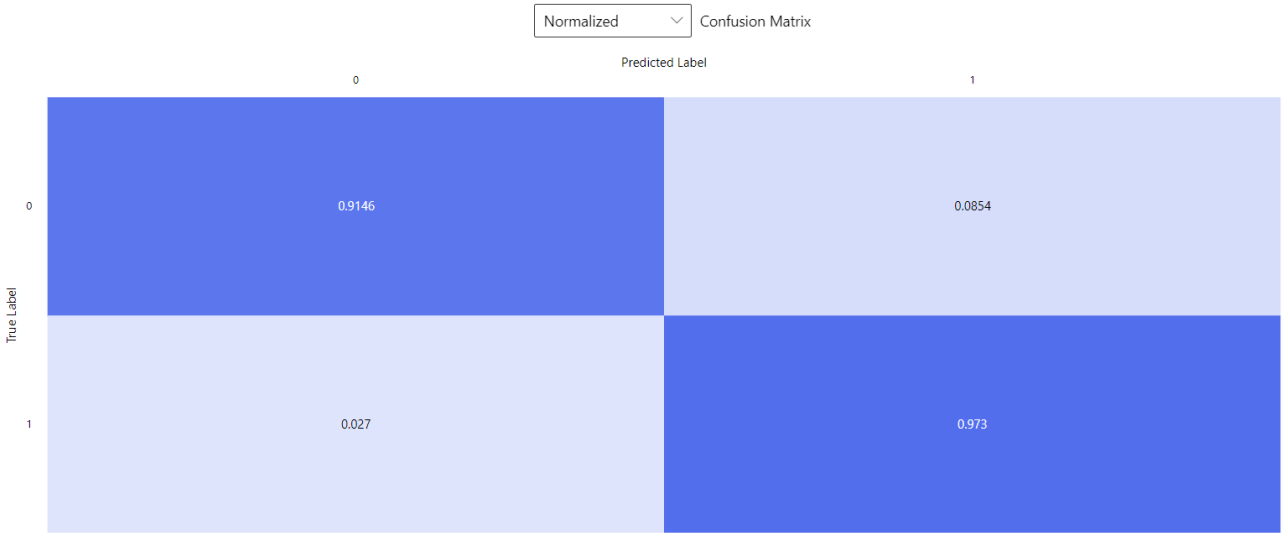
Refresh | Edit and submit (preview) | Register model | Cancel | Delete | Deplo

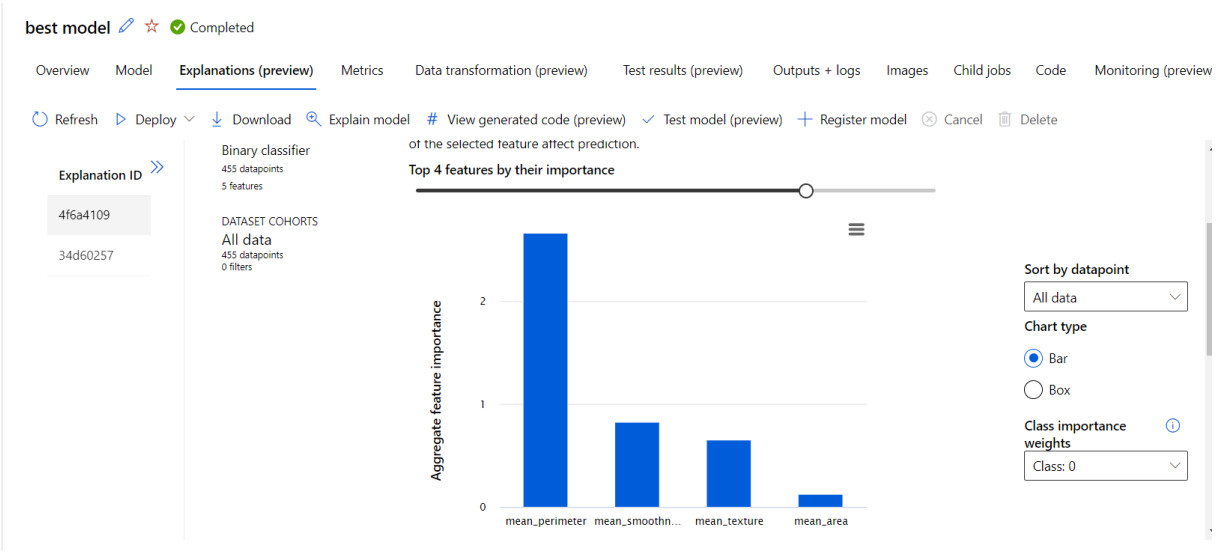
Search

Showing 1-25 of 64 models

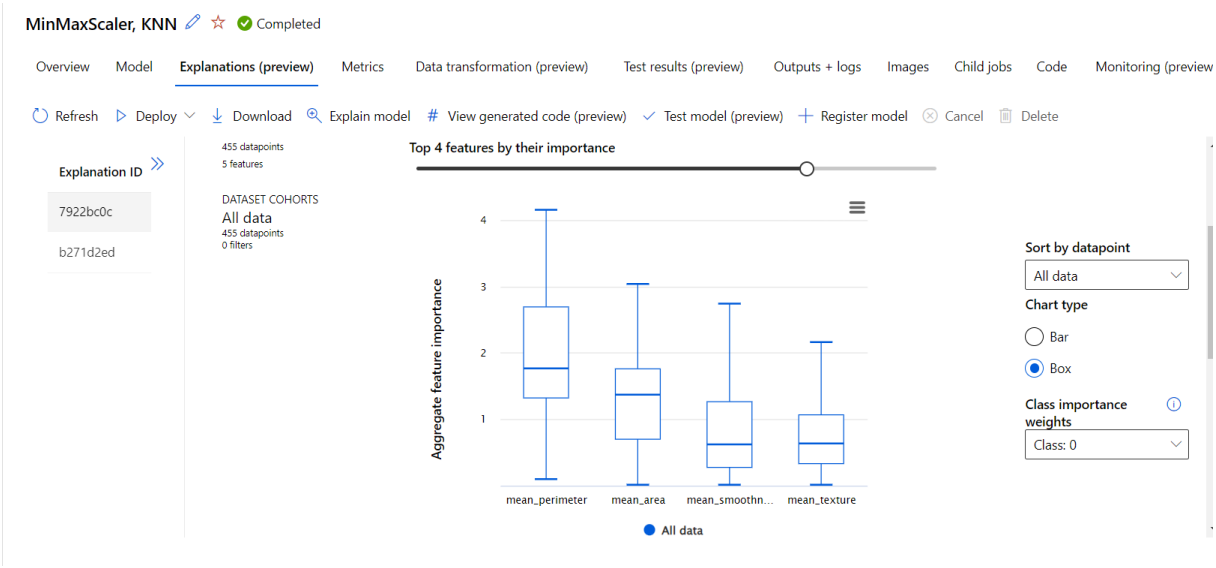
Algorithm name	Explained	Accuracy ↓
VotingEnsemble	View explanation	0.95217
StackEnsemble		0.94783
StandardScalerWrapper, SGD		0.94783
StandardScalerWrapper, LogisticRegression		0.94783
MinMaxScaler, LogisticRegression		0.94783
MaxAbsScaler, LogisticRegression		0.94348



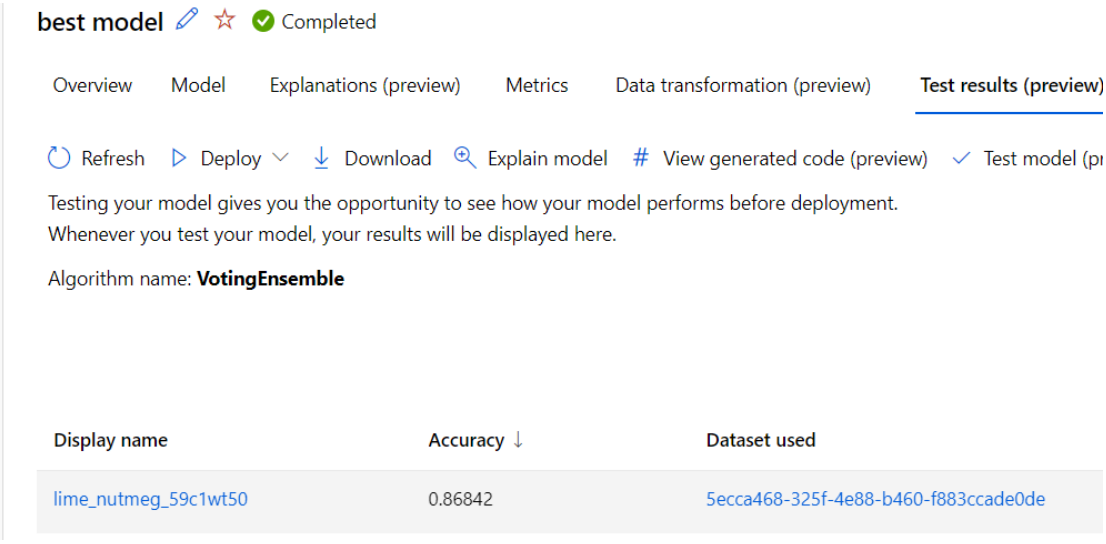




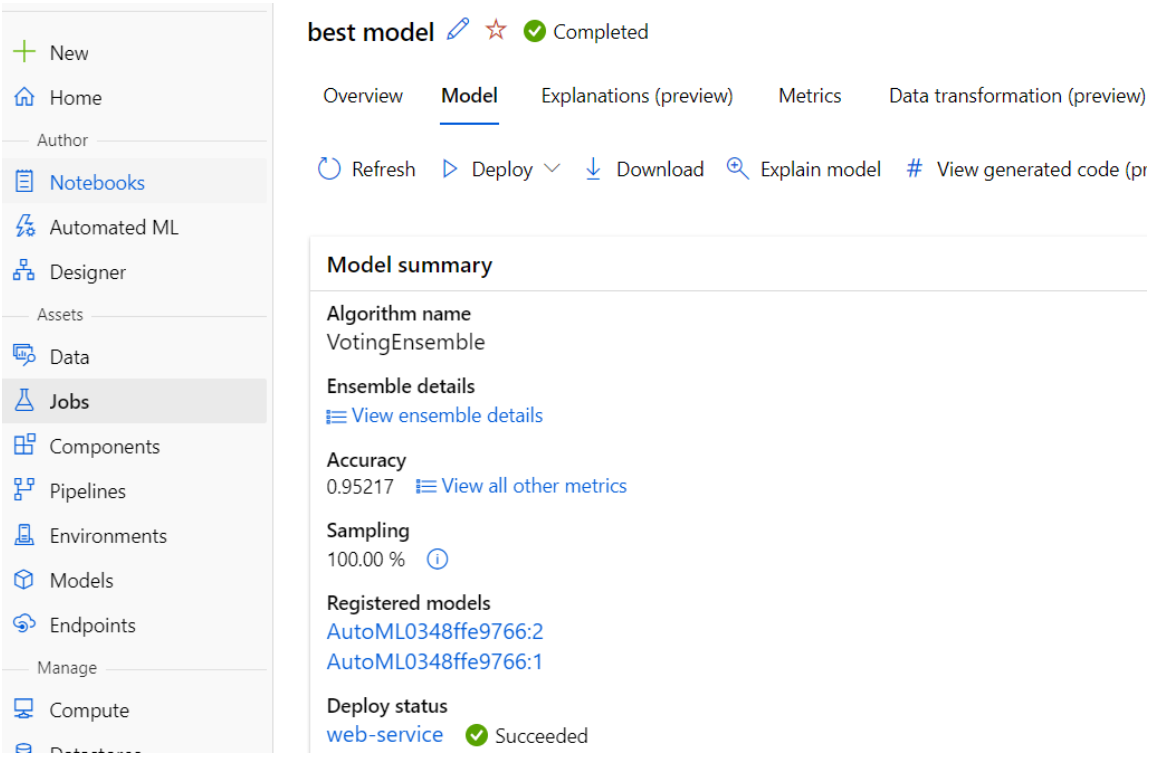
3. **Discover the best model which has the best accuracy.**
4. **Ask for explanations for models which didn't provide the best results.**



5. **Test the model**



6. Deploy the model as a web service



7. Consume the model using PowerBI to predict Breast cancer existence

- Download PowerBI desktop
- PowerBI service subscription
- Load the dataset
- Use « Azure machine learning » to rely the PowerBI desktop with your previous deployed model
- Predict the « disgnosis »

- Visualize the actual and predicted diagnosis values
- Save the report and publish it to PowerBI service's dashboard