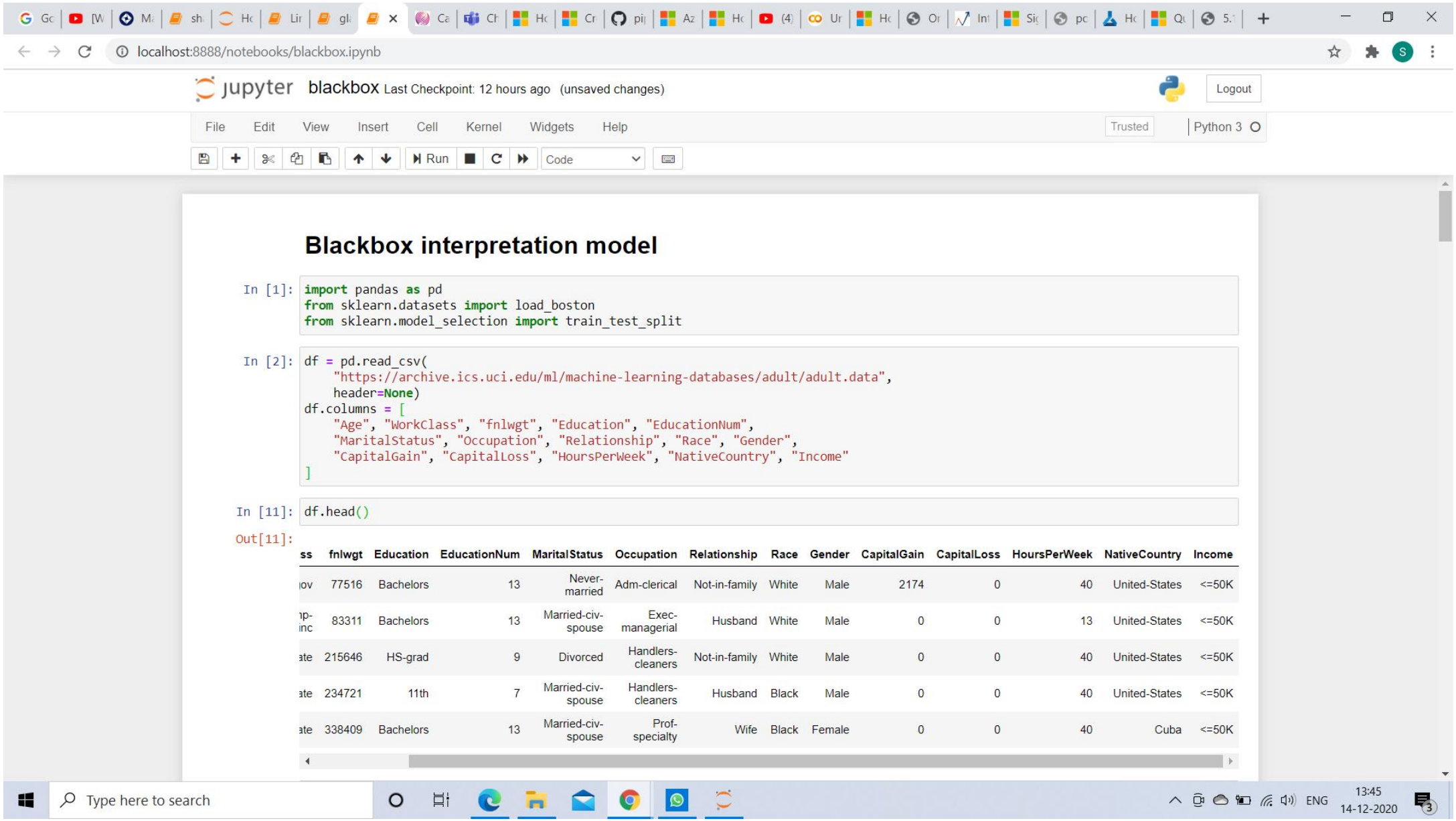


localhost:8888/notebooks/blackbox.ipynb

jupyter blackbox Last Checkpoint: 12 hours ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3

Run Code

Blackbox interpretation model

In [1]:
```
import pandas as pd
from sklearn.datasets import load_boston
from sklearn.model_selection import train_test_split
```

In [2]:
```
df = pd.read_csv(
    "https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data",
    header=None)
df.columns = [
    "Age", "WorkClass", "fnlwgt", "Education", "EducationNum",
    "MaritalStatus", "Occupation", "Relationship", "Race", "Gender",
    "CapitalGain", "CapitalLoss", "HoursPerWeek", "NativeCountry", "Income"
]
```

In [11]:
```
df.head()
```

Out[11]:

ss	fnlwgt	Education	EducationNum	MaritalStatus	Occupation	Relationship	Race	Gender	CapitalGain	CapitalLoss	HoursPerWeek	NativeCountry	Income
ıov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
ıp-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
ate	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K
ate	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K
ate	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<=50K

Type here to search

ENG 13:45 14-12-2020

localhost:8888/notebooks/blackbox.ipynb

jupyter **blackbox** Last Checkpoint: 12 hours ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Run | Code

ate	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K
ate	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K
ate	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<=50K

In [3]:
```
train_cols = df.columns[0:-1]
label = df.columns[-1]
X = df[train_cols]
y = df[label].apply(lambda x: 0 if x == " <=50K" else 1) #Turning response into 0 and 1

# To transform categorical variables to use sklearn models
X_enc = pd.get_dummies(X, prefix_sep='.')
feature_names = list(X_enc.columns)

seed = 1
X_train, X_test, y_train, y_test = train_test_split(X_enc, y, test_size=0.20, random_state=seed)
```

In [4]:
```
# Training the model

from sklearn.ensemble import RandomForestClassifier
from sklearn.decomposition import PCA
from sklearn.pipeline import Pipeline

pca = PCA()
rf = RandomForestClassifier(n_estimators=100, n_jobs=-1)

blackbox_model = Pipeline([('pca', pca), ('rf', rf)])
blackbox_model.fit(X_train, y_train)
```

Out[4]:
```
Pipeline(steps=[('pca', PCA()), ('rf', RandomForestClassifier(n_jobs=-1))])
```

In [5]:
```
# Model performance
from interpret import show
from interpret.perf import ROC
```

Type here to search

ENG 13:45 14-12-2020

localhost:8888/notebooks/blackbox.ipynb

jupyter **blackbox** Last Checkpoint: 12 hours ago (unsaved changes)

Logout

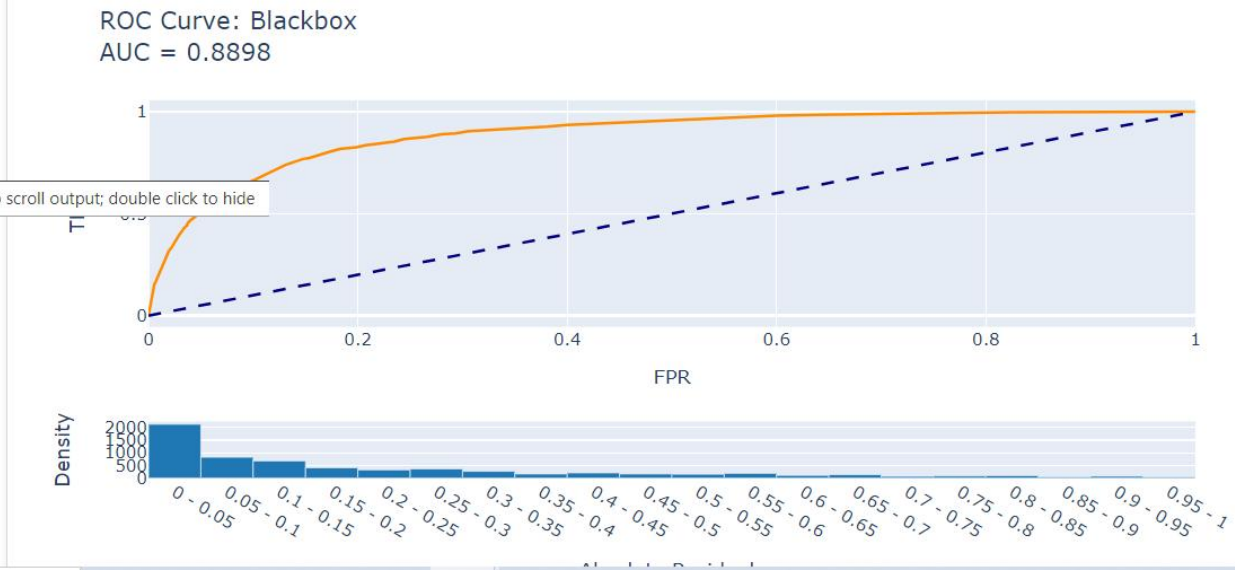
File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Run | Code

In [5]:
```
# Model performance
from interpret import show
from interpret.perf import ROC

blackbox_perf = ROC(blackbox_model.predict_proba).explain_perf(X_test, y_test, name='Blackbox')
show(blackbox_perf)
```

Blackbox (Overall)



click to scroll output; double click to hide

Type here to search

ENG 13:45 14-12-2020

localhost:8888/notebooks/blackbox.ipynb

jupyter blackbox Last Checkpoint: 12 hours ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Run Code

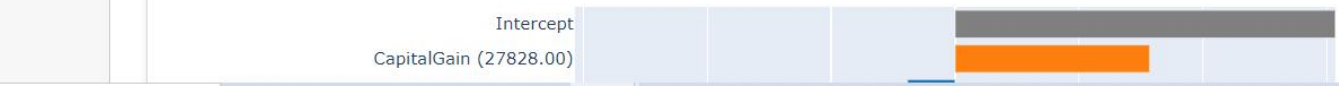
In [6]:
```
# Local Explanations: How an individual prediction was made
from interpret.blackbox import LimeTabular
from interpret import show

#Blackbox explainers need a predict function, and optionally a dataset
lime = LimeTabular(predict_fn=blackbox_model.predict_proba, data=X_train, random_state=1)

lime_local = lime.explain_local(X_test[:5], y_test[:5], name='LIME')

show(lime_local)
```

Select Component to Graph

2 : Predicted (0.64) | Actual (1.0) | Resid (0.36)

LIME [2]

Predicted (0.64) | Actual (1)



Type here to search

ENG 13:45 14-12-2020

localhost:8888/notebooks/blackbox.ipynb

jupyter blackbox Last Checkpoint: 12 hours ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Run | Code

In [7]:
```
from interpret.blackbox import ShapKernel
import numpy as np

background_val = np.median(X_train, axis=0).reshape(1, -1)
shap = ShapKernel(predict_fn=blackbox_model.predict_proba, data=background_val, feature_names=feature_names)
shap_local = shap.explain_local(X_test[:5], y_test[:5], name='SHAP')
show(shap_local)
```

100% 5/5 [00:06<00:00, 1.27s/it]

Select Component to Graph

2 : Predicted (0.64) | Actual (1.0) | Resid (0.36)

SHAP [2]

Predicted (0.64) | Actual (1)

Base Value

Type here to search

ENG 13:45 14-12-2020

localhost:8888/notebooks/blackbox.ipynb

jupyter blackbox Last Checkpoint: 12 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3

Run Code

Morris sensitivity

This method is also called the elementary effects method,which can effectively identify and rank the importance of input parameters of a model by changing the value of only 1 parameter in an instance and finding its effect on the model output.

In [8]:
```
# Global Explanations: How the model behaves overall
from interpret.blackbox import MorrisSensitivity

sensitivity = MorrisSensitivity(predict_fn=blackbox_model.predict_proba, data=X_train)
sensitivity_global = sensitivity.explain_global(name="Global Sensitivity")

show(sensitivity_global)
```

Select Component to Graph

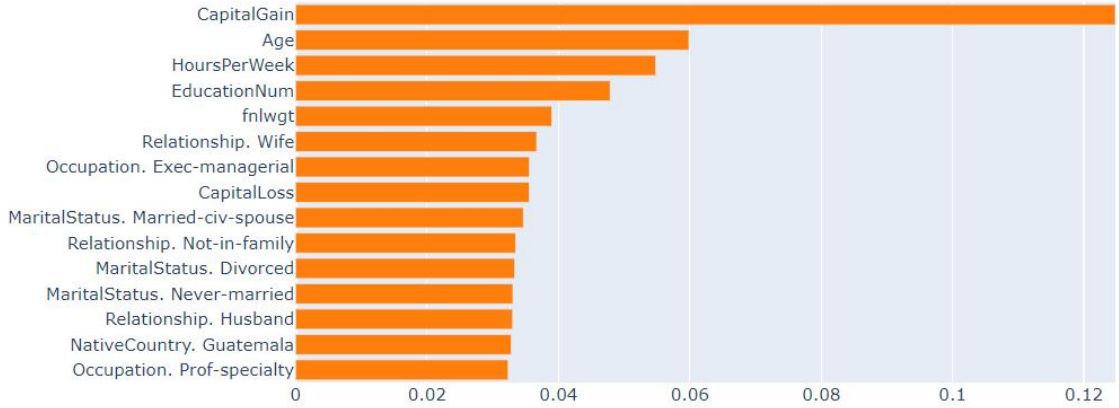
Summary

Global Sensitivity (Overall)

Morris Sensitivity
Convergence Index: 0.070

Type here to search

ENG 13:45 14-12-2020



Partial dependence plot(PDP)

The partial dependence plot shows the marginal effect one or two features have on the predicted outcome of a machine learning model . A partial dependence plot can show whether the relationship between the target and a feature is linear, monotonic or more complex.

localhost:8888/notebooks/blackbox.ipynb

jupyter blackbox Last Checkpoint: 12 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Run Code

feature is linear, monotonic or more complex.

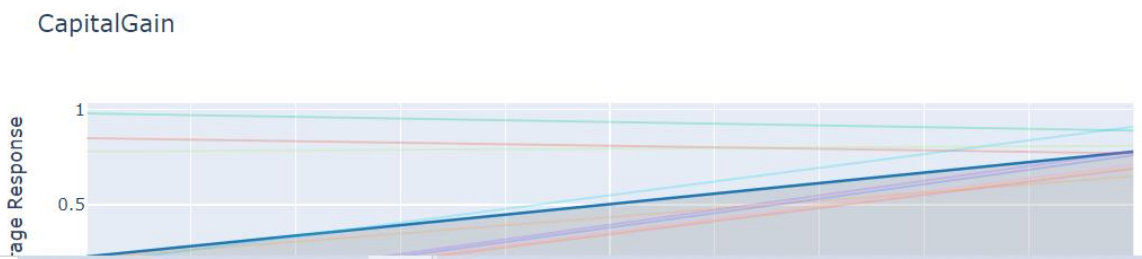
In [9]:

```
from interpret.blackbox import PartialDependence

pdp = PartialDependence(predict_fn=blackbox_model.predict_proba, data=X_train)
pdp_global = pdp.explain_global(name='Partial Dependence')

show(pdp_global)
```

Select Component to Graph

3 : Name (CapitalGain) | Type (continuous) | # Unique (116)

Partial Dependence [3]



Type here to search

ENG 13:46 14-12-2020

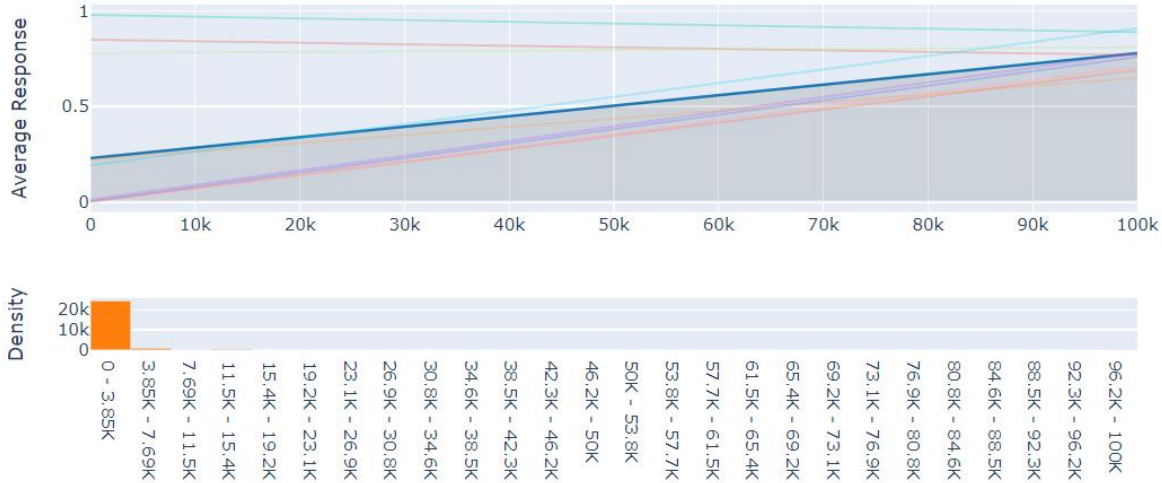
jupyter blackbox Last Checkpoint: 12 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Run | Code

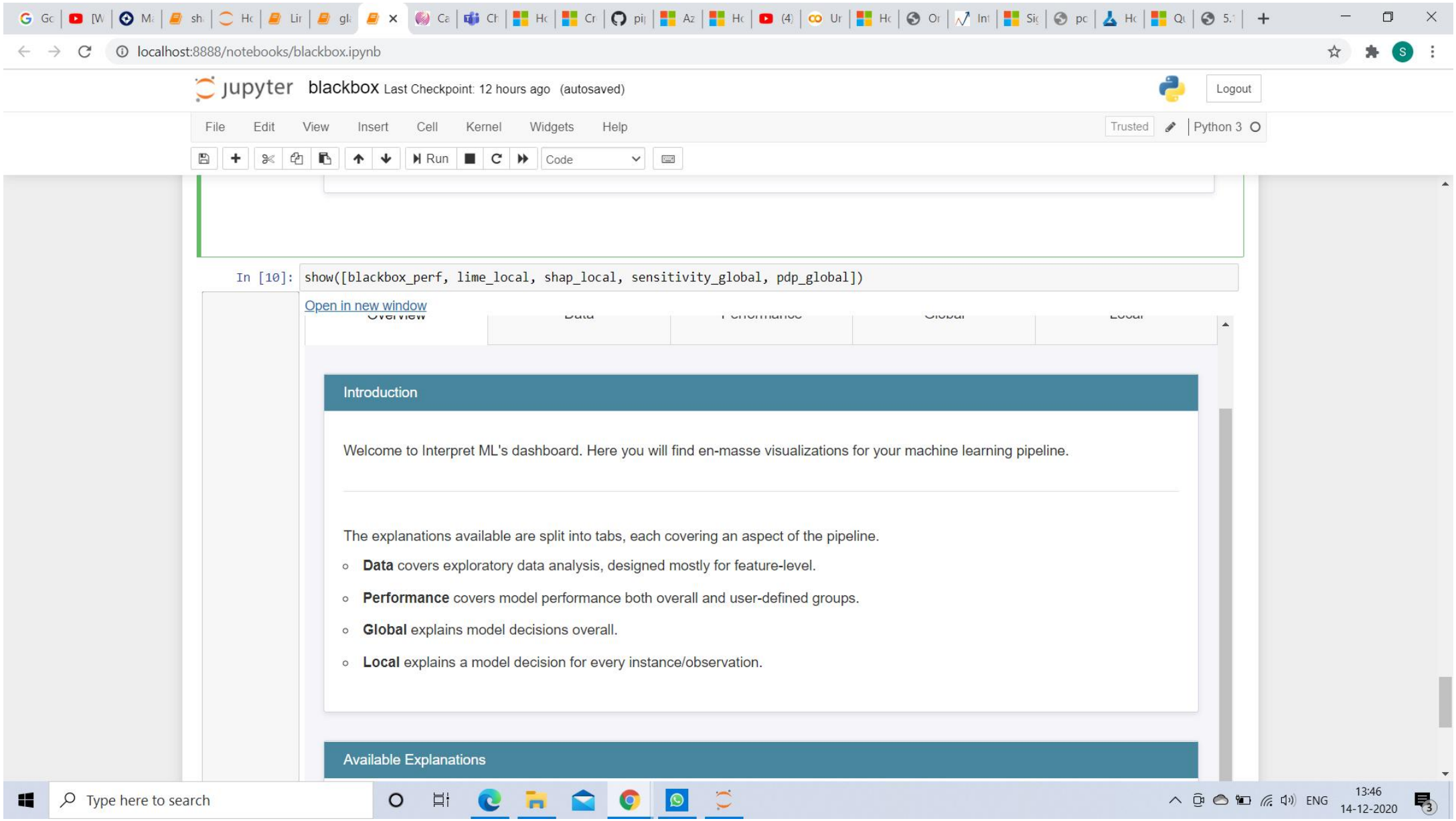


In [10]:
```
show([blackbox_perf, lime_local, shap_local, sensitivity_global, pdp_global])
```

Open in new window

Interpret ML Dashboard

Overview | Data | Performance | Global | Local


localhost:8888/notebooks/blackbox.ipynb
jupyter blackbox Last Checkpoint: 12 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
In [10]: show([blackbox_perf, lime_local, shap_local, sensitivity_global, pdp_global])
Open in new window
Introduction
Welcome to Interpret ML's dashboard. Here you will find en-masse visualizations for your machine learning pipeline.
The explanations available are split into tabs, each covering an aspect of the pipeline.
Data covers exploratory data analysis, designed mostly for feature-level.
Performance covers model performance both overall and user-defined groups.
Global explains model decisions overall.
Local explains a model decision for every instance/observation.
Available Explanations
Type here to search
ENG
13:46
14-12-2020