In [15]:

```
import warnings
warnings.filterwarnings('ignore')
import numpy as np
import pandas as pd
from catboost import CatBoostClassifier
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import RepeatedStratifiedKFold
```

In [2]:

```
CITY = 'Atlanta'
X_train = np.load('train_set/X_train_'+CITY+'.npy',allow_pickle=True)[:,0:-1]
y_train = np.load('train_set/y_train_'+CITY+'.npy',allow_pickle=True)
X_test = np.load('train_set/X_test_'+CITY+'.npy',allow_pickle=True)[:,0:-1]
y_test = np.load('train_set/y_test_'+CITY+'.npy',allow_pickle=True)
```

In [3]:

```
model = CatBoostClassifier(verbose=0, n_estimators=100)
#cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)
#n_scores = cross_val_score(model, X_train, y_train, scoring='accuracy', cv=cv, n_jo
bs=-1, error_score='raise')

# fit the model on the whole dataset
model = CatBoostClassifier(verbose=0, n_estimators=100)
model.fit(X_train, y_train)
```

Out[3]:

```
<catboost.core.CatBoostClassifier at 0x8ae0743160>
```

In [4]:

```
y_pred = pd.DataFrame(model.predict(X_test),columns=['pred'])
```

In [5]:

```
X_testdf = pd.DataFrame(X_test)
```

In [13]:

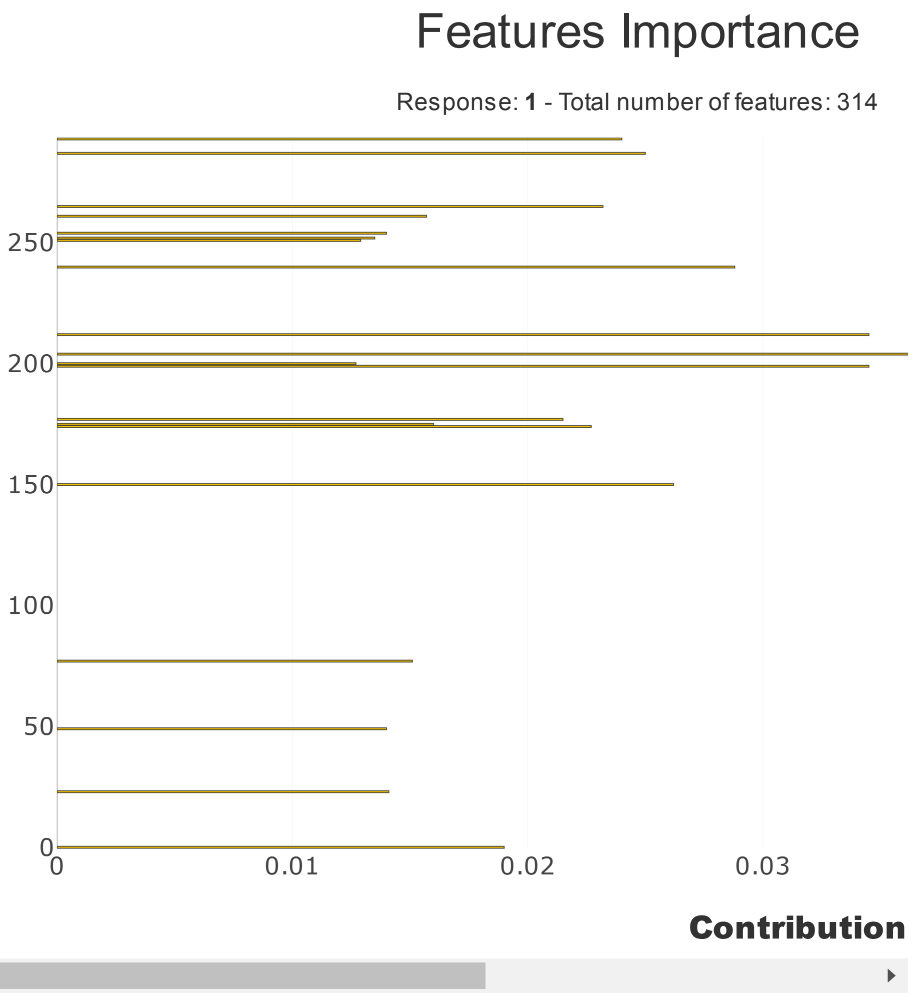
```
from sklearn.metrics import f1_score
precision = 2
f1_weighted = f1_score(y_test, y_pred, average='weighted')
f1_macro = f1_score(y_test, y_pred, average='macro')
f1_micro = f1_score(y_test, y_pred, average='micro')
f1_None = f1_score(y_test, y_pred, average=None)

print("f1_weighted:", round(f1_weighted, precision), "\n",
      "f1_macro:", round(f1_macro, precision),"\n",
      "f1_micro:", round(f1_micro, precision), "\n",
      "f1_None:", f1_None, "\n")
```

```
f1_weighted: 0.83
 f1_macro: 0.73
 f1_micro: 0.84
 f1_None: [0.90564226 0.54566474]
```

In [16]:

```
from shapash.explainer.smart_explainer import SmartExplainer
xpl = SmartExplainer()
xpl.compile(x=X_testdf, model=model, y_pred=y_pred)
xpl.plot.features_importance()
```

```
Backend: Shap TreeExplainer
```



In [17]:

```
app = xpl.run_app(title_story='accident prediction')
```

```
Dash is running on http://0.0.0.0:8050/

INFO:root:Your Shapash application run on http://chetans:8050/


INFO:shapash.webapp.smart_app:Dash is running on http://0.0.0.0:8050/

INFO:root:Use the method .kill() to down your app.

 * Serving Flask app "shapash.webapp.smart_app" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environmen
t.
   Use a production WSGI server instead.
 * Debug mode: off

INFO:werkzeug: * Running on http://0.0.0.0:8050/ (Press CTRL+C to quit)
```

In [ ]:

```
app.kill()
```