```
TypeError: no supported conversion for types: (dtype('O'),)
---------------------------------------------------------------------------
-----
TypeError                                 Traceback (most recent call
last)
File /databricks/python/lib/python3.10/site-packages/scipy/sparse/
_base.py:376, in spmatrix.asformat(self, format, copy)
    375 try:
--> 376     return convert_method(copy=copy)
    377 except TypeError:

File /databricks/python/lib/python3.10/site-packages/scipy/sparse/
_coo.py:402, in coo_matrix.tocsr(self, copy)
    401 indices = np.empty_like(col, dtype=idx_dtype)
--> 402 data = np.empty_like(self.data, dtype=upcast(self.dtype))
    404 coo_tocsr(M, N, self.nnz, row, col, self.data,
    405           indptr, indices, data)

File /databricks/python/lib/python3.10/site-packages/scipy/sparse/
_sputils.py:50, in upcast(*args)
     48         return t
---> 50 raise TypeError('no supported conversion for types: %r' %
(args,))

TypeError: no supported conversion for types: (dtype('O'),)

During handling of the above exception, another exception occurred:

TypeError                                 Traceback (most recent call
last)
File <command-2703671354833513>, line 17
     15 # Use Kernel SHAP to explain feature importance on the sampled
rows from the validation set.
     16 predict = lambda x: model.predict_proba(pd.DataFrame(x,
columns=X_train.columns))
---> 17 explainer = KernelExplainer(predict, train_sample,
link="logit")
     18 shap_values = explainer.shap_values(example, l1_reg=False,
nsamples=500)
     19 summary_plot(shap_values, example, class_names=model.classes_)

File /databricks/python/lib/python3.10/site-packages/shap/explainers/
_kernel.py:70, in Kernel.__init__(self, model, data, link, **kwargs)
     68 self.keep_index_ordered = kwargs.get("keep_index_ordered",
False)
     69 self.data = convert_to_data(data, keep_index=self.keep_index)
---> 70 model_null = match_model_to_data(self.model, self.data)
     72 # enforce our current input type limitations
     73 assert isinstance(self.data, DenseData) or
isinstance(self.data, SparseData), \
```

```
     74        "Shap explainer only supports the DenseData and
SparseData input currently."

File /databricks/python/lib/python3.10/site-packages/shap/utils/
_legacy.py:112, in match_model_to_data(model, data)
    110         out_val = model.f(data.convert_to_df())
    111     else:
--> 112         out_val = model.f(data.data)
    113 except:
    114     print("Provided model function fails when applied to the
provided data set.")

File <command-2703671354833513>, line 16, in <lambda>(x)
     13 example = X_val.sample(n=min(100, X_val.shape[0]),
random_state=527471026).fillna(mode)
     15 # Use Kernel SHAP to explain feature importance on the sampled
rows from the validation set.
---> 16 predict = lambda x: model.predict_proba(pd.DataFrame(x,
columns=X_train.columns))
     17 explainer = KernelExplainer(predict, train_sample,
link="logit")
     18 shap_values = explainer.shap_values(example, l1_reg=False,
nsamples=500)

File /databricks/python/lib/python3.10/site-packages/sklearn/
pipeline.py:523, in Pipeline.predict_proba(self, X,
**predict_proba_params)
    521 Xt = X
    522 for _, name, transform in self._iter(with_final=False):
--> 523     Xt = transform.transform(Xt)
    524 return self.steps[-1][1].predict_proba(Xt,
**predict_proba_params)

File /databricks/python/lib/python3.10/site-packages/sklearn/compose/
_column_transformer.py:746, in ColumnTransformer.transform(self, X)
    741 else:
    742     # ndarray was used for fitting or transforming, thus we
only
    743     # check that n_features_in_ is consistent
    744     self._check_n_features(X, reset=False)
--> 746 Xs = self._fit_transform(
    747     X,
    748     None,
    749     _transform_one,
    750     fitted=True,
    751     column_as_strings=fit_dataframe_and_transform_dataframe,
    752 )
    753 self._validate_output(Xs)
    755 if not Xs:
    756     # All transformers are None
```

```
File /databricks/python/lib/python3.10/site-packages/sklearn/compose/
_column_transformer.py:604, in ColumnTransformer._fit_transform(self,
X, y, func, fitted, column_as_strings)
    598 transformers = list(
    599     self._iter(
    600         fitted=fitted, replace_strings=True,
column_as_strings=column_as_strings
    601     )
    602 )
    603 try:
--> 604     return Parallel(n_jobs=self.n_jobs)(
    605         delayed(func)(
    606             transformer=clone(trans) if not fitted else trans,
    607             X=_safe_indexing(X, column, axis=1),
    608             y=y,
    609             weight=weight,
    610             message_clsname="ColumnTransformer",
    611             message=self._log_message(name, idx,
len(transformers)),
    612         )
    613         for idx, (name, trans, column, weight) in
enumerate(transformers, 1)
    614     )
    615 except ValueError as e:
    616     if "Expected 2D array, got 1D array instead" in str(e):

File /databricks/python/lib/python3.10/site-packages/joblib/
parallel.py:1088, in Parallel.__call__(self, iterable)
   1085 if self.dispatch_one_batch(iterator):
   1086     self._iterating = self._original_iterator is not None
-> 1088 while self.dispatch_one_batch(iterator):
   1089     pass
   1091 if pre_dispatch == "all" or n_jobs == 1:
   1092     # The iterable was consumed all at once by the above for
loop.
   1093     # No need to wait for async callbacks to trigger to
   1094     # consumption.

File /databricks/python/lib/python3.10/site-packages/joblib/
parallel.py:901, in Parallel.dispatch_one_batch(self, iterator)
    899     return False
    900 else:
--> 901     self._dispatch(tasks)
    902     return True

File /databricks/python/lib/python3.10/site-packages/joblib/
parallel.py:819, in Parallel._dispatch(self, batch)
    817 with self._lock:
    818     job_idx = len(self._jobs)
```

```
--> 819     job = self._backend.apply_async(batch, callback=cb)
    820     # A job can complete so quickly than its callback is
    821     # called before we get here, causing self._jobs to
    822     # grow. To ensure correct results ordering, .insert is
    823     # used (rather than .append) in the following line
    824     self._jobs.insert(job_idx, job)

File /databricks/python/lib/python3.10/site-packages/joblib/
_parallel_backends.py:208, in SequentialBackend.apply_async(self,
func, callback)
    206 def apply_async(self, func, callback=None):
    207     """Schedule a func to be run"""
--> 208     result = ImmediateResult(func)
    209     if callback:
    210         callback(result)

File /databricks/python/lib/python3.10/site-packages/joblib/
_parallel_backends.py:597, in ImmediateResult.__init__(self, batch)
    594 def __init__(self, batch):
    595     # Don't delay the application, to avoid keeping the input
    596     # arguments in memory
--> 597     self.results = batch()

File /databricks/python/lib/python3.10/site-packages/joblib/
parallel.py:288, in BatchedCalls.__call__(self)
    284 def __call__(self):
    285     # Set the default nested backend to self._backend but do
not set the
    286     # change the default number of processes to -1
    287     with parallel_backend(self._backend, n_jobs=self._n_jobs):
--> 288         return [func(*args, **kwargs)
    289                 for func, args, kwargs in self.items]

File /databricks/python/lib/python3.10/site-packages/joblib/
parallel.py:288, in <listcomp>(.0)
    284 def __call__(self):
    285     # Set the default nested backend to self._backend but do
not set the
    286     # change the default number of processes to -1
    287     with parallel_backend(self._backend, n_jobs=self._n_jobs):
--> 288         return [func(*args, **kwargs)
    289                 for func, args, kwargs in self.items]

File /databricks/python/lib/python3.10/site-packages/sklearn/utils/
fixes.py:117, in _FuncWrapper.__call__(self, *args, **kwargs)
    115 def __call__(self, *args, **kwargs):
    116     with config_context(**self.config):
--> 117         return self.function(*args, **kwargs)

File /databricks/python/lib/python3.10/site-packages/sklearn/
```

```
pipeline.py:853, in _transform_one(transformer, X, y, weight,
**fit_params)
    852 def _transform_one(transformer, X, y, weight, **fit_params):
--> 853     res = transformer.transform(X)
    854     # if we have a weight for this transformer, multiply
output
    855     if weight is None:

File /databricks/python/lib/python3.10/site-packages/sklearn/
pipeline.py:635, in Pipeline.transform(self, X)
    633 Xt = X
    634 for _, _, transform in self._iter():
--> 635     Xt = transform.transform(Xt)
    636 return Xt

File /databricks/python/lib/python3.10/site-packages/databricks/
automl_runtime/sklearn/one_hot_encoder.py:94, in
OneHotEncoder.transform(self, X)
     92 X_updated = self.base_one_hot_encoder.transform(X)
     93 if self.sparse:
---> 94     X_updated = sparse.csr_matrix(X_updated)
     95 return X_updated

File /databricks/python/lib/python3.10/site-packages/scipy/sparse/
_compressed.py:84, in _cs_matrix.__init__(self, arg1, shape, dtype,
copy)
     81     except Exception as e:
     82         raise ValueError("unrecognized {}_matrix constructor
usage"
     83                          "".format(self.format)) from e
---> 84     self._set_self(self.__class__(
     85         self._coo_container(arg1, dtype=dtype)
     86     ))
     88 # Read matrix dimensions given, if any
     89 if shape is not None:

File /databricks/python/lib/python3.10/site-packages/scipy/sparse/
_compressed.py:33, in _cs_matrix.__init__(self, arg1, shape, dtype,
copy)
     31         arg1 = arg1.copy()
     32     else:
---> 33         arg1 = arg1.asformat(self.format)
     34     self._set_self(arg1)
     36 elif isinstance(arg1, tuple):

File /databricks/python/lib/python3.10/site-packages/scipy/sparse/
_base.py:378, in spmatrix.asformat(self, format, copy)
    376     return convert_method(copy=copy)
    377 except TypeError:
--> 378     return convert_method()
```

```
File /databricks/python/lib/python3.10/site-packages/scipy/sparse/
_coo.py:402, in coo_matrix.tocsr(self, copy)
    400 indptr = np.empty(M + 1, dtype=idx_dtype)
    401 indices = np.empty_like(col, dtype=idx_dtype)
--> 402 data = np.empty_like(self.data, dtype=upcast(self.dtype))
    404 coo_tocsr(M, N, self.nnz, row, col, self.data,
    405           indptr, indices, data)
    407 x = self._csr_container((data, indices, indptr),
shape=self.shape)

File /databricks/python/lib/python3.10/site-packages/scipy/sparse/
_sputils.py:50, in upcast(*args)
     47         _upcast_memo[hash(args)] = t
     48         return t
---> 50 raise TypeError('no supported conversion for types: %r' %
(args,))

TypeError: no supported conversion for types: (dtype('O'),)
```